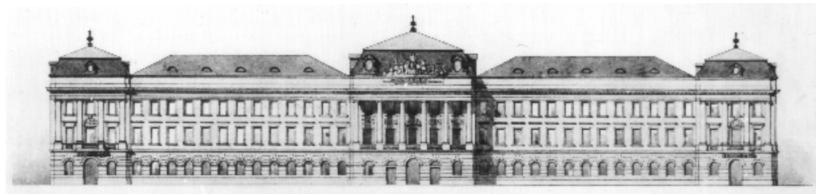
dbai

TECHNICAL

REPORT



INSTITUT FÜR INFORMATIONSSYSTEME

ABTEILUNG DATENBANKEN UND ARTIFICIAL INTELLIGENCE

Intertranslatability of Abstract Argumentation Frameworks

DBAI-TR-2017-104

Sylwia Polberg

DBAI TECHNICAL REPORT

2017

Institut für Informationssysteme
Abteilung Datenbanken und
Artificial Intelligence
Technische Universität Wien
Favoritenstr. 9
A-1040 Vienna, Austria
Tel: +43-1-58801-18403
Fax: +43-1-58801-18493
sekret@dbai.tuwien.ac.at
www.dbai.tuwien.ac.at

TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

DBAI TECHNICAL REPORT
DBAI TECHNICAL REPORT DBAI-TR-2017-104, 2017

Intertranslatability of Abstract Argumentation Frameworks

Sylwia Polberg [1]

Abstract. At the heart of abstract argumentation lies the Dung's argumentation framework. Over the years, many of its extensions were proposed, ranging from the ones employing various strengths and preferences to those that focus on researching new types of relations between arguments. With such an amount of available structures, it is only natural to ask whether one can move between the frameworks while still preserving the behavior of the semantics, what would be the costs of such a process and what we gain or lose in it. The aim of this work is to introduce new translations between the available frameworks and to recall the existing ones when possible. Thus, our aim is to create a comprehensive study on the intertranslatability of abstract argumentation frameworks. We also propose a translation classification system and new transformation approaches for structures with support. Finally, we discuss the quality of our translations, point out what can be improved and if possible, show the limits of the enhancements we can make.

[1]The author is a PhD student of Technical University of Vienna, Austria, and is currently employed by University College London, United Kingdom.

Contents

1 Introduction

Over the last years, argumentation has become an influential subfield of artificial intelligence, with applications ranging from legal reasoning [16] to dialogues and persuasion [60, 80] to medicine [46, 54] and eGovernment [6]. Various formalisms and classifications of types of argumentation have been created [83]. In principle, we can distinguish two major lines of research: logic–based and abstract approaches. In the former [17], we assume arguments have a certain logical structure. This provides us with means of constructing the arguments from an underlying knowledge base and allows us to create relations between them in terms of the properties of their structure [51]. In the abstract approaches, with which we will work in this report, we consider arguments as abstract atomic entities and focus entirely on the relations between them. Thus, we assume that these elements have already been constructed, for instance from a given knowledge base, and proceed with evaluating the obtained argumentation system. The answers are often given in the form of sets of arguments that can be jointly accepted and meet our requirements. Finally, these results are interpreted in terms of the original knowledge base. This three–step creation, evaluation and interpretation process is known as the argumentation process or instantiation–based argumentation [24, 26]. Although abstract argumentation can be seen as mostly concerned with the middle step, we will see that framework intertranslatability often resembles the whole process itself.

At the heart of abstract argumentation lies the argumentation framework developed by Phan Minh Dung [36]. Since the structure itself was relatively limited, as it took into account only the conflict relation between the arguments, it inspired the search for more general models [22]. Throughout the years, many of its extensions have been proposed, ranging from the ones employing various strengths and preferences [5, 14, 62] to those that focus on researching new types of relations between arguments [9, 23, 30, 68, 69, 78]. Such an amount of frameworks should not come as a surprise. Argumentation is a wide area with numerous applications, in which one has to face different classes of problems. Frameworks of a given type can be seen as tools to model particular issues and concepts, which on one side gives us more insight into how to approach the problems, but on the other side affects the framework's design. When facing such an amount of available structures, it is only natural to ask whether one can translate one framework into another and, if yes, what would be the best way to do this. However, as framework intertranslatability is of both practical and theoretical value, what we consider the "best" depends on how we intend to use the transformations.

The ability to transform one framework into another can be used in designing various argumentation–based applications. The majority of the existing structures does not have a dedicated solver, thus a translation into one that does [45, 49] is of practical use. In such a case we would be interested in an algorithm that preserves the behavior of the semantics, however, it is the efficiency of the translation that is vital. We not only want the algorithm to have a reasonable running time; we would also like the resulting framework to retain a size that can be managed by its solver. Moreover, if our purpose is to solve a variety of problems for which different frameworks are suitable, translations would allow us to choose the most adequate one to work "in the background". Finally, what is also important to notice is the need for a translation to exist, even if its performance leaves a lot to be desired. A researcher would normally look for a translation that has certain properties, like faithfulness or exactness, or show that such a translation cannot exist. However, a programmer would still look for a working approach, even if it meant that it would not possess all of the valuable attributes.

An efficient translation is still desirable from a research perspective. However, the behavior of the semantics and what structural changes a framework has to undergo are now more important than in the practical perspective. The answers given by a translated framework can coincide with the initial ones, or they themselves need to be transformed back, be it just by removing auxiliary elements or reinterpreting the target arguments in terms of the source arguments. Here is where properties such as faithfulness or exactness come into play. Every argumentation framework differs from another by some element and a translation gives us an insight into how this element works and how it can or cannot be simulated by other ones. We can try to transform one form of support into another, support into attack, preference into an argument and so on. Moreover, we would like for a given relation and its associated arguments to be translated in a "recognizable" way. By this we understand that if we need auxiliary or meta arguments, we would expect them to have some meaning that would explain the cause for its creation, or that framework elements are not removed without a reason. All of those structural changes can connect or detach parts of the framework in question. Depending on how intrusive the modifications are, propagating the change in the source structure to the target one can become nearly impossible without repeating the translation altogether. Therefore, the efficiency, semantics behavior and structural changes can finally be used to compare both translations and different argumentation frameworks and to analyze the frameworks' expressive power.

Our work is meant to answer these questions as much as possible. The aim of this report is to provide the translations between the aforementioned relation–introducing frameworks and provide their analysis in terms of functional, syntactical, semantical and complexity properties. In doing do, we create an in–depth compendium consisting of almost ninety translations. The frameworks with support, such as bipolar argumentation frameworks BAFs [28,30], argumentation frameworks with necessities AFNs [69], evidential argumentation systems EASs [73,78] and abstract dialectical frameworks ADFs [21,23,76], are of particular interested to us. Although some methods have already been researched, especially transformations from and into the Dung's framework [9,18,21,28,64,68,69,73], there are less results concerning moving between AF generalizations [30, 73, 78]. Moreover, the existing methods for frameworks with support can be mostly classified as coalition approaches, i.e. the arguments in the target structure represent sets of arguments of the source one, usually connected by support. In our research we propose alternative transformations, based on the properties of the frameworks in question and the analysis carried out in [28, 30] – the attack propagation and defender translations. The first one simulates the behavior of support by combinations of attacks, while the latter transforms support to defense with the use of auxiliary arguments. We will show that similar principles can be used for e.g. transforming a group attack into a binary attack.

We will start the report by recalling all the required argumentation frameworks in Section 2. We will also recall the research on the semantics realizability [37, 43] in Dung's frameworks, which will prove useful in showing what translations from other frameworks into AFs are possible and which are not. In Section 4 we will introduce various normal forms and subclasses of the studied frameworks. Section 3 will be devoted to formalizing the concept of a translation and its properties and introducing a classification of our approaches. The rest of the work will consist of translations themselves, one section per framework, and pointers for future research. We close the introduction with providing pointers to appropriate translations in Table 1. The abbreviations of the frameworks are as follows:

- AF – Dung's Framework
- SETAF – Framework for Arguing with Sets of Attacking Arguments
- AFRA – Argumentation Framework with Recursive Attacks
- EAF – Extended Argumentation Framework
- EAFC – Extended Argumentation Framework with Collective Defense Attacks
- BAF – Bipolar Argumentation Framework
- AFN – Argumentation Framework with Necessities
- EAS – Evidential Argumentation System
- ADF – Abstract Dialectical Framework

The table should be read in the following way; frameworks in the first column stand for source frameworks, while the ones in the first row are the target frameworks. A given cell contains a number that is either associated with an appropriate translation, or with the relevant section if a translation is not defined or the issue is more complicated (in this case, the number is preceded by Sec.).

Table 1: Translation Chart of Argumentation Frameworks

<table>
<tr><th></th><th>AF</th><th>SETAF</th><th>AFRA</th><th>EAF</th><th>EAFC</th><th>BAF</th><th>AFN</th><th>EAS</th><th>ADF</th></tr>
<tr><td>AF</td><td>x</td><td>17</td><td>18</td><td>19</td><td>Sec. 5.3</td><td>20</td><td>21</td><td>22</td><td>23</td></tr>
<tr><td>SETAF</td><td>25, 26, 27</td><td>x</td><td colspan="3">Sec. 6.6</td><td>Sec. 6.2</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>AFRA</td><td>32, 33</td><td>Sec. 7.6</td><td>x</td><td colspan="2">Sec. 7.2</td><td>34</td><td>35</td><td>Sec. 7.6</td><td>Sec. 7.5</td></tr>
<tr><td>EAF</td><td>38</td><td>40</td><td>42</td><td>x</td><td>36</td><td>Sec. 8.7</td><td>43, 44</td><td>Sec. 8.7</td><td>47, 48</td></tr>
<tr><td>EAFC</td><td>39</td><td>41</td><td>Sec. 8.4</td><td>37</td><td>x</td><td>Sec. 8.7</td><td>45, 46</td><td>Sec. 8.7</td><td>49, 50</td></tr>
<tr><td>BAF</td><td>51,52, 53, 54, 55, 57, 58</td><td colspan="4">Sec. 9.5</td><td>x</td><td>59</td><td>60</td><td>Sec. 9.4</td></tr>
<tr><td>AFN</td><td>61, 62</td><td>63, 65, 66</td><td>Sec. 10.7</td><td colspan="2">Sec. 10.3</td><td>67</td><td>x</td><td>68</td><td>69, 70</td></tr>
<tr><td>EAS</td><td>71</td><td>72, 73, 75, 76</td><td colspan="3">Sec. 11.6</td><td>77</td><td>78,79</td><td>x</td><td>80, 81</td></tr>
<tr><td>ADF</td><td>82, 83, 85, 86</td><td>87, 88</td><td colspan="4">Sec. 12.4</td><td>89</td><td>Sec. 12.4</td><td>x</td></tr>
</table>

2 Background

In this section we will recall the argumentation frameworks used in our study. We will go through them one by one, starting from the original framework by Dung (AF for short) [36], through the set attack (SETAF) [68], recursive attack (AFRA) [9], to bipolar ones (BAF, AFN, EAS) [30, 69, 72] and abstract dialectical framework (ADF) [23]. Moreover, we will also recall the extended argumentation framework (EAF) and its collective version (EAFC) [62, 66], which use higher–level attacks to express preferences. In many cases we will provide additional results and analysis that will become useful in the later parts of this work. At the end of this section we will also briefly go through the research on signatures in Dung's framework, i.e. descriptions of sets of arguments that can be jointly accepted as a collection of extensions under certain semantics. However, before we start, we would like to introduce some basic concepts and notation.

Argumentation frameworks, along with their associated semantics, are the basic tools for abstract argumentation. The framework itself is primarily built from arguments and various relations between them, however, it can also contain information such as preferences, probabilities, labels and more. While the framework represents a given problem, the semantics are meant to "solve it". A semantics encompasses what we consider a rational opinion; for example, we would like to be able to defend our position and prefer not to contradict ourselves. This is grasped by admissibility and conflict–freeness. We might also take a stand in which we can provide a counterargument to whatever our opponent says, which brings us to the notion of stability. Semantics can return the "answer" in various formats, such as sets of arguments, three or four–valued labelings, rankings and more [1, 7, 56]. We will be interested in the first two types and thus focus on extension–based (for sets) and labeling–based (for three–valued labelings) semantics in this work.

There are many abstract argumentation frameworks available [22], far more than we can analyze in this work. We will only focus on the aforementioned structures and refer to them by their abbreviations. By a framework type we will understand $T \in \{AF, SETAF, AFRA, EAF, EAFC, BAF, AFN, EAS, ADF\}$. Every argumentation framework has at least one element, be it syntactical or semantical, distinguishing it from any other structure. However, they all have one single thing in common: a set of (abstract) arguments. All other elements can be defined differently, including even the most basic attack relation from Dung's framework [36]. Thus, from now on we will assume that we are working with a domain of abstract arguments $\mathcal{U}$, unless stated otherwise. Although it is typical to limit oneself to e.g. a countably infinite domain [37], it is often the case that the translation does not preserve countability. In other words, even if we make certain assumptions for the source domain, the target domain might not satisfy them. By $Fr^T_{\mathcal{U}}$ we will understand the collection of all frameworks of type T s.t. its set of arguments is a subset of $\mathcal{U}$. Our focus will be primarily on those frameworks in which the set of arguments is finite, though please note that many of the results can be applied to the infinite cases as well.

Apart from speaking about the argument domain, we will also need the semantics domain. It will define how the answers produced by a given semantics look like. Such a domain can be a power set of the argument domain, typically used when we work with extension–based semantics of most of the argumentation frameworks. It can also be a collection of three–valued interpretations on argument subsets of $\mathcal{U}$ for the labeling–based semantics. However, please note that it might not necessarily depend on arguments only and can contain other framework elements, such as attacks in case of AFRAs (see Section 2.1.3). Consequently, the semantics domain ought to be stated explicitly, and we will speak about the general domain $\mathcal{U}_\sigma$ for a given semantics σ. In certain cases it might be more natural to look at the extensions or labelings w.r.t. the framework in question and take into account a limited domain U_σ^{Fr}, where Fr is our framework of interest. For example, if we speak about extension–based semantics and the set of arguments is A, then U_σ^{Fr} can be 2^A. The union of all such limited domains of all frameworks of a given type will give us the general domain. In this light, a given semantics defined for frameworks of a given type, can be seen as a function assigning to a particular framework a number of answers it can produce, i.e. a subset of the limited (or general) semantics domain. We will simply write $\sigma(Fr)$ to denote the set of all extensions (labelings) produced by the semantics σ for framework Fr. Of course, not every such subset of the domain will actually correspond to the collection of all extensions or labelings a semantics can produce. The set of all such subsets that are, in fact, possible, will form a semantics signature and will be described in Section 2.4 for the Dung's framework. We can now proceed with introducing the argumentation frameworks we will be working with.

2.1 Conflict–Based Argumentation Frameworks

2.1.1 Binary Conflict: Dung's Argumentation Framework

Let us now recall one of the most prominent structures in abstract argumentation – the abstract argumentation framework by Dung [36] – and its semantics (a more detailed analysis can be found in [7]). The framework consists of the set of arguments and the set of binary attacks between them:

Definition 2.1. A **Dung's abstract argumentation framework** (AF for short) is a pair $F = (A, R)$, where A is a set of **arguments** and $R \subseteq A \times A$ represents an **attack** relation.

We say that an argument a attacks b iff aRb. Based on conflicts, we can derive a certain indirect, positive relation between arguments, referred to as defense. To put it simply, an argument is defended if all of its attackers are in turn attacked. By combining defense, attacks and various notions of maximality, we obtain a number of semantics for AFs:

Definition 2.2. Let $F = (A, R)$ be a Dung's framework. An argument $a \in A$ is **defended** by a set $E \subseteq A$ in F[1], if for every $b \in A$ s.t. $(b, a) \in R$, there exists $c \in E$ s.t. $(c, b) \in R$. A set $E \subseteq A$ is:

[1]Defense is often also referred to as acceptability: we say that a is acceptable w.r.t. E if E defends a.

- **conflict–free** in F iff for each $a, b \in E$, $(a, b) \notin R$.
- **admissible** in F iff it is conflict–free in F and defends in F all of its members.
- **preferred** in F iff it is maximal w.r.t. set inclusion admissible in F.
- **complete** in F iff it is admissible in F and all arguments defended by E are contained in E.
- **stable** in F iff it is conflict–free in F and for each $a \in A \setminus E$ there exists an argument $b \in E$ s.t. $(b, a) \in R$.

We will often abbreviate the semantics with cf, adm, $pref$, $comp$ and stb when using the functional representation. Please note that the stable semantics is somewhat different from the rest of the approaches in the sense that depending on the given framework, it might not produce any extensions. This problem was addressed by introducing approaches focused maximizing range, i.e. the amount of arguments covered by the extension both in terms of acceptance and rejection [27]. Although we will not be dealing much with these semantics in this work, the idea of range will be useful to us.

Definition 2.3. Let $F = (A, R)$ be a Dung's framework and $E \subseteq A$ a set of arguments. The **set of arguments attacked by** E is defined as $E^+ = \{a \in A \mid \exists e \in E, (e, a) \in R\}$ and the set of attackers of E is $E^- = \{a \in A \mid \exists e \in E, (a, e) \in R\}$. The set $E^{Ran} = E^+ \cup E$ is the **range** of E in F.

Definition 2.4. Let $F = (A, R)$ be a Dung's framework. A set of of arguments $E \subseteq A$ is a **semi–stable** extension of F iff it is range maximal w.r.t. set inclusion complete extension of F. [2]

To every argument in the framework which is not attacked by any other argument we will refer as *initial*.

We close the list with the grounded semantics, abbreviated grd. It basically represents the knowledge that we can only build from the initial arguments, i.e. starting with an empty set we add all elements defended by the set and continue until nothing more is added. The formal definition is given by the means of the characteristic function of F:

Definition 2.5. Let $F = (A, R)$ be a Dung's framework. The **characteristic function** of F, $\mathcal{F}_F : 2^A \rightarrow 2^A$, is defined as: $\mathcal{F}_F(E) = \{a \mid a \text{ is defended by } E \text{ in } F\}$. The **grounded** extension of F is the least fixed point of $\mathcal{F}_F$.

Also other semantics, in particular admissible and complete, can be defined in terms of the characteristic operator:

Lemma 2.6. *Let $F = (A, R)$ be a Dung's framework and $E \subseteq A$ be a conflict–free set of F. E is admissible in F iff $E \subseteq \mathcal{F}_F(E)$. E is complete in F iff $E = \mathcal{F}_F(E)$.*

[2] It is easy to see that stable extension is a special case when $E^+ = A \setminus E$.

Please note that there is also an alternative, iterative way to compute the grounded extension, which can be used also in frameworks in which e.g. the operator is no longer monotonic:

Proposition 2.7. *Let $F = (A, R)$ be a Dung's framework. The unique* ***grounded extension*** *of F is defined as the outcome E of the following "algorithm". Let us start with $E = \emptyset$:*

1. *put each argument $a \in A$ which is not attacked in F into E; if no such argument exists, return E, and*
2. *remove from F all (new) arguments in E and all arguments attacked by them (together with all adjacent attacks) and continue with Step 1.*

We would also like to recall several important lemmas and theorems from the original paper on AFs [36]. The so–called Fundamental Lemma is as follows:

Lemma 2.8. ***Dung's Fundamental Lemma*** *Let $F = (A, R)$ be a Dung's framework, $E \subseteq A$ an admissible extension of F and a and b arguments that are defended by E in F. Then $E' = E \cup \{a\}$ is admissible in F and b is defended by E' in F.*

The next two theorems show some of the relations between the existing semantics.

Theorem 2.9. *Let $F = (A, R)$ be a Dung's framework. Every stable extension of F is a preferred extension, but not vice versa.*

Theorem 2.10. *Let $F = (A, R)$ be a Dung's framework. The following holds:*

- *every preferred extension of F is a complete extension of F, but not vice versa.*
- *the grounded extension of F is the least w.r.t. $\subseteq$ complete extension of F.*
- *the complete extensions of F form a complete semilattice w.r.t. set inclusion.* [3]

Although in general the available semantics that satisfy the completeness requirements produce different extensions, various AF subclasses have been identified on which at least some of them coincide. Among the strongest of them is the well–founded class, already described in the original paper [36]:

Definition 2.11. Let $F = (A, R)$ be a Dung's framework. F is **well–founded** iff there exists no infinite sequence of arguments $a_0, a_1, ..., a_n, ...$ s.t. for each i, a_{i+1} attacks a_i.

Theorem 2.12. *Every well–founded Dung's argumentation framework has exactly one complete extension which is grounded, preferred and stable.*

We will now show the extensions of the presented semantics on an example.

[3] A partial order $(B, \leq)$ is a complete semilattice iff each nonempty subset of B has a glb and each increasing sequence of B has a lub.

Example 1. Consider the Dung's framework $F = (A, R)$ with $A = \{a, b, c, d, e\}$ and the attack relation $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$, as depicted in Figure 1. It has eight conflict–free extensions in total: $\{a, c\}$,$\{a, d\}$, $\{b, d\}$, $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$ and $\emptyset$. As b is attacked by an unattacked argument, it cannot be defended against it and will not be in any admissible extension. From this $\{a, c\}$, $\{a, d\}$ and $\{a\}$ are complete. We end up with two preferred extensions, $\{a, c\}$ and $\{a, d\}$. However, only $\{a, d\}$ is stable, and $\{a\}$ is the grounded extension.

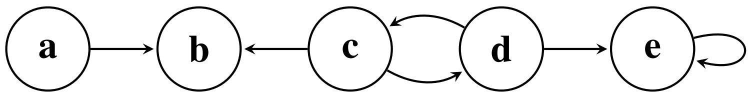


Figure 1: Sample Dung's framework

The Dung's framework has also more families of semantics, one of them being the labeling–based [7, 25]. Instead of returning sets of accepted arguments, they produce mappings in which an argument can be assigned status in, out, or $undec$:

Definition 2.13. Let $F = (A, R)$ be a Dung's framework. A three–valued labeling is a total function $Lab : A \to \{in, out, undec\}$[4]. An in–labeled argument is **legally in** iff all its attackers are labeled out. An out–labeled argument is **legally out** iff at least one its attacker is labeled in. An $undec$–labeled argument is **legally undec** iff not all of its attackers are labeled out and it does not have an attacker that is labelled in.

By $in(Lab), out(Lab)$ and $undec(Lab)$ we will denote the arguments mapped respectively to in, out and $undec$ by Lab. We will also write a labeling Lab as a triple (I, O, U), where $I = in(Lab)$, $O = out(Lab)$ and $U = undec(Lab)$.

Definition 2.14. Let $F = (A, R)$ be a Dung's framework and Lab a three–valued labeling on A. Lab is:

- **conflict–free** in F iff it holds that if $a \in A$ is labeled in. then none of its attackers is labeled in, and if it is labeled out, then it has at least one attacker labeled in.
- **admissible** in F iff each in–labeled argument is legally in and each out–labeled argument is legally out.
- **complete** in F if it is admissible in F and every $undec$–labeled argument is legally $undec$.
- **preferred** in F if it is complete in F and the set of arguments labeled in is maximal w.r.t. set inclusion.
- **grounded** in F if it is complete in F and the set of arguments labeled in is minimal w.r.t. set inclusion.

[4] Sometimes the $\mathtt{t}, \mathbf{f}$ and $\mathbf{u}$ notation is also used.

- **stable** in F if it is complete in F and the set of elements mapped to $undec$ is empty.

We will shorten these semantics by adding the lab– prefix to the usual abbreviations.

The properties of the labeling–based semantics and their correspondence to the extension–based family have already been studied in [7, 25].

Theorem 2.15. *Let $F = (A, R)$ be a Dung's framework and $E \subseteq A$ be a σ–extension of F, where $\sigma \in \{$admissible, complete, grounded, preferred, stable$\}$. Then $(E, E^+, A \setminus (E \cup E^+))$ is a σ–labeling of F.*

Let Lab be a σ–labeling of F, where $\sigma \in \{$admissible, complete, grounded, preferred, stable$\}$. Then $in(Lab)$ is a σ–extension of F.

Remark. Depending on the semantics, there can be more than one labeling corresponding to a given extension. Let E^- be the set of arguments that attack E. Obviously, E defends its members iff $E^- \subseteq E^+$. Therefore, for a labeling to be admissible it suffices that the set of out arguments contains E^-; on the other hand, due to legality it cannot map more than E^+. This gives us a certain freedom in assignments. On the other hand, for example the stable extensions are in one–to–one correspondence with the stable labelings.

Theorem 2.16. *Let $F = (A, R)$ be an AF. The following statements are equivalent:*

- *Lab is a grounded labeling of F.*
- *Lab is a complete labeling of F where $in(Lab)$ is minimal w.r.t. $\subseteq$ among all complete labelings of F.*
- *Lab is a complete labeling of F where $out(Lab)$ is minimal w.r.t. $\subseteq$ among all complete labelings of F.*
- *Lab is a complete labeling of F where $undec(Lab)$ is maximal w.r.t. $\subseteq$ among all complete labelings of F.*

Theorem 2.17. *Let $F = (A, R)$ be an AF. The following statements are equivalent:*

- *Lab is a preferred labeling of F.*
- *Lab is a complete labeling of F where $in(Lab)$ is maximal w.r.t. $\subseteq$ among all complete labelings of F.*
- *Lab is a complete labeling of F where $out(Lab)$ is maximal w.r.t. $\subseteq$ among all complete labelings of F.*

Example 2. Let us come back to the framework $F = (A, R)$ with $A = \{a, b, c, d, e\}$ and the attack relation $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$, as depicted in Figure 1 and described in Example 1. It had six admissible extensions, namely $\{a, c\}, \{a, d\}, \{a\}, \{c\}, \{d\}$ and $\emptyset$. Let us focus on the first one. We can observe that the labeling $(\{a, c\}, \emptyset, \{b, d, e\})$ is not admissible, since c has an attacker not mapped to out;

however, $(\{a, c\}, \{d\}, \{b, e\})$ already meets the requirements. This is not the only labeling corresponding to $\{a, c\}$; also $(\{a, c\}, \{b, d\}, \{e\})$ is admissible. It is also the complete labeling associated with $\{a, c\}$; the first one does not qualify due to b not being legally undec. We can observe that due to e being mapped to undec, this complete labeling cannot be stable. On the other hand, the assignment $(\{a, d\}, \{b, c, e\}, \emptyset)$ related to $\{a, d\}$ meets the stability requirements. Finally, $(\{a\}, \{b\}, \{c, d, e\})$ is our grounded labeling. Please observe we cannot omit assigning *out* to b, as it would render the labeling not complete.

2.1.2 Set Conflict: Framework for Arguing with Sets of Attacking Arguments

Although AFs are quite powerful tools, permitting only binary conflict can be limiting and cannot model certain situations in a natural manner. This has led to the development of a number of more general structures. It can often be the case that a single argument might not be enough to carry out an attack on another argument. For example, a piece of evidence becomes incriminating only when put in bigger context, while alone it might not point to anything specific. In order to grasp such problems, a framework with set form of conflict was developed [68]:

Definition 2.18. A **framework for arguing with sets of attacking arguments** (SETAF or CAF for short) is a pair $SF = (A, R)$, where A stands for the set of **arguments**, and $R \subseteq (2^A \setminus \emptyset) \times A$ represents the (set) **attack** relation.

It might seem that this framework does not modify the Dung's framework very significantly. After all, frameworks with support appear to be much more complicated. As we will also see below, the semantics are easily shifted into this new setting as well – it is basically only the notion of attack that now takes sets and not just single arguments into account. However, as we will see in Section 6.1, SETAFs go beyond Dung's frameworks and their semantics can give us a set of extensions that cannot be handled by the Dung's framework without the use of auxiliary arguments. We will also show that for certain types of translations, SETAFs are more natural targets than AFs (see Sections 10.2 and 11.2). Consequently, this structure should not be underestimated.

The semantics from Dung's setting carry over naturally to the new one. Please note that in what follows, by saying that a set of arguments E attacks an argument y we will understand that there exists a set $E' \subseteq E$ s.t. $E'Ry$. All other notions follow accordingly:

Definition 2.19. Let $SF = (A, R)$ be a SETAF. A set of arguments $E \subseteq A$ **defends** an argument $a \in A$ in SF iff for every set of arguments B attacking a, E attacks at least one member of B. Then set $E \subseteq A$ is:

- **conflict–free** in SF if it does not attack itself, i.e. there is no $s \in E$ s.t. E attacks s.
- **admissible** in SF iff it is conflict–free in SF and defends in SF all of its members.
- **preferred** in SF iff it is maximal w.r.t. set inclusion admissible in SF.

- **complete** in SF iff it is admissible in SF and all arguments defended by E in SF are contained in E.
- **stable** in SF iff it is conflict–free in SF and for all $a \in A \setminus E$, there is a $E' \subseteq E$, s.t.$(E', a) \in R$.

Like in the Dung's setting, grounded semantics is based on a characteristic function. The operator also preserves the relation to admissible and complete semantics.

Definition 2.20. Let $SF = (A, R)$ be a SETAF. The **characteristic function** of SF is a function $\mathcal{F}_{SF} : 2^A \rightarrow 2^A$ defined as $\mathcal{F}_{SF}(E) = \{a \mid a \text{ is defended by } E \text{ in } SF\}$. The **grounded** extension of SF is the least fixed point of $\mathcal{F}_{SF}$.

Lemma 2.21. *Let $SF = (A, R)$ be a SETAF and $E \subseteq A$ a conflict–free extension of SF. E is admissible in SF iff $E \subseteq \mathcal{F}_{SF}(E)$. E is complete in SF iff $E = \mathcal{F}_{SF}(E)$.*

All basic properties of the Dung's setting carry over to the SETAFs.

Lemma 2.22. ***SETAF Fundamental Lemma*** *Let $SF = (A, R)$ be a SETAF, $E \subseteq A$ an admissible extension of SF and a and b arguments that are defended by E in SF. Then $E' = E \cup \{a\}$ is admissible in SF and b is defended by E' in SF.*

Theorem 2.23. *Let $SF = (A, R)$ be a SETAF. Every stable extension of SF is a preferred extension of SF, but not vice versa.*

Theorem 2.24. *Let $SF = (A, R)$ be a SETAF. The following holds:*

- *every preferred extension of SF is a complete extension of SF, but not vice versa.*
- *the grounded extension of SF is the least w.r.t. $\subseteq$ complete extension of SF.*
- *the complete extensions of SF form a complete semilattice w.r.t. set inclusion.*

Finally, similarly as in AFs, we will introduce the set of attacked arguments in SETAFs. We can observe that defense can be easily redefined with this notion, which will come in handy when proving translations from SETAFs to certain other frameworks.

Definition 2.25. Let $SF = (A, R)$ be a SETAF and $E \subseteq A$ a set of arguments. The **discarded set** of E is defined as $E^+ = \{a \mid a \in A, \exists B \subseteq E, (B, a) \in R\}$. The **range** of E is $E^{Ran} = E \cup E^+$.

Lemma 2.26. *Let $SF = (A, R)$ be a SETAF and $E \subseteq A$ a conflict–free extension. E is admissible iff for every set $B \subseteq A$ s.t. $\exists a \in E, (B, a) \in R$, it holds that $B \cap E^+ \neq \emptyset$.*

Example 3. Let us consider the SETAF $SF = (A, R)$ with $A = \{a, b, c, d, e\}$ and the attack relation $R = \{(\{a\}, c), (\{a\}, b), (\{b\}, a), (\{c\}, d), (\{e\}, a), (\{b, d\}, e)\}$, as depicted in Figure 2. The admissible extensions of this framework are $\emptyset$, $\{b\}$, $\{b, c\}$, $\{c, e\}$ and $\{b, c, e\}$. Only $\emptyset$ and $\{b, c, e\}$ are complete. The grounded extension is $\emptyset$, while $\{b, c, e\}$ is both preferred and stable.

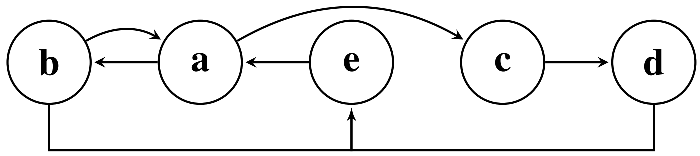


Figure 2: Sample SETAF

2.1.3 Recursive Binary Conflict: Argumentation Framework with Recursive Attacks

While the majority of the available generalizations of the Dung's framework focuses on augmenting the original structure with further relations or properties, the argumentation framework with recursive attack (AFRA for short), just like SETAF, studies the concept of attack further. In [8, 9] the authors argue that we should not only be able to attack arguments, but the attacks themselves as well. In doing so, they raise the relation to the level of arguments, thus making them appear explicitly in the extensions under a given semantics[5]. Let us now formally define the framework and move on to the semantics.

Definition 2.27. An **argumentation framework with recursive attacks** (AFRA) is a pair (A, R) where A is a set of **arguments** and R is a set of **attacks**, namely pairs (a, X) s.t. $a \in A$ and $X \in A \cup R$.

Given an attack $\alpha = (a, X) \in R$, we will say that a is the source of α, denoted as $src(\alpha) = a$ and X is the target of α, denoted as $trg(\alpha) = X$. Due to the new structure of attacks, one can introduce several notions of defeat:

Definition 2.28. Let $FR = (A, R)$ be an AFRA, $V \in R$ and $W \in A \cup R$. V **directly defeats** W in FR iff $W = trg(V)$.

Definition 2.29. Let $FR = (A, R)$ be an AFRA and $V, W \in R$. If V directly defeats $src(W)$ in FR, then V **indirectly defeats** W in FR.

Definition 2.30. Let $FR = (A, R)$ be an AFRA, $V \in R$ and $W \in A \cup R$. V **defeats** W in FR iff V directly or indirectly defeats W in FR. [6]

The definition of acceptability is very similar to the one in the Dung's framework. It is naturally extended by defending not only arguments, but also attacks.

Definition 2.31. Let $FR = (A, R)$ be an AFRA, $E \subseteq A \cup R$ and $W \in A \cup R$. W is **acceptable** w.r.t. E in FR iff $\forall Z \in R$ s.t. Z defeats W in FR, $\exists V \in E$ s.t. V defeats Z in FR.

[5] Similar concept of attack–based semantics in the Dung's setting, but without recursion, can be found in [88].

[6] Please note that this general definition of defeats is for the ease of use and the usual restrictions still apply. For example there will be no indirect defeat in case W is an argument, not a relation.

Lemma 2.32. *Let $FR = (A, R)$ be an AFRA and $E \subseteq A \cup R$. If an attack $V \in R$ is acceptable w.r.t. E in FR, then $src(V)$ is acceptable w.r.t. E in FR as well.*

With these definitions, we can now move on to describing the semantics developed for AFRA so far. Please note that in this setting, extensions no longer consist of arguments only, but can also contain attacks.

Definition 2.33. Let $FR = (A, R)$ be an AFRA and $E \subseteq A \cup R$. E is **conflict–free** in FR iff $\nexists V, W \in E$ s.t. V defeats W in FR.

As a side–effect of this formulation, every set consisting of arguments only is conflict–free. Attacks need to be taken explicitly into account in order to "break" conflict–freeness.

Definition 2.34. Let $FR = (A, R)$ be an AFRA. A set $E \subseteq A \cup R$ is:

- an **admissible extension** of FR iff it is conflict–free in FR and each element of E is acceptable w.r.t. E in FR.
- a **preferred extension** of FR is a maximal w.r.t. set inclusion admissible extension of FR.
- a **complete extension** of FR iff it is admissible in FR and contains every element of $A \cup R$ that is acceptable w.r.t. E in FR.
- a **stable extension** of FR iff it is conflict–free in FR and $\forall V \in (A \cup R) \setminus E$, $\exists W \in E$ s.t. W defeats V in FR.

The grounded extension is again defined via the characteristic function, which is naturally shifted to the new setting:

Definition 2.35. Let $FR = (A, R)$ be an AFRA. The **characteristic function** of FR $\mathcal{F}_{FR} : 2^{A \cup R} \to 2^{A \cup R}$ is defined as $\mathcal{F}_{FR}(E) = \{V \mid V$ is acceptable w.r.t. E in $FR\}$. The **grounded extension** of FR is the least fixed point of $\mathcal{F}_{FR}$.

Please note that the discarded set can also be defined in the AFRA setting. However, unlike in AFs and SETAFs, it consists of both arguments and attacks, not arguments only:

Definition 2.36. Let $FR = (A, R)$ be an AFRA and $E \subseteq A$ a set of arguments. The **discarded set** of E in FR is defined as $E^+ = \{a \mid a \in A \cup R, \exists b \in E$ s.t. b defeats $a\}$.

Finally, we can observe that all the usual relations between the semantics from the Dung's setting carry over to AFRAs:

Lemma 2.37. ***AFRA Fundamental Lemma.*** *Let $FR = (A, R)$ be an AFRA, $E \subseteq A \cup R$ an admissible extension of FR and $V, V' \in A \cup R$ elements acceptable w.r.t. E in FR. Then $E' = E \cup \{V\}$ is admissible in FR and V' is acceptable w.r.t. E' in FR.*

Theorem 2.38. *Let $FR = (A, R)$ be an AFRA. The following holds:*

- *the grounded extension of FR is the least w.r.t. set inclusion complete extension of FR.*
- *every preferred extension of FR is complete in FR, but not vice versa.*
- *every stable extension of FR is preferred in FR, but not vice versa.*

Example 4. (taken from [9]) Let us consider the AFRA $FR = (A, R)$ where $A = \{a, b, c, d, e, f, g\}$ and $R = \{\alpha, \beta, \gamma, \delta, \varepsilon, \eta, \zeta, \vartheta, \iota, \kappa\}$, with $\alpha = (a, b)$, $\beta = (b, \alpha)$, $\gamma = (c, \alpha)$, $\delta = (c, d)$, $\varepsilon = (e, \delta)$, $\eta = (d, \varepsilon)$, $\zeta = (a, f)$, $\vartheta = (f, a)$, $\iota = (f, g)$ and $\kappa = (g, g)$. The framework has in total 212 admissible extensions and we will not list them here. The complete extensions are $\{b, c, e, \beta, \gamma\}$ $\{b, c, e, f, \beta, \gamma, \vartheta, \iota\}$ and $\{a, b, c, e, \beta, \gamma, \zeta\}$. The first one is grounded, while the latter two are preferred. None of them is stable, since $\delta, \varepsilon, \eta$ and d are not defeated by any of the extensions.

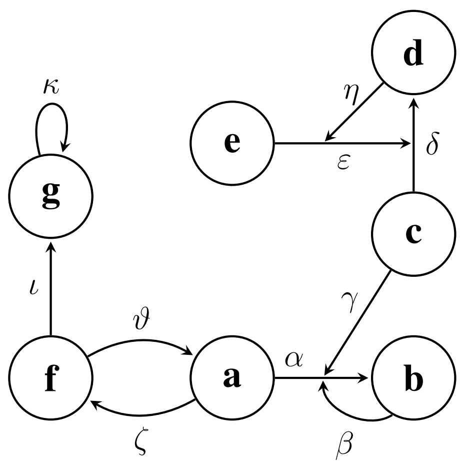


Figure 3: Sample AFRA

2.1.4 Conflict as Preference: Extended Argumentation Frameworks

The last framework we will consider in this section on conflict–based frameworks is the extended argumentation framework EAF [15, 39, 41, 61, 63–66]. It extends the classical Dung's framework by introducing the notion of defense attacks, which occur between arguments and standard conflicts. Their initial function was to "override" a given attack due to the target's importance. In practical situations we might sometimes be forced to accept facts that are in conflict. For example, a hospital patient needs to undergo a certain medical procedure **p**. It is, however, expensive **e**. We can observe that in this scenario, **e** is a counterargument for **p**. Let us now assume it turns out that the procedure is necessary to save the life of the patient **s**. Argument **s** as such is not in conflict with **e**; it is both true that the procedure is expensive and that it is necessary. A more accurate way to model this situation is by allowing **s**, due to its importance, to cancel the attack by **e** on **p**. In this case the patient would undergo the surgery, despite its price. The additional benefit of having arguments that express the preferences is the fact that we can now argue not only about the arguments, but also about their importance, as opposed to assuming a predefined preference assignment to which every party has to agree. If, for some reason, it turned out that the procedure is in fact not necessary to save his life and there is a cheaper alternative, we would have a counterargument for **s** and a choice not to perform the initial procedure after all. We can now proceed with the formal introduction of the extended argumentation framework.

Definition 2.39. The **extended argumentation framework** (EAF for short) is a tuple $EF = (A, R, D)$, where A is a set of **arguments**, $R \subseteq A \times A$ is the **attack** relation, $D \subseteq A \times R$ is the **defense attack** relation and if $(x, (y, z)), (x', (z, y)) \in D$, then $(x, x'), (x', x) \in R$.

We can observe that the EAF definition includes the symmetric attacks between (preference) arguments induced by the symmetric conflicts they can override. This restriction was introduced based on the preference interpretation of the defense attack. If an argument c overrides attack from a to b, claiming e.g. that b is more preferred, and c' overrides attack from b to a, claiming that a is more preferred, then introducing a symmetric conflict between c and c' is a reasonable decision. We will discuss this restriction further in the next section.

Although there are certain similarities, please note there are notable differences between the defense attacks and recursive attacks from AFRAs. The defense attacks exist alongside the standard conflicts and can be directed only at them. The recursive attacks replace the standard ones and can be of arbitrary depth. Moreover, since they are treated on the same level as arguments, they appear in the extensions and are one of the reasons why there are also significant semantical differences between EAF and AFRA.

Since now attacks can be overridden by other attacks, EAFs adopt the notion of "defeats" to denote the successful ones:

Definition 2.40. Let $EF = (A, R, D)$ be an extended argumentation framework and $E \subseteq A$ a set of arguments. An argument a **defeats** an argument b w.r.t. E in EF, denoted defeats$_E$, iff $(a, b) \in R$ and there is no argument $c \in E$ s.t. $(c, (a, b)) \in D$. If a defeats$_E$ b in EF and b does not defeat$_E$ a in EF, then a **strictly defeats**$_E$ b in EF.

From this also follows a simple proposition:

Proposition 2.41. *Let $EF = (A, R, D)$ be an extended argumentation framework, $E \subseteq A$ a set of arguments and $a, b \in A$. If a defeats$_E$ b in EF, then for every $E' \subseteq E$, a defeats$_{E'}$ b in EF.*

We can now continue with the semantics. As seen in the example, preferences can overrule certain attacks and thus make it possible for us to accept arguments that appear to be in conflict. Thus, simply using Dung's conflict–freeness in EAFs is insufficient.

Definition 2.42. Let $EF = (A, R, D)$ be an EAF. A set of arguments $E \subseteq A$ is **conflict–free** in EF iff for every $a, b \in E$, if $(a, b) \in R$, then $(b, a) \notin R$ and $\exists c \in E$ s.t. $(c, (a, b)) \in D$.

Although conflict–freeness implies that there are no defeats in the set, a set without defeats is not necessarily conflict–free, as observed in Example 5.

Proposition 2.43. *Let $EF = (A, R, D)$ be an EAF and $E \subseteq A$ a conflict–free set of EF. Then for any $a, b \in E$, a does not defeat$_E$ b.*

Example 5 ([62])**.** Let $(\{a, b, c, d\}$, $\{(a, b)$, (b, a), (c, d), $(d, c)\}$, $\{(a, (d, c))$, $(b, (c, d))$, $(c, (b, a))$, $(d, (a, b))\})$ be the EAF depicted in Figure 4. We can observe that if the symmetry requirement was dropped from conflict–freeness (i.e. the definition would boil down to defeats$_E$), then the set $\{a, b, c, d\}$ would be conflict–free, which was against the intuition of the authors [62].

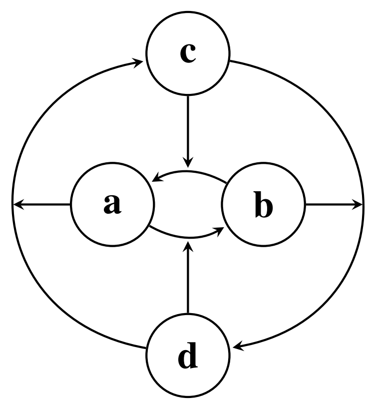


Figure 4: Sample EAF

Let us now continue with acceptability. In the Dung's framework, an attacked argument "stays" attacked, i.e. it will be defeated by any set containing its attacker. In EAFs, this is not the case – since attacks can be overridden, an argument can be "brought back", i.e. reinstated, which poses an additional challenge for defining defense.

Definition 2.44. Let $EF = (A, R, D)$ be an EAF and $E \subseteq A$. A set of pairs $R_E = \{(x_1, y_1), ..., (x_n, y_n)\}$ s.t. x_i defeats$_E$ y_i in EF is a **reinstatement set** on E for a defeat$_E$ by an argument c on argument b iff:

- $(c, b) \in R_E$,
- $\bigcup_{i=1}^{n} x_i \subseteq E$, and
- for every defeating$_E$ pair $(x, y) \in R_E$ in EF and every defense attack $(y', (x, y)) \in D$ on this pair, there is some x' s.t. $(x', y') \in R_E$.

The definition of acceptability is now as follows:

Definition 2.45. Let $EF = (A, R, D)$ be an EAF. An argument $a \in A$ is **acceptable** w.r.t. a set of arguments $E \subseteq A$ in EF iff for every argument $b \in A$ s.t. b defeats$_E$ a in EF, there is an argument $c \in E$ s.t. c defeats$_E$ b in EF and there is a reinstatement set on E for this defeat$_E$.

Please note that possessing a reinstatement set can also be verified with a defense attack analysis:

Theorem 2.46. *Let $EF = (A, R, D)$ be a finite EAF and $E \subseteq A$ a conflict–free extension of EF. If an argument $a \in E$ defeats$_E$ an argument $b \in A$, then there is no reinstatement set for this defeat$_E$ on E, iff there exists a sequence $((z_1, (x_1, y_1)), ..., (z_n, (x_n, y_n)))$ of distinct defense attacks from D s.t.*

- *there is an argument $g \in A$ s.t. $x_n = a$, $y_n = b$ and $z_n = g$,*
- *no two pairs (x_i, y_i) and (x_j, y_j) are the same for $i \neq j$,*
- *for every $(z_i, (x_i, y_i))$ where $1 < i \leq n$, either no argument h in E defeats$_E$ z_i or for every such defeat, there exists an argument $l \in A$ s.t. $(l, (h, z_i)) \in \{(z_1, (x_1, y_1)), ..., (z_{i-1}, (x_{i-1}, y_{i-1}))\}$, and*
- *no argument in E defeats$_E$ z_1.*

With this at hand, the semantics are defined in the usual manner.

Definition 2.47. Let $EF = (A, R, D)$ be an EAF and $E \subseteq A$ a conflict–free extension of EF. E is:

- an **admissible** extension of EF iff every argument in E is acceptable w.r.t. E in EF.
- a **preferred** extension of EF iff it is a maximal w.r.t. $\subseteq$ admissible extension of EF.
- a **complete** extension of EF iff it is admissible in EF and every argument acceptable w.r.t. E in EF is in E.
- a **stable** extension of EF iff for every argument $b \notin E$, $\exists a \in E$ s.t. a defeats$_E$ b in EF.

Although the aforementioned semantics are defined quite similarly as in the Dung's framework, the fact that acceptability is understood differently has a strong effect on the definition of the grounded semantics and its relation to e.g. complete extensions. We can define the characteristic operator in a manner similar to AFs, however, in the case of EAFs it is not necessarily monotonic. For these reasons, we will operate only on conflict–free sets and follow the iterative definition of the grounded extension rather than the least–fixed point one:

Definition 2.48. Let $EF = (A, R, D)$ be an EAF, $E \subseteq A$ a set of arguments and let 2^{CF} denote the set of all conflict–free subsets of A in EF. The **characteristic function** $\mathcal{F}_{EF} : 2^{CF} \to 2^A$ of EF is defined as $\mathcal{F}_{EF}(E) = \{a \mid a \text{ is acceptable w.r.t. } E \text{ in } EF\}$.

Definition 2.49. Let $EF = (A, R, D)$ be an EAF. EF is **finitary** iff for every argument $a \in A$, the set $\{b \mid (b, a) \in R\}$ is finite and for every $(a, b) \in R$, the set $\{c \mid (c, (a, b)) \in D\}$ is finite.

Definition 2.50. Let $EF = (A, R, D)$ be a **finitary** EAF. For EF we define a sequence of subsets of A s.t. $\mathcal{F}^0_{EF} = \emptyset$ and $\mathcal{F}^{i+1}_{EF} = \mathcal{F}_{EF}(\mathcal{F}^i_{EF})$. The **grounded** extension of EF is $\bigcup_{i=0}^{\infty}(\mathcal{F}^i_{EF})$.

In other words [64], the grounded extension can be obtained by starting with the empty set and iteratively applying the operator – in this special case, we do in fact obtain a monotonically increasing sequence of extensions.

Fortunately, the usual relation between the admissible and complete extensions and the operator still holds.

Theorem 2.51. *Let $EF = (A, R, D)$ be an EAF and $\mathcal{F}_{EF}$ its characteristic operator. A conflict–free set E is admissible in EF iff $E \subseteq \mathcal{F}_{EF}(E)$ and complete in EF iff $E = \mathcal{F}_{EF}(E)$.*

Just like in the previous frameworks, we introduce the notion of the discarded set to EAFs. It consists of those arguments that are defeated with reinstatement. We can easily use it to redefine acceptability. However, we can also apply it in the case of stable semantics. If a given attack becomes a defeat, then a defense attacker for it has to be outside the set of accepted arguments. Consequently, it will be defeated itself due to the requirements of the stable semantics. Therefore, we can easily show that the collection of all defeats carried out by our set is in fact a reinstatement set for any of them. Although this redefinition might appear a bit of an overkill in this case, some of the proofs will depend on this property.

Definition 2.52. Let $EF = (A, R, D)$ be an EAF and $E \subseteq A$ a set of arguments. The **discarded set** of E in EF is defined as $E^+ = \{a \mid a \in A, \exists b \in E$ s.t. b defeats$_E$ a and there is a reinstatement set on E for this defeat$\}$.

Lemma 2.53. *Let $EF = (A, R, D)$ be an EAF. The set $E \subseteq A$ is a **stable** extension of EF iff E is conflict–free and $E^+ = A \setminus E$.*

Although the usual relation between stable–preferred and complete–grounded extensions does not hold in EAFs (we will discuss this issue further in Section 2.1.4.2), there is still some connection between them:

Theorem 2.54. *Let $EF = (A, R, D)$ be a finitary EAF. The following holds:*

- *every preferred extension is complete, but not vice versa.*

- *every stable extension is complete, but not vice versa.*
- *the grounded extension is a minimal complete extension, but not necessarily the least one.*

Example 6. Let us show the extensions of the presented semantics on an example. Assume an EAF $(\{a, b, c, d, e, f, g\},\ \{(a, b), (b, c), (c, b), (c, d), (d, c), (d, e), (f, g), (g, f)\},\ \{(f, (c, d)),\ (g, (d, c))\})$ depicted in Figure 5. It has in total twelve admissible extensions: $\emptyset$, $\{a\}$, $\{f\}$, $\{g\}$, $\{a, f\}$, $\{a, g\}$, $\{c, g\}$, $\{d, f\}$, $\{a, d, f\}$, $\{a, c, g\}$, $\{c, e, g\}$ and $\{a, c, e, g\}$. Aside from that, we can observe that even though the set $\{a, c\}$ defeats$_{\{a,c\}}$ d, $\{a, c, f\}$ does not – this is one of the reasons the characteristic operator is not monotone. We have only three complete extensions – $\{a\}$, $\{a, d, f\}$ and $\{a, c, e, g\}$. The latter two are also our preferred and stable extensions, while the first one is grounded.

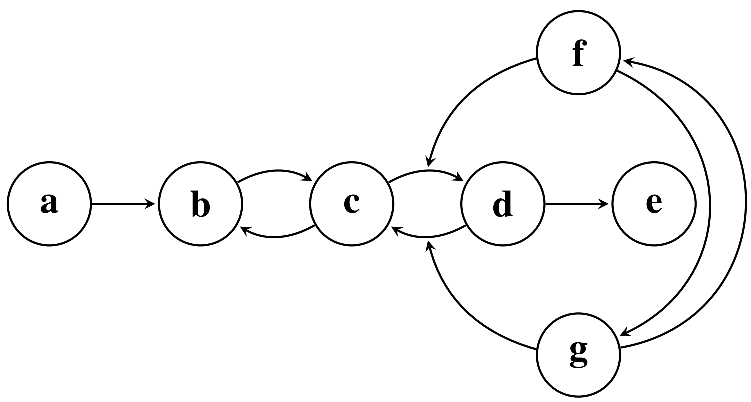


Figure 5: Sample EAF

2.1.4.1 Hierarchical EAFs

Apart from EAFs, the study in [62] also analyzes their particular subclass, referred to as hierarchical. Hierarchy means that the framework is separated into levels w.r.t. the standard attack relation. By this we understand that the attacks in a given group do not affect the arguments outside it. They can only be connected by defense attacks in a way that a the defense attacking argument has to be on a higher level than the arguments participating in the direct conflict.

Definition 2.55. An EAF $HF = (A, R, D)$ is a **hierarchical** EAF (HEAF for short) iff there exists a partition $HF_H = (((A_1, R_1), D_1), ..., ((A_j, R_j), D_j), ...)$ s.t. $A = \bigcup_{i=1}^{\infty} A_i$, $R = \bigcup_{i=1}^{\infty} R_i$, $D = \bigcup_{i=1}^{\infty} D_i$, for every $i = 1...\infty$ (A_i, R_i) is a Dung's framework, and $(c, (a, b)) \in D_i$ implies $(a, b) \in R_i$, $c \in A_{i+1}$. [7]

HF is a **bounded hierarchical** EAF iff its partition HF_H is of the form $(((A_1, R_1), D_1), \ldots, ((A_n, R_n), D_n)$, where $D_n = \emptyset$.

[7] Please recall that partitioning means that all A_i, respectively R_i, D_i sets, are disjoint.

It is worth mentioning that on this class of EAFs, the characteristic operator becomes monotonic and thus the grounded extension can be defined as the usual least fixed point. Moreover, we also retrieve the relation between the stable–preferred semantics known from the Dung's setting:

Proposition 2.56. *Let $HF = (A, R, D)$ be a HEAF, $\mathcal{F}_{HF}$ its characteristic operator and $E, E' \subseteq A$ two conflict–free sets of HF s.t. $E \subseteq E'$. Then $\mathcal{F}_{HF}(E) \subseteq \mathcal{F}_{HF}(E')$.*

Definition 2.57. Let $HF^b = (A, R, D)$ be a bounded hierarchical EAF and $\mathcal{F}_{HF}$ its characteristic operator. The **grounded extension** of HF^b is the least fixed point of $\mathcal{F}_{HF}$.

Along with Theorem 2.51, this gives us the typical relation between the grounded and complete extensions:

Theorem 2.58. *Let $HF^b = (A, R, D)$ be a bounded hierarchical EAF. The grounded extension is the least complete extension of HF^b.*

Theorem 2.59. *Let $bh - EF = (A, R, D)$ be bounded hierarchical EAF. Every stable extension of $bh - EF$ of is preferred, but not vice versa.*

Finally, we can identify classes of EAFs where the definition of conflict–freeness can be replaced with a defeat–based one:

Lemma 2.60. *Let $EF = (A, R, D)$ be a bounded hierarchical EAF or an EAF s.t. there are no arguments $a, b \in A$ for which $(a, b), (b, a) \in R$. A set $E \subseteq A$ is a conflict–free extension of EF iff there are no defeats$_E$ in E.*

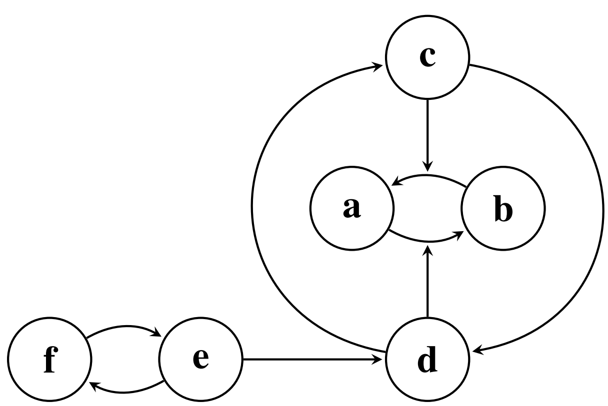


Figure 6: Sample EAF

Example 7. Let $(\{a, b, c, d, e, f\}, \{(a, b), (b, a), (c, d), (d, c), (e, d), (e, f), (f, e)\}, \{(c, (b, a)), (d, (a, b))\})$ be the EAF depicted in Figure 6. We can observe that the framework can be partitioned in several ways; the one with the smallest amount of subframeworks separates a and b from other arguments. We thus obtain a partition (P_1, P_2), where

$P_1 = ((\{a, b\}, \{(a, b), (b, a)\}), \emptyset)$ and $P_2 = ((\{c, d, e, f\}, \{(d, c), (c, d), (e, f), (f, e)\}), \{(d, (a, b)), (c, (b, a))\})$. The admissible extensions of this framework are $\emptyset$, $\{c\}$, $\{e\}$, $\{f\}$, $\{a, c\}$, $\{a, e\}$, $\{c, e\}$, $\{c, f\}$, $\{d, f\}$, $\{b, d, f\}$, $\{a, c, f\}$ and $\{a, c, e\}$. The complete ones are $\emptyset$, $\{f\}$, $\{a, c\}$, $\{a, c, e\}$, $\{a, c, f\}$ and $\{b, d, f\}$. The first is grounded and the last three are both preferred and stable. We can observe that the stable extensions are now preferred and the grounded extension is the least complete one. We thus retrieve the traditional relations between the semantics.

2.1.4.2 EAF Controversies

EAFs are interesting tools, however, there are also certain controversies concerning their design, which make full translations from EAFs to other frameworks problematic. They are also the reason why the transformations have only been done for hierarchical EAFs so far [64]. First of all, we will show on an example that the usual relation between the complete and grounded extensions does not hold, as already mentioned in the previous section. Second of all, we perform a similar analysis for the stable and preferred semantics. Please note that we treat these problems simply as different design intuitions – after all, certain families of ADF semantics behave in a very similar manner. Unfortunately, the issues arising concerning the conflict–free semantics of EAFs and inducing the attacks between defense attacking arguments are a bit more than that. We will describe these problems mostly from the structural perspective. However, we will also show that certain notions become somewhat counterintuitive when we consider EAF instantiations. Please note that these problems were already observed by the authors in [66] and led to the proposal of collective EAFs with modified definitions, which will be introduced in the next section. Discussing them here is meant to serve as an explanation for some of the choices we will have to make when translating between EAFs and other argumentation frameworks.

Let us start with the complete and grounded semantics. Unlike in AFs and various other frameworks, the EAF complete extensions do not form a complete semilattice. As a result, we no longer have the least complete extension, only a number of minimal ones. Thus, although the grounded extension is still a complete set, it is not necessarily the least complete one.

Example 8 ([9])**.** Let us consider the EAF $(\{a, b, c\}, \{(b, a), (c, b)\}, \{(b, (c, b))\})$ depicted in Figure 7. The sets $\{a, c\}$ and $\{b, c\}$ are its complete extensions, with the first one also being grounded.

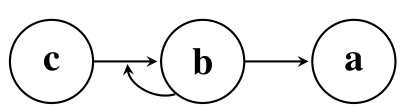


Figure 7: Sample EAF

Let us now consider the stable semantics. Unlike in any other framework, the stable extensions might not be preferred extensions. Although it was claimed otherwise at first [62], this issue was already noted in [39]. We will explain it on an example:

Example 9. Let $EF = (\{a, b, c, d\}, \{(a, b), (d, c)\}, \{(b, (d, c)), (c, (a, b))\})$ be the EAF depicted in Figure 8. The set $E_1 = \{a, d\}$ is conflict–free; there are no attacks between a and d to start with. Moreover, a defeats$_{E_1}$ b and d defeats$_{E_1}$ c. Thus, E_1 is also a stable extension. The set $E_2 = \{a, b, c, d\}$ is also conflict–free; for every attack there is a defense attack and there are no symmetric conflicts in the framework. Since there are no arguments not covered by the set, it is also trivially stable. Consequently, we have a stable extension E_1 which is not preferred.

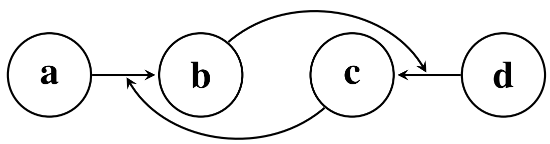


Figure 8: Sample EAF

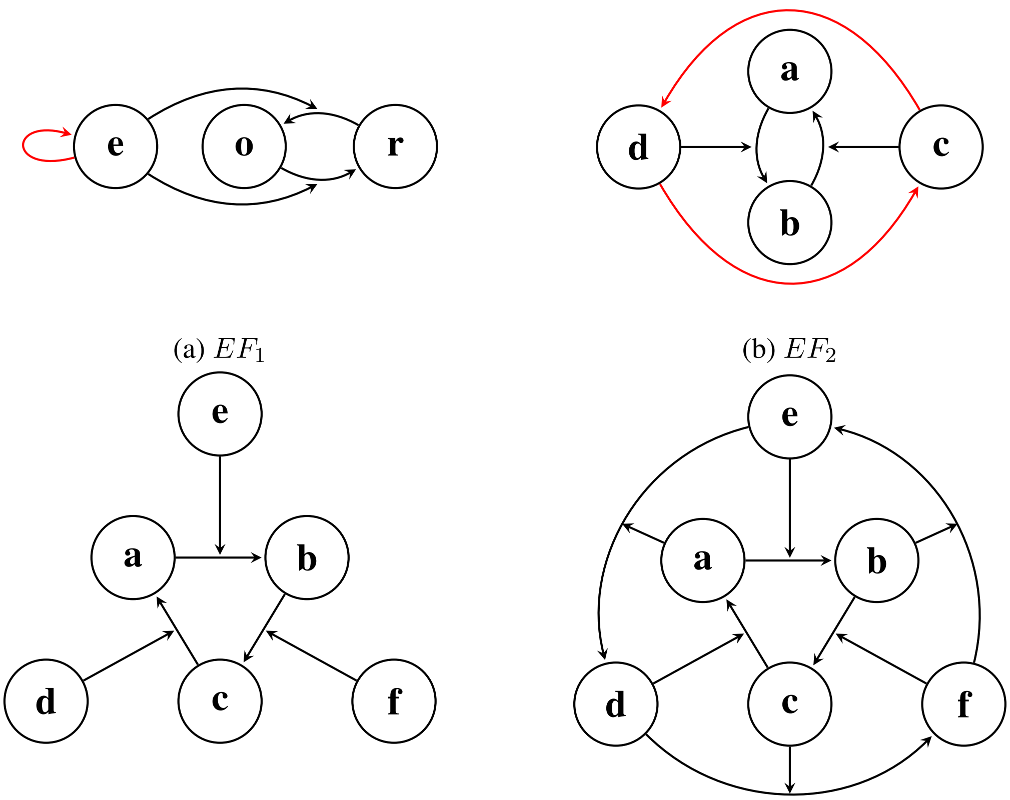
In the definitions of EAFs we can observe that symmetric conflicts are induced between arguments that defense attack symmetric conflicts. While the "is more preferred" reading of the defense attack, presented at the beginning of this section, motivates this restriction, defense attack can also possess different interpretations. To start with, preferences can be used to handle knowledge we are not certain of, and thus to some extent can express probabilities of arguments or relations [3, 4]. However, as seen in the following example, such usage can create unexpected behaviors in the framework.

Example 10. Let us modify an example from [53]; during a robbery, the car of the guilty part has been spotted. However, some witnesses say it was orange, while other claim it was red. Due to human perception being imperfect and the two colors being somewhat similar, it is possible that those two claims are not in actual conflict. Let us thus assume such an error is very likely and thus would like to "override" the attacks between the testimonies.

We introduce the following arguments: o for the claim that the getaway car is orange, r that it is red, and es for the perception error and color similarity. The intuitive representation of this situation would be $(\{o, r, e\}, \{(o, r), (r, o)\}, \{(e, (o, r)), (e, (r, o))\})$, as depicted in Figure 9. However, this framework is not a proper EAF. According to the attack restriction, e should become a self–attacker, even though we can observe that such modeling is counterintuitive and there is no reason in the meaning e that would make it self–conflicting.

We can create more examples in the similar spirit with different defense attackers as well, not just one. This issue has also been noticed in [66]. The authors look at a subclass of EAFs from the structured argumentation point of view and show that not for all instantiations the induced attacks are constructed.

A different – structural – issue of the induced attacks concerns the fact that they are only generated for cycles of length two. Let us consider two frameworks $EF_2 = (\{a, b, c, d\}, \{(a, b), (b, a), (c, d), (d, c)\}, \{(c, (a, b)), (d, (b, a))\})$ and $EF_3 = (\{a, b, c, d, e\}, \{(a, b), (b, c), (c, a)\}, \{(d, (c, a)), (e, (a, b)), (f, (b, c))\})$, depicted in Figure 9. In the first case, an attack cycle between c and d is induced from their respective defense attacks. We can observe that the framework EF_2 resembles EF_1 from construction; we have an attack cycle and each attack has a corresponding defense attack. The only difference is that the EF_1 cycle is of length two, and the EF_2 one of three. Nevertheless, in the latter case no attacks between d, e and f are enforced, which appears to be a certain inconsistency in the design.



(a) EF_1 (b) EF_2

(c) EF_3 (d) EF_4

Figure 9: Sample EAFs

The next problem lies in conflict–freeness. Recall the framework $(\{a, b, c, d\}, \{(a, b), (b, a), (c, d), (d, c)\}, \{(a, (d, c)), (b, (c, d)), (c, (b, a)), (d, (a, b))\})$ from Figure 4 and Example 5, in which we discussed why conflict–freeness cannot be defined with defeats only. Let us analyze the (a, b) attack; it is overridden by d, attack on which by c is in turn overridden by b. We thus have a case in which b, even though somewhat indirectly, helps to “cancel” an attack that is carried out against it. This is a cyclic behavior, perhaps undesirable, but in certain ways not much different from the self–defense that is permitted in most of the semantics apart from grounded.

Let us now look at the framework $EF_4 = (\{a, b, c, d, e\}, \{(a, b), (b, c), (c, a)\}, \{(d, (c, a)), (e, (a, b)), (f, (b, c)), (a, (e, d)), (b, (f, e)), (c, (d, f))\})$ depicted in Figure 9. It is, in principle, a construction similar to the previously described structure. c is attacked by b, but the attack can be overridden by f, the defense attack on which is overridden by c in turn. Thus, c again promotes its own acceptance, and unlike in the first framework, this situation is permitted. Similarly, in a very straightforward framework $(\{a, b\}, \{(a, b)\}, \{(b, (a, b))\})$, where b directly defense attacks the attack on it, $\{a, b\}$ is still a conflict–free extension. This self–reinstatement is permitted in EAF semantics, with the exception of the grounded case. Therefore, again we are faced with a situation in which only cycles of length two are treated uniquely by the definition, and there is an inconsistency within the design. Please note that in [66], the authors created a modification of bounded hierarchical EAFs, which included collective defense attacks and where the current notion of conflict–freeness is replaced by requiring that no defeats are present in a given extension. Their choice was to let instantiation create appropriate frameworks, rather than make such restrictions on an abstract level. They also show that despite the change, their construction still satisfied the rationality postulates [24]. Therefore, we now move on to describing the new framework, and end this section.

2.1.4.3 EAFs with Collective Defense Attacks

In the previous section we have already noted that in order to address certain issues of EAFs, the authors in [66] introduce a modification of the framework changing the current notion of conflict–freeness and allowing defense attacks to be collective. Thus, in the new approach, no additional attacks between arguments are induced:

Definition 2.61. An **extended argumentation framework with collective defense attacks** (EAFC for short) is a tuple $EFC = (A, R, D)$, where A is a set of **arguments**, $R \subseteq A \times A$ is a set of **attacks** and $D \subseteq (2^A \setminus \emptyset) \times R)$ is the set of **collective defense attacks**. EFC is **bounded hierarchical** iff there exists a partition $\delta_H = (((A_1, R_1), D_1), ..., ((A_n, R_n), D_n))$ s.t. $D_n = \emptyset$, $A = \bigcup_{i=1}^n A_i$, $R = \bigcup_{i=1}^n R_i$, $D = \bigcup_{i=1}^n D_i$, for every $i = 1...n$ (A_i, R_i) is a Dung's framework, and $(c, (a, b)) \in D_i$ implies $(a, b) \in R_i$, $c \subseteq A_{i+1}$.

Although the study was limited to the bounded hierarchical subclass, we will define the semantics for the general case. While the definition of conflict–freeness is changed, the other notions are just adapted to collective defense attacks:

Definition 2.62. Let $EFC = (A, R, D)$ be an extended argumentation framework with collective defense attacks.

An argument a **defeats**$_E$ an argument $b \in A$ in EFC w.r.t. set $E \subseteq A$ iff $(a, b) \in R$ and there is no $c \subseteq A$ s.t. $(c, (a, b)) \in D$.

A set of pairs $R_E = \{(x_1, y_1), ..., (x_n, y_n)\}$ s.t. x_i defeats$_E$ y_i in EFC and for $i = 1...n$, $x_i \in E$, is a **reinstatement set** on E for a defeat$_E$ by argument a on argument b iff $(a, b) \in R_E$ and for every pair $(x, y) \in R_E$ and set of arguments $c \subseteq A$ s.t. $(c, (x, y)) \in D$, there is a pair $(x', y') \in R_E$ for some $y' \in c$.

A set of arguments $E \subseteq A$ is **conflict–free** in EFC iff for every $a, b \in E$, if $(a, b) \in R$, then $\exists c \subseteq A$ s.t. $(c, (a, b)) \in D$ (i.e. there are no $a, b \in E$ s.t. a defeats$_E$ b in EFC).

The remaining semantics are defined in the same way as for standard EAFs, though the fundamental lemma and monotonicity of the characteristic function hold only in the bounded hierarchical case.

Proposition 2.63. *Let $EFC = (A, R, D)$ be a bounded hierarchical EAFC. If $E \subseteq A$ is an admissible extension of EFC and $a, b \in A$ are acceptable w.r.t. E in EFC, then $E' = E \cup \{a\}$ is admissible in EFC and b is acceptable w.r.t. E' in EFC. For two conflict–free extensions $E, E' \subseteq A$ of EFC s.t. $E \subseteq E'$, it holds that $\mathcal{F}_{EF}(E) \subseteq \mathcal{F}_{EF}(E')$.*

Finally, please note that just like we could verify reinstatement by defense attack analysis in EAFs, we can do it in EAFCs. The following theorem can be proved in a fashion similar to Theorem 2.46:

Theorem 2.64. *Let $EFC = (A, R, D)$ be a finite EAFC and $E \subseteq A$ be a conflict–free extension of EFC. If an argument $a \in E$ defeats$_E$ an argument $b \in A$, then there is no reinstatement set for this defeat$_E$ on E, iff there exists a sequence $((z_1, (x_1, y_1)), ..., (z_n, (x_n, y_n)))$ of distinct defense attacks from D s.t.*

- *there is a set of arguments argument $g \subseteq A$ s.t. $x_n = a$, $y_n = b$ and $z_n = g$,*
- *no two pairs (x_i, y_i) and (x_j, y_j) are the same for $i \neq j$,*
- *for every $(z_i, (x_i, y_i))$ where $1 < i \leq n$, either no argument h in E defeats$_E$ any argument $z \in z_i$ or for every such defeat, there exists a set of arguments $l \subseteq A$ s.t. $(l, (h, z)) \in \{(z_1, (x_1, y_1)), ..., (z_{i-1}, (x_{i-1}, y_{i-1}))\}$, and*
- *no argument in E defeats$_E$ any argument in z_1.*

Example 11. Let us consider the EAFC $EFC = (\{a, b, c, d, e, f\}, \{(c, f), (d, e), (f, a)\}, \{(\{f\}, (d, e)), (\{a, b, c\}, (d, e)), (\{e\}, (c, f))\})$ depicted in Figure 10. First of all, $\emptyset$ is a trivial admissible extension. We can also observe that none of the arguments b, c and d are attacked; therefore, any combination of them will form an admissible extension as well, and this gives us seven more sets. Next, we have the $\{e, f\}$ extension; w.r.t. this set, neither c defeats f nor d defeats e, and thus there is nothing to defend from. Adding any of the $\{b, c, d\}$ arguments does not change that, and we obtain the final seven extensions – $\{b, e, f\}$, $\{c, e, f\}$, $\{d, e, f\}$, $\{b, c, e, f\}$, $\{b, d, e, f\}$, $\{c, d, e, f\}$ and $\{b, c, d, e, f\}$. Out of all the admissible sets, only $\{b, c, d\}$ and $\{b, c, d, e, f\}$ are complete, with the first being the grounded extension and the latter the single preferred one. Furthermore, $\{b, c, d, e, f\}$ is our only stable set, as $\{b, c, d\}$ does not defeat a.

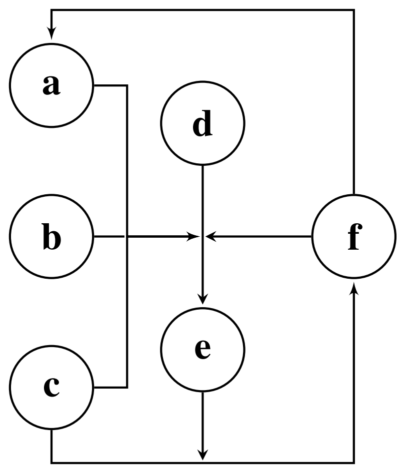


Figure 10: Sample EAFC

2.2 Support–Based Argumentation Frameworks

In the Dung's framework, based on the existing conflicts we can derive defense, which can be seen as a type of a positive indirect relation between arguments. However, with time it was acknowledged that a structure going beyond that is required and that the defense cannot account for all interactions between arguments that are not negative. Consequently, the notion of support was introduced. There was an initial hope that just like in the case of attack, we will be able to preserve a certain level of abstraction [28], and thus abstract support and the bipolar argumentation framework [28–30] were introduced. However, various arguments against this claim have been found, and more specialized forms of support have been researched. This includes deductive [19], necessary [69, 70] and evidential supports [72, 73, 78], with the latter two developed in their own dedicated frameworks. In this section, we will go through them one by one.

2.2.1 Abstract and Deductive Supports: Bipolar Argumentation Framework

We will start by introducing the abstract and deductive approaches. Let us first recall the definition of a bipolar argumentation framework [30], which extends the Dung's framework by adding a new binary relation that is meant to account for support:

Definition 2.65. The **bipolar argumentation framework** (BAF for short) is a tuple (A, R, S), where A is a set of **arguments**, $R \subseteq A \times A$ represents the **attack** relation and $S \subseteq A \times A$ the **support** [8].

The first type of support studied in this setting is the abstract support [28]. The most significant difference between this one and any other interpretation of support, or even

[8]Please note that in the original version from [28], it was also assumed that $R \cap S = \emptyset$. However, this restriction is dropped in later works.

conflict, is the fact that it does not affect the acceptability of an argument. By this we understand that an argument does not require its supporters or the arguments it supports to be presented in an extension. Abstract support merely encompasses a certain, undefined positive link between the two arguments and the presented research was focused on studying the consequences of using it in a framework. This has lead to the development of additional, indirect forms of conflict, which were later used to enhance the semantics known from the Dung's setting. The first developed type was the *supported attack*. Later, in [29] the *secondary attack* was also introduced (first referred to as diverted).

Another type of a positive relation in BAFs, the *deductive support* [19] was introduced in order to address certain issues with the coalition translation from BAFs to AFs [29]. We say that a **deductively supports** b if acceptance of a implies the acceptance of b and not acceptance of b implies non acceptance of a. Although originally used rather for coalitions and meta–argumentation purposes, it is also studied in a standard setting in [30]. The deductive behavior of support in BAFs is achieved by introducing another type of indirect conflict, namely the *mediated attack*. Further study [30] also included the *extended attack* in order to accommodate the handling of necessary support (we will discuss this form of support in Section 2.2.2). Please note we will recall only one form of the extended attack, as every other one is already subsumed by the existing notions (see Definition 2.82).

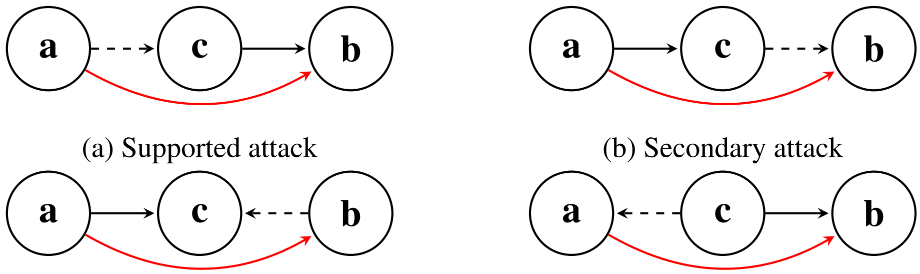
Definition 2.66. Let $BF = (A, R, S)$ be a BAF and $a, b \in A$ two arguments. There is:

- a **supported attack** from a to b in BF iff there exists an argument c s.t. there is a sequence of supports from a to c (i.e. $aS...Sc$) and cRb.
- a **secondary attack** from a to b in BF iff there is an argument c s.t. $cS...Sb$ and aRc.
- a **mediated attack** from a to b in BF iff there is an argument c s.t. there is a sequence of supports from b to c and aRc.
- an **extended attack** from a to b in BF iff there is an argument c s.t. there is a sequence of supports form c to a (i.e. $cS...Sa$) and cRb.

The collections of the respective indirect attacks will be abbreviated to R^{sup}, R^{sec}, R^{med} and R^{ext}. Please note that even though there are many types of conflicts available, it does not mean that all of them need to be used – the choice depends on what we intend to use a given BAF for. We can see the examples of all of the listed attacks in Figure 11 – the indirect conflicts are marked in red.

What was noticed in [30] is that the combinations of different types of conflicts in the framework would give us a basis to derive even more indirect attacks. This led e.g. to the introduction of the *super–mediated attack*, which takes into account both direct and supported attacks:

Definition 2.67. Let $BF = (A, R, S)$ be a BAF and $a, b \in A$ two arguments. There is a **super–mediated attack** from a to b in BF iff there is an argument c s.t. there is a sequence of supports from b to c and a direct or supported attacks c.



(a) Supported attack

(b) Secondary attack

(c) Mediated attack

(d) Extended attack

Figure 11: Indirect attacks in BAFs

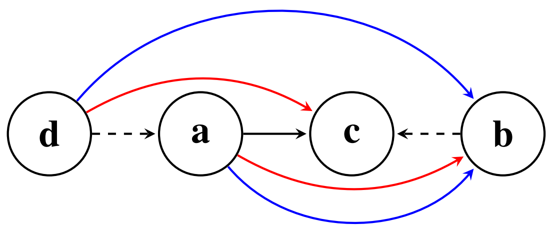


Figure 12: Super–mediated attacks

An example of the super–mediated attack can be observed in Figure 12; in red we have marked the existing supported and mediated conflicts, while the super ones are depicted in blue.

In summary, we can observe that we can have indirect attacks built from direct ones and the existing support, then additional indirect ones built on previous indirect ones and support and so on. We will thus speak about first tier attacks, second tier attacks etc., and leave it to the reader to choose which tiers of which conflicts are to be used. We therefore propose a new definition of indirect attacks, though please note that it is not our intent to replace the existing ones. We only want to grasp certain patterns that appear in BAFs and be able to show whether some general properties hold, without being forced to assume a fixed set of indirect attacks.

Definition 2.68. Let $BF = (A, R, S)$ be a BAF. The **tiered indirect attacks** of BF are as follows:

- $R_0^{ind} = \emptyset$
- $R_1^{ind} = \{R_\emptyset^{sup}, R_\emptyset^{sec}, R_\emptyset^{med}, R_\emptyset^{ext}\}$
- $R_i^{ind} = \{R_E^{sup}, R_E^{sec}, R_E^{med}, R_E^{ext} \mid E \subseteq R_{i-1}^{ind}\}$ for $i > 1$, where:
  - $R_E^{sup} = \{(a, b) \mid$ there exists an argument c s.t. there is a sequence of supports from a to c and $(c, b) \in R \cup \bigcup E\}$.

- $R^{sec}_E = \{(a,b) \mid$ there exists an argument c s.t. there is a sequence of supports from c to b and $(a,c) \in R \cup \bigcup E\}$.
- $R^{med}_E = \{(a,b) \mid$ there exists an argument c s.t. there is a sequence of supports from b to c and $(a,c) \in R \cup \bigcup E\}$.
- $R^{ext}_E = \{(a,b) \mid$ there exists an argument c s.t. there is a sequence of supports from c to a and $(c,b) \in R \cup \bigcup E\}$.

With R^{ind} we will denote the collection of all sets of indirect attacks $\bigcup_{i=0}^{\infty} R^{ind}_i$.

In this notation, the set of super–mediated attacks is $R^{med}_{R^{sup}}$. It is worth mentioning that not all conflicts are created in a unique manner and especially the higher tier attacks sets might not be disjoint. Furthermore, not every tier has to bring something new; for example, $R^{sup} = R^{sup}_{R^{sup}}$.

Please note that although the presented definition might look intimidating at first, usually only some of the conflicts are studied at a time. What needs to be stated explicitly is that BAFs were meant as a research framework for analyzing different types of supports and their consequences. Therefore, there is no "absolute" way to choose what sort of indirect attacks need to be taken into account and different interpretations of support might call for different attacks. Our intent is merely to gather and organize the available approaches. Moreover, even in the case of a set meaning of support, various modeling approaches can be found (see Section 2.2.2). As a result, defining the semantics of BAFs is quite difficult. We will now recall the two main styles.

The original approach [28] for handling abstract support used indirect attacks to strengthen the notion of conflict–freeness and to introduce coherence restrictions, such as not attacking and supporting the same argument. The definition of defense was left the same as in the Dung's setting (i.e. required direct attack against direct attack). One of the reasons to motivate such choice was the fact that in BAFs, support is not seen as having the same strength as attack. Moreover, since the argument did not require its (abstract) supporters to be present in order to be accepted, there is no reason why having its supporters attacked should render the argument unacceptable.

The new approach [30] defined the semantics of BAFs through the translations to AFs and was meant to handle more than just abstract support. The argumentation graph was extended with the desired indirect attacks and the support relation removed. The remaining structure was then basically a Dung's framework and its extensions gave us the results of the original BAF under the desired semantics. To this approach we will later refer to as attack propagation translation (see Sections 3.3 and 9.1). What can also be observed is that by elevating indirect attacks to the level of direct ones, we now can defend from and with them. This approach is adequate for stronger forms of support, such as deductive, where the presence of supporters does affect the status of an argument.

In what follows we will extend the semantics of [28] so that they can be parametrized with the desired indirect attacks. We will also parametrize defense, so that one can choose which types of conflicts should be taken into account.

Definition 2.69. Let $BF = (A, R, S)$ be a BAF, $E \subseteq A$ and $R' \subseteq R^{ind}$ the collections of indirect attacks in BF that we want to consider. The set E **defends** a in BF w.r.t. R' if for every $b \in A$ s.t. $(b, a) \in (R \cup \bigcup R')$, there exists $c \in E$ s.t. $(c, b) \in (R \cup \bigcup R')$.

We can observe that if we assume $R' = \emptyset$, we are brought back to the Dung's definition of defense which is used in abstract support. We will now introduce two strengthened forms of the AF conflict–freeness and the support closure. Please note that in addition to the original definition, we will also include the inverse closure, present e.g. in the binary version of AFNs (see Section 2.2.2.1).

Definition 2.70. Let $BF = (A, R, S)$ be a BAF and $R' \subseteq R^{ind}$ the collections of indirect attacks in BF. A set $E \subseteq A$ is **+conflict–free** in BF w.r.t. R' iff $\nexists a, b \in E$ s.t. $(a, b) \in (R \cup \bigcup R')$. E is **safe** in BF w.r.t. R' iff $\nexists b \in A$ s.t. b is at the same time attacked by a member of E in $R \cup \bigcup R'$ and either there is a sequence of supports from an element of E to b, or $b \in E$. E is **closed** under S in BF iff $\forall b \in E, a \in A$, if bSa then $a \in E$. E is **inverse closed** under S in BF iff $\forall b \in E, a \in A$, if aSb then $a \in E$.

We can now proceed with further semantics. Please note we will not assume that conflict–freeness and defense are parametrized with the same types of indirect attacks. This approach could be useful when we mix different types of support within the framework. Moreover, strengthening conflict–freeness does not necessarily mean we want to broaden the notion of defense, which was the case in abstract support. However, we will keep the definition of stability almost the same as the original one [28], i.e. the same types of attacks will be used for arguments inside and outside the extension.

Definition 2.71. Let $BF = (A, R, S)$ be a BAF and $R' \subseteq R^{ind}$, $R'' \subseteq R^{ind}$ two collections of indirect attacks in BF. The set $E \subseteq A$ is:

- **d–admissible** w.r.t. (R', R'') in BF iff it is +conflict–free w.r.t. R' in BF and defends all its elements w.r.t. R'' in BF.
- **s–admissible** w.r.t. (R', R'') in BF iff it is safe w.r.t. R' and defends all its elements w.r.t. R'' in BF.
- **c–admissible** w.r.t. (R', R'') in BF iff it is +conflict–free w.r.t. R', closed for S and defends all its elements w.r.t. R'' in BF.
- **i–admissible** w.r.t. (R', R'') in BF iff it is +conflict–free w.r.t. R', inverse closed for S and defends all its elements w.r.t. R'' in BF.
- **d–/s–/c–/i–preferred** w.r.t. (R', R'') in BF iff it is maximal w.r.t. set inclusion d–/s–/c–/i–admissible w.r.t. (R', R'') in BF.
- **stable** w.r.t. R' in BF iff it is +conflict–free w.r.t. R' in BF and $\forall b \notin E$, there is a member of E attacking b in $R \cup \bigcup R'$.

We can observe that the provided definition did not include the grounded and complete semantics. We will come back to this problem in a moment and would like to discuss certain general properties first:

Lemma 2.72. ***BAF Fundamental Lemma*** *Let $BF = (A, R, S)$ be a BAF, $R' \subseteq R^{ind}, R'' \subseteq R^{ind}$ two collections of indirect attacks in BF, $E \subseteq A$ a d–admissible extension w.r.t. (R', R'') and $a, b \in A$ arguments defended by E w.r.t. R'' in BF. If $R' = R''$, then $E' = E \cup \{a\}$ is d–admissible w.r.t. (R', R'') and b is defended by E w.r.t. R'' in BF.*

Unfortunately, this is the only general positive result we have. If we assume that $R' \subset R''$, $R'' \subset R'$, or look at s–/c–/i–admissibility, we can find examples in which proceeding with adding a defended argument can break +conflict–freeness or given us admissible extensions of a different type:

Example 12. Let us consider the framework $BF_1 = (\{a, b, c\}, \{(b, c)\}, \{(a, b)\})$ and parametrize +conflict–freeness with $\{R^{med}\}$ and defense with $\emptyset$. We can observe that b mediate attacks a. However, $\{a\}$ is a d–admissible extension with our parametrization; it is both +conflict–free and is not directly attacked by any other argument in the set. Similarly, there are no arguments directly attacking b, and thus the argument is defended by $\{a\}$. Unfortunately, the set $\{a, b\}$ is not even +conflict–free, let alone d–admissible, due to the indirect attack.

We can now analyze the framework $BF_2 = (\{a, b\}, \{(a, b)\}, \{(a, b)\})$ in which a both supports and attacks b. Let us parametrize +conflict–freeness with $\emptyset$ and defense with $\{R^{med}\}$. We can observe that a mediate attacks itself. However, as this is not a direct attack $\{a\}$ is +conflict–free. Furthermore, a attacks its own attacker – namely, itself – and thus is defended by $\{a\}$ w.r.t. $\{R^{med}\}$. Thus, technically speaking, the set is d–admissible. Moreover, for similar reasons, it also defends b, even though $\{a, b\}$ is not +conflict–free to start with.

Let us look at the same framework but from the point of safety. No supported or secondary attacks are introduced in BF_2, while a (super) mediate attacks itself and b extended attacks itself. We can observe that $\emptyset$ is easily s–admissible. Moreover, as long as we do not parametrize defense with a collection of attacks that includes (super) mediated conflicts, a can be seen as an initial argument. Therefore, $\emptyset$ is s–admissible and defends a. Unfortunately, $\{a\}$ is not s–admissible; at the same time b is attacked and supported by the set.

Let us now consider a simplified version of the previous framework with a single supporting edge, i.e. $BF_3 = (\{a, b\}, \emptyset, \{(a, b)\})$. $\emptyset$ is an admissible extension of any type. Furthermore, as there are no attacks in the framework whatsoever, no additional indirect conflicts of any type will be created and both a and b are defended by $\emptyset$ w.r.t. any parametrization. However, we can observe that even though $\emptyset$ is c–admissible and defends a, $\{a\}$ is not c–admissible due to the absence of b. Similarly, $\emptyset$ is i–admissible and defends b, but $\{b\}$ is not i–admissible.

There are two ways the loss of admissibility can be addressed. Either we limit ourselves to a particular collection of indirect conflicts for which certain properties can still hold, or we adapt the definition of defense. Both in AFNs and EAFs, which will be described in the next sections, defense is extended in order to ensure that a given argument is not only protected from attacks, but also sufficiently supported. The fact that the BAF defense does include the support counterpart led to the loss of e.g. c–/i– admissibility in our example. Unfortunately, this might mean that each type of admissible semantics may require its own notion of acceptability. Due to the fact that in the newer works [30, 31] the focus on BAF semantics shifted from the d–/s–/i–/c–families to translation families that will be described in Section 9.1, we leave this issue with defense for future work.

The fact that in principle, the BAF version of the Fundamental Lemma does not hold is the main reason why we will not attempt to recreate most of the complete and grounded semantics. Although the majority of the "ingredients" are already available, following the Dung's style definition would create semantics that are in principle not universally defined. Let us look at the following example:

Example 13. Let $BF_1 = (\{a, b\}, \{(b, b)\}, \{(a, b)\})$ be a BAF. Clearly, $\emptyset$ is +conflict–free, safe, closed under support, and is an admissible extension of any type and with any parametrization. We will now focus on the first tier attacks and assume the same conflicts both in +conflict–freeness and defense. The combination of (a, b) support and (b, b) attack creates a supported attack (a, b) and a mediated attack (b, a). No secondary or extended conflict is produced.

Let us now consider defense being parametrized with either secondary attacks, or extended attacks, or no indirect attacks at all. In this case, a is an unattacked argument and is thus defended by $\emptyset$. Therefore, $\emptyset$ cannot be considered complete if we assume the "defended arguments are included in the extension" approach. Unfortunately, while $\{a\}$ is still +conflict–free and safe, it is no longer closed under support. Moreover, it will never be; b directly attacks itself and cannot appear in a +conflict–free extension with any parametrization and thus cannot be defended neither by $\{a\}$ nor $\emptyset$. This means that no set of arguments in BF_1 qualifies for the c–complete extension.

We can now consider parametrization with the supported attack. Again, a is not attacked by any argument, and thus is defended by $\emptyset$. However, even though $\emptyset$ is safe, $\{a\}$ is not – a at the same times supports and support attacks b. Since neither $\{a, b\}$ nor $\{b\}$ can possibly be admissible in any way, our framework has no s–complete extensions.

Let $BF_2 = (\{a, b\}, \{(a, a)\}, \{(a, b)\})$ be a modification of BF_1; in this case, it is a that is a self–attacker, not b. There are no mediated attacks in this framework. Consequently, b is an unattacked argument w.r.t. direct and mediated conflicts, and as such is defended by the empty set. However, even though $\emptyset$ is i–admissible, $\{b\}$ is not due to the absence of a. Since a is a self–attacker, it will never be a part of any extension. Hence, our framework has no i–complete extension w.r.t. the aforementioned conflicts.

The conclusion thus is that the complete semantics cannot be that easily shifted into the BAF setting and that the general results are not very appealing. We may obtain better results if we limit ourselves to particular types of indirect conflicts. Aside from that, we can only define d–complete and d–grounded extensions if we assume the same parameters both for +conflict–freeness and defense.

Definition 2.73. Let $BF = (A, R, S)$ be a BAF and $R' \subseteq R^{ind}$ be a collections of indirect attacks in BF. A d–admissible extension $E \subseteq A$ w.r.t. (R', R') of BF is **d–complete** w.r.t. (R', R') in BF iff all arguments defended w.r.t. R' by E are in E.

Definition 2.74. Let $BF = (A, R, S)$ be a BAF and $R' \subseteq R^{ind}$ a collections of indirect attacks in BF. The **d–characteristic operator** $d - \mathcal{F}_{BF} : 2^A \rightarrow 2^A$ of BF w.r.t. R' is defined as $d - \mathcal{F}_{BF}(E) = \{a \mid a$ is defended by E w.r.t. R' in $BF\}$.

We can easily observe that our operator is monotonic w.r.t. $\subseteq$. If a given set E contains arguments attacking given elements w.r.t. R', then so does any E' s.t. $E \subseteq E'$. Moreover, we can observe that applying the operator to a +conflict–free set with the same parametrization yields a +conflict–free set.

Lemma 2.75. *Let $BF = (A, R, S)$ be a BAF, $R' \subseteq R^{ind}$ a collections of indirect attacks in BF and $d - \mathcal{F}_{BF}^{R'} : 2^A \rightarrow 2^A$ the d–characteristic operator of BF w.r.t. R'. Given two sets of arguments $E, E' \subseteq A$, if $E \subseteq E'$, then $d - \mathcal{F}_{BF}^{R'}(E) \subseteq d - \mathcal{F}_{BF}^{R'}(E')$.*

Lemma 2.76. *Let $BF = (A, R, S)$ be a BAF and $R' \subseteq R^{ind}$ a collection of indirect attacks in BF. If a set $E \subseteq A$ is +conflict–free w.r.t. R', then so is $d - \mathcal{F}_{BF}^{R'}(E)$.*

From this, the following properties follow:

Lemma 2.77. *Let $BF = (A, R, S)$ be a BAF, $R' \subseteq R^{ind}$ a collection of indirect attacks in BF and $d - \mathcal{F}_{BF}^{R'}$ the d–characteristic operator of BF w.r.t. R'. A set of arguments $E \subseteq A$ that is +conflict–free w.r.t. R', is d–admissible w.r.t. (R', R') iff $E \subseteq d - \mathcal{F}_{BF}^{R'}(E)$. A set of arguments $E \subseteq A$ that is +conflict–free w.r.t. R', is d–complete w.r.t. (R', R') iff $E = d - \mathcal{F}_{BF}^{R'}(E)$.*

We can now define the d–grounded extension as the least fixed point of our operator and observe that if we use the same parametrization, it is +conflict–free as well.

Definition 2.78. Let $BF = (A, R, S)$ be a BAF and $R' \subseteq R^{ind}$ the collections of indirect attacks in BF. The **d–grounded extension** of BF w.r.t. R' is the least fixed point of the d–characteristic operator of BF w.r.t. R'.

Proposition 2.79. *Let $BF = (A, R, S)$ be a BAF, $R' \subseteq R^{ind}$ a collection of indirect attacks in BF and $E \subseteq A$ the d–grounded extension w.r.t. R' of BF. E is +conflict–free w.r.t. R'.*

Theorem 2.80. *Let $BF = (A, R, S)$ be BAF and $R' \subseteq R^{ind}$ the collections of indirect attacks in BF. The following holds:*

- *every d–preferred extension of BF w.r.t. (R', R') is a d–complete extension of BF w.r.t. (R', R'), but not vice versa.*

- *the d–grounded extension of BF w.r.t. R' is the least w.r.t. set inclusion d–complete extension of BF w.r.t. (R', R').*
- *every stable extension of BF w.r.t. R' is a d–preferred extension w.r.t. (R', R'), but not vice versa.*

The translation–based family of BAF semantics will be described in Section 9.1.

2.2.2 Necessary Support: Abstract Frameworks with Necessities

The introduction of abstract support started a new line of research into various types of positive relations. One of them is the *necessary support*, first developed in [70]. We say that an argument a *necessary supports* b if we need to assume a in order to accept b. Consequently, argument's supporters had to be present in an extension. Since cutting off such supporters would now discard an argument, it was natural to expect that this sort of indirect attack should be suitable for defense. Naturally, the abstract support and the BAF semantics did not meet these requirements, and a new type of a framework was created. We will recall its original form from [70], which was heavily inspired by BAFs and had certain design issues. Then we will continue with the current definition [69] that introduces a number of important changes and fixes.

2.2.2.1 Binary AFNs

Let us recall the original formulation of AFNs, presented in [70]:

Definition 2.81. A (binary) **abstract argumentation framework with necessities** (AFN) is a tuple (A, R, N) where A is a set of **arguments**, $R \subseteq A \times A$ represents the **attack** relation and $N \subseteq A \times A$ the **necessity** relation.

Support in AFNs leads to the development of additional notions of attack:

Definition 2.82. Let $FN = (A, R, N)$ be an AFN and $a, b \in A$. There is an **extended necessity** from a to b, denoted as $a\, N^+ b$, iff there is a sequence $a_1\, N \ldots N\, a_n$ $(n \geq 2)$ where $a_1 = a$ and $a_n = b$. There is an **extended attack** of b by a, denoted as $a\, R^+ b$, iff any of the following is the case: *i*) $a\, R\, b$, *ii*) $\exists c \in A$ s.t. $a\, R\, c\, N^+ b$, or *iii*) $\exists c \in A$ s.t. $c\, R\, b$ and $c\, N^+ a$.

The first two cases of extended attack correspond to direct and secondary ones from BAF. The reasoning behind the last one is that if we accept a, we have to accept its supporter c, and thus exclude b. Please note the more complex attacks formally do not appear in any semantics; they are only used to show certain properties. Moreover, while the secondary attack is, in a certain sense, present in the semantics of set AFNs, the last type of extended attack is dropped.

At the core of all AFN semantics are the concepts of acyclicity and coherence:

Definition 2.83. Let $FN = (A, R, N)$ be an argumentation framework with necessities.

An argument $a \in A$ is **necessity–cycle free** (N–Cycle–Free) in FN iff it is not the case that $a\, N^{+} a$ or that jointly $b\, N^{+} a$ and $b\, N^{+} b$. A set of arguments $S \subseteq A$ is N–Cycle–Free in FN iff every $s \in S$ is N–Cycle–Free in FN.

A set $S \subseteq A$ is **coherent** in FN iff it is N–Cycle–Free and closed under N^{-1}, i.e. if $a \in S$ then $b \in S$ for each $b\, N\, a$. A set S is **strongly coherent** in FN iff it is coherent and conflict–free w.r.t. R.

We can now proceed with explaining the semantics:

Definition 2.84. Let $FN = (A, R, N)$ be an AFN. A set of arguments $E \subseteq A$ is:

- **stable** in FN iff it is strongly coherent in FN and for each $a \in A \setminus E$ either $\exists e \in E,\ eR\, a$ or $b\, N\, a$ for some $b \in A \setminus E$.
- **admissible** in FN iff E is strongly coherent in FN and if there is an argument $a \in E$ s.t. bRa, then for each coherent subset $E' \subseteq A \setminus E$ s.t. $b \in E'$, there exist arguments $e \in E, c \in E'$ s.t. eRc.
- **preferred** in FN iff it is maximal w.r.t. set inclusion admissible in FN.

Unfortunately, this formulation of AFN semantics can provide extensions that do not follow the intuitions that the authors have presented in their paper, as visible in Example 14.

Example 14. Let $FN = (\{a, b, c\}, \{(c, b)\}, \{(a, b), (a, c)\})$ be the framework depicted in Figure 13 (we use dotted lines to depict support).

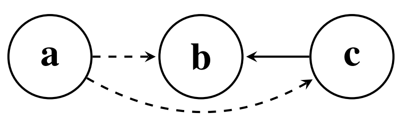


Figure 13: Sample AFN

Intuitively, the admissible extensions should be $\emptyset$, $\{a\}$ and $\{a, c\}$. No set containing b can be considered admissible as it has no way of defending from the attack from c. However, the Definition 2.84 in addition to the sets above considers $\{a, b\}$ admissible:

- $\{a, b\}$ is strongly coherent: it is N–Cycle–Free, includes the necessary supporters and is conflict–free w.r.t. R. [9]
- c attacks $\{a, b\}$. However, $\{a, b, c\} \setminus \{a, b\}$ is simply $\{c\}$ and it is the only set containing c; it is unfortunately not coherent (does not include the necessary supporter a). Hence there is no set we have to attack and the requirements are satisfied.

Due to these complications and the fact that the new version of AFNs does not seem to suffer from such problems, we will not focus on this formulation anymore.

[9] However, please note it is not conflict–free w.r.t. R^{+}.

2.2.2.2 Set AFNs

In [70], apart from AFNs, their generalizations GAFNs were introduced. They allowed support from a set of arguments. In the more recent work [69], the binary version of AFNs was dropped in favor of the set approach completely. Moreover, some of the problems with the previous semantics were fixed. The new formulation is thus as follows:

Definition 2.85. A (set) **abstract argumentation framework with necessities** (AFN) is a tuple (A, R, N) where A is a set of **arguments**, $R \subseteq A \times A$ represents the **attack** relation and $N \subseteq (2^A \setminus \emptyset) \times A$ represents the **necessity** relation.

Although the definitions of N–Cycle–Freeness and being closed under N^{-1} can be adapted to this setting, the concept of a powerful sequence has also been introduced:

Definition 2.86. Let $FN = (A, R, N)$ be an AFN and $E \subseteq A$ a set of arguments. An argument $a \in A$ is **powerful** in E iff $a \in E$ and there is a sequence $a_0, ..., a_k$ of elements of E s.t. :

- $a_k = a$,
- there is no $B \subseteq A$ s.t. BNa_0, and
- for $1 \leq i \leq k$: for each $B \subseteq A$, if BNa_i then $B \cap \{a_0, ..., a_{i-1}\} \neq \emptyset$.

Definition 2.87. Let $FN = (A, R, N)$ be an AFN. A set of arguments $E \subseteq A$ is **coherent** in FN iff each $a \in E$ is powerful in E. A coherent set if **strongly coherent** in FN iff it is conflict–free w.r.t. R in FN.

We will abbreviate these basic semantics with coh and $str\text{-}coh$ for functional representation. We can now continue introducing the usual notions:

Definition 2.88. Let $FN = (A, R, N)$ be an AFN, $E \subseteq A$ and $a \in A$. A set E **defends** a in FN iff $E \cup \{a\}$ is coherent and for each $b \in A$, if bRa then for each coherent $C \subseteq A$ that contains b, there exist arguments $e \in E, c \in C$ s.t. eRc. The **characteristic function** of FN is defined as $\mathcal{F}_{FN} : 2^A \to 2^A$ where $\mathcal{F}_{FN}(E) = \{a \mid E \text{ defends } a \text{ in } FN\}$.

Definition 2.89. Let $FN = (A, R, N)$ be an AFN and $E \subseteq A$ a set of arguments. The set of arguments **deactivated** by E is defined by $E^+ = \{a \mid \exists e \in E$ s.t. eRa or there is a $B \subseteq A$ s.t. BNa and $E \cap B = \emptyset\}$.

Definition 2.90. Let $FN = (A, R, N)$ be an AFN. A set of arguments $E \subseteq A$ is :

- **admissible** in FN iff it is strongly coherent and defends all of its arguments in FN.
- **preferred** in FN iff it is maximal w.r.t. set inclusion admissible in FN.
- **complete** in FN iff it is admissible and contains any argument it defends in FN.

- **grounded** in FN iff it is the least fixed–point of $\mathcal{F}_{FN}$.

- **stable** in FN iff it is complete in FN and $E^+ = A \setminus E$.

Please note that the notion of the deactivated set in AFNs is somewhat weaker than the usual definition of an E^+ set. For example, in AFs, given two sets E and E' s.t. $E \subseteq E'$, it followed that $E^+ \subseteq E'^+$. This is not the case with the deactivated set. Let us show it on an example:

Example 15. Let $FN = (\{a, b, c\}, \emptyset, \{(\{a\}, b), (\{b\}, c)\})$ be an AFN. The deactivated set of $\{a\}$ is $\{c\}$ - c is supported by b, which is not present in the set. However, the deactivated set of $\{a, b\}$ is just $\emptyset$, as c now receives sufficient support form the set.

For this reason, we define an auxiliary notion for AFN that is meant to represent the set of arguments for which all powerful sequences are attacked by a given set E. The fact that this discarded set is a subset of the deactivated one can be shown quite easily based on their definitions.

Definition 2.91. Let $FN = (A, R, N)$ be an AFN. The set of arguments **discarded** by E in FN is defined as $E^{att} = \{a \mid \text{every coherent set containing } a \text{ is attacked by } E\}$.

Lemma 2.92. *Let $FN = (A, R, N)$ be an AFN and $E \subseteq A$ be a strongly coherent set. Then $E^{att} \subseteq E^+$.*

Using the E^{att} set, we can redefine both defense and stability. In particular, we can now use strongly coherent sets instead of complete as a basis for the stable extensions.

Lemma 2.93. *Let $FN = (A, R, N)$ be an AFN, $E \subseteq A$ and $a \in A$. a is defended by E in FN iff $E \cup \{a\}$ is coherent and $\forall b \in A$ s.t. bRa, $b \in E^{att}$.*

Lemma 2.94. *Let $FN = (A, R, N)$ be an AFN. A set $E \subseteq A$ is a stable in FN iff it is strongly coherent and $E^{att} = A \setminus E$.*

Finally, the usual properties between the semantics carry over to AFNs:

Theorem 2.95. *Let $FN = (A, R, N)$ be an AFN. The following holds:*

- *the grounded extension of FN is the least w.r.t. $\subseteq$ complete extension of FN.*

- *a preferred extension of FN is a maximal w.r.t. $\subseteq$ complete extension of FN.*

- *each stable extension of FN is preferred in FN, but not vice versa.*

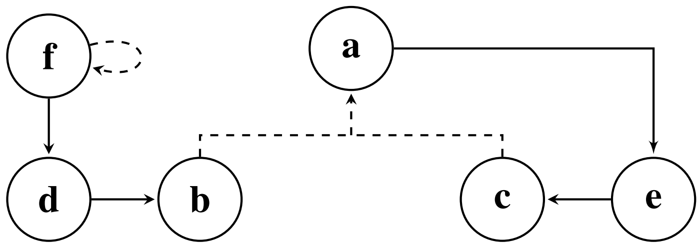


Figure 14: Sample AFN

Example 16. Consider the AFN $(\{a, b, c, d, e, f\}, \{(a, e), (d, b), (e, c), (f, d)\}, \{(\{b, c\}, a), (\{f\}, f)\})$ depicted in Figure 14. The coherent sets include $\emptyset$, $\{a, b\}$, $\{a, c\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$ and any of their combinations. We can observe that f does not appear in any of them - it does not possess a powerful sequence in the framework. The strongly coherent sets are $\emptyset$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$, $\{a, b\}$, $\{a, c\}$, $\{b, c\}$, $\{b, e\}$, $\{c, d\}$, $\{d, e\}$, $\{a, b, c\}$ and $\{a, c, d\}$. $\emptyset$ is trivially admissible. So is $\{d\}$, due to the fact that its only attacker does not posses a coherent set. However, $\{e\}$ is not admissible; it does not attack one of the coherent sets of a, namely $\{a, b\}$. Fortunately, $\{d, e\}$ is already admissible. Due to the fact that no coherent argument can attack d, no strongly coherent set containing b will be admissible. The two final extensions are $\{a, c\}$ and $\{a, c, d\}$; although c is supporting a and a attacks e, the indirect conflict between c and e is not enough to consider c as defending itself in the AFN terms. The sets $\{d\}$, $\{d, e\}$ and $\{a, c, d\}$ are our complete extensions, with the first one being grounded and the latter two being preferred. In this case, both $\{d, e\}$ and $\{a, c, d\}$ are stable.

2.2.3 Evidential Support: Evidential Argumentation Systems

Unattacked arguments serve as the strongest source of defense within AFs. However, in many cases, the lack of an attack is insufficient to consider an argument acceptable. In areas such as legal reasoning and medicine, one is required to support a claim with facts or evidence to be convincing. For example, it does suffice to claim that a given person committed a crime in order to sentence them. Instead, the prosecution has to prove the guilt, by means of evidence. Similarly, medical diagnoses have to be supported by facts such as symptoms or test results.

We can therefore distinguish between two types of arguments. The special arguments, often referred to as *prima facie* or evidence, act as an indisputable source of truth and can be accepted without further restrictions. In contrast, the standard arguments need to be supported by the special ones in order to be considered acceptable. In order to handle such reasoning, the evidential argumentation systems were created. Furthermore, since standard arguments must be supported, evidential frameworks address one of the main drawbacks of abstract support in BAFs [28], namely that an argument could be present in an extension regardless of whether it is supported or not (see [71] for details). In this section we introduce the framework and describe some of its properties, as presented in

[77, 78].

Definition 2.96. An **evidential argumentation system** (EAS) is a tuple (A, R, E) where A is a set of **arguments**, $R \subseteq (2^A \setminus \emptyset) \times A$ is the **attack** relation, and $E \subseteq (2^A \setminus \emptyset) \times A$ is the **support** relation. We distinguish a special argument $\eta \in A$ s.t. $\nexists (x, y) \in R$ where $\eta \in x$; and $\nexists x$ where $(x, \eta) \in R$ or $(x, \eta) \in E$.

We will refer to η as evidence or environment. The core idea of evidential argument systems is that valid arguments (and attackers) need to trace back to the environment. It is captured with the notions of e–support and e–supported attack.

Definition 2.97. Let $ES = (A, R, E)$ be an EAS. An argument $a \in A$ has **evidential support** (e–support) from a set $S \subseteq A$ iff $a = \eta$ or there is a non-empty $S' \subseteq S$ such that $S'Ea$ and $\forall x \in S'$, x has evidential support from $S \setminus \{a\}$.

An argument a has **minimal e–support** from a set S if there is no set $S' \subset S$ such that a has e–support from S'.

Remark. Note that by this definition η has evidential support from any set.

In [77, 78], an alternative way to verify whether an argument is supported by evidence is proposed:

Definition 2.98. Let $ES = (A, R, E)$ be an EAS. Given a set of arguments $X \subseteq A$, a sequence $(a_0, .., a_n)$ of distinct elements of X is an **evidential sequence** for an argument $a \in X$ in ES iff it is the case that $a_n = a$, $a_0 = \eta$, and if $n > 0$, then for $i = 1$ to n there exists a nonempty $T \subseteq \{a_0, ..., a_{i-1}\}$ s.t. TEa_i.

Theorem 2.99. *Let $ES = (A, R, E)$ be an EAS, $X \subseteq A$ a set of arguments and $a \in A$. The argument a is e–supported by X in ES iff there exists an evidential sequence for a on $X \cup \{a\}$ in ES.*

Definition 2.100. Let $ES = (A, R, E)$ be an EAS. A set $S \subseteq A$ carries out an **evidence supported attack** (e–supported attack) on a in ES iff $(S', a) \in R$ where $S' \subseteq S$, and for all $s \in S'$, s has e–support from S in ES.

An e–supported attack by S on a is **minimal** iff there is no $S' \subset S$ that carries out an e–supported attack on a in ES.

Given these notions, we can define semantics for EASs built around the notion of acceptability in a manner similar to those of Dung's. However, in the latter, only the attack relation was considered. For EASs, not only must arguments be defended from attacks, but they must also have sufficient support in order to be acceptable:

Definition 2.101. Let $ES = (A, R, E)$ be an EAS and $S \subseteq A$ a set of arguments. An argument $a \in A$ is **acceptable** w.r.t. S in ES iff

- a is e–supported by S, and

- given a minimal e–supported attack by a set $T \subseteq A$ against a, it is the case that S carries out an e–supported attack against a member of T.

Following the AFN notation, we will also introduce the strongly self–supporting sets, which is meant to represent extensions that are both self–supporting and conflict–free:

Definition 2.102. Let $ES = (A, R, E)$ be an EAS. A set of arguments $S \subseteq A$ is:

- **self–supporting** in ES iff all arguments in S are e–supported by S.
- **conflict–free** in ES iff there is no $a \in S$ and $S' \subseteq S$ such that $S'Ra$.
- **strongly self–supporting** in ES iff it is both self–supporting and conflict–free in ES.
- **admissible** in ES iff it is conflict–free and all elements of S are acceptable w.r.t. S in ES.
- **preferred** in ES iff it is maximal w.r.t. set inclusion admissible in ES.
- **complete** in ES iff it is admissible and all arguments acceptable w.r.t. S in ES are in S.
- **stable** in ES iff it is strongly self–supporting and for any argument a e–supported by A where $a \notin S$, S e–support attacks either a or every set of arguments minimally e–supporting a.

We will shorten self–supporting and its strong version to $ssup$ and $str\text{-}sup$ respectively for functional representation. The rest of the semantics will be abbreviated in the usual manner. Please note that a stable extension can be equivalently described using the notion of a discarded set:

Definition 2.103. Let $ES = (A, R, E)$ be an EAS and $S \subseteq A$ a set of arguments. The **discarded set** of S is defined as $S^+ = \{a \mid$ for every self–supporting set $C \subseteq A$ s.t. $a \in C$, there exists $S' \subseteq S$ and $c \in C$ s.t. $SRc\}$.

Lemma 2.104. *Let $ES = (A, R, E)$ be an EAS and $S \subseteq A$ a set of arguments. Then, S is a stable extension of ES iff it is strongly self–supporting and $A \setminus S = S^+$.*

Just like in the Dung's setting, the grounded semantics is defined via the characteristic function. Also its connection to the admissible and complete extensions holds.

Definition 2.105. Let $ES = (A, R, E)$ be an EAS. The **characteristic function** of ES $\mathcal{F}_{ES} : 2^A \rightarrow 2^A$ is defined as: $\mathcal{F}_{ES}(S) = \{a \mid a$ is acceptable w.r.t. S in $ES\}$. The **grounded extension** of a finitary framework $ES = (A, R, E)$ is the least fixed point of $\mathcal{F}_{ES}$.

Lemma 2.106. *Let $ES = (A, R, E)$ be an EAS. A conflict–free set of arguments $S \subseteq A$ is admissible in ES iff $S \subseteq \mathcal{F}_{ES}(S)$. A conflict–free set S is complete in ES iff $S = \mathcal{F}_{ES}(S)$.*

In AFNs, the coherence of extensions and its role in defense are explicitly stated. Although the use of self–supporting sets is not equivalently stressed in EASs, they are still present in the background:

Lemma 2.107. *Let $ES = (A, R, E)$ be an EAS. If a set of arguments $S \subseteq A$ is a minimal e–support for an argument $a \in A$, then it is self–supporting.*

By using Definition 2.100 and Lemma 2.107, we can connect self–support to e–support attack as well:

Lemma 2.108. *Let $ES = (A, R, E)$ be an EAS. If a set of arguments $S \subseteq A$ carries out a minimal e–supported attacked on an argument $a \in A$, then it is self–supporting.*

Lemma 2.109. *Let $ES = (A, R, E)$ be an EAS. If $S \subseteq A$ is admissible in ES, then it is self–supporting.*

Finally, we can recall the EAS Fundamental Lemma and the relations between the EAS semantics.

Lemma 2.110. ***EAS Fundamental Lemma*** *Let $ES = (A, R, E)$ be an EAS, $S \subseteq A$ an admissible extension of ES and x, y two arguments acceptable w.r.t. S in ES. Then $S \cup \{x\}$ is admissible and y is acceptable w.r.t. $S \cup \{x\}$ in ES.*

Lemma 2.111. *Let $ES = (A, R, E)$ be an EAS. A set $S \subseteq A$ is an e–stable extension of ES iff $S = \{a \mid a$ is not e–support attacked by S and is e–supported by $S\}$.*

Theorem 2.112. *Let $ES = (A, R, E)$ be an EAS. The following holds:*

- *every stable extension of ES is a preferred extension, but not vice versa.*
- *every preferred extension of ES is a complete extension, but not vice versa.*
- *the grounded extension of ES is the least w.r.t. set inclusion complete extension of ES.*

Example 17. Let $(\{\eta, a, b, c, d, e, f\}$, $\{(\{b\}, a)$, $(\{b\}, c)$, $(\{c\}, b)$, $(\{c\}, d)$, $(\{d\}, f)$, $(\{f\}, f)\}$, $\{(\{\eta\}, b)$, $(\{\eta\}, c)$, $(\{\eta\}, d)$, $(\{\eta\}, f)$, $(\{d\}, e)\})$ be the EAS depicted in Figure 15. The admissible extensions are $\emptyset$, $\{\eta\}$, $\{\eta, b\}$, $\{\eta, c\}$, $\{\eta, b, d\}$ and $\{\eta, b, d, e\}$, with $\{\eta\}$, $\{\eta, c\}$ and $\{\eta, b, d, e\}$ being the complete ones. Obviously, the latter two are preferred. However, only $\{\eta, b, d, e\}$ is stable. Since a is not a valid argument (it is not e–supported in the framework), we do not have to attack it. Although $\{\eta, c\}$ attacks b and d (and by this, also e), it is not in any way in conflict with f. The grounded extension is just $\{\eta\}$.

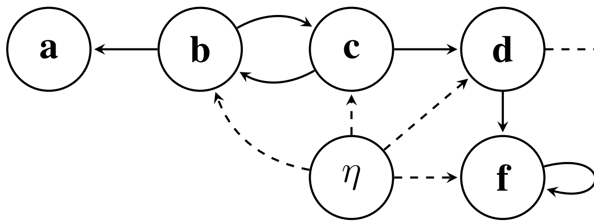


Figure 15: Sample EAS

2.3 Abstract Dialectical Frameworks

Abstract dialectical frameworks have been defined in [23] and till today various results as to their semantics, instantiation and complexity have already been published in [21, 79, 84, 85, 87]. Although they can be seen as a type of framework with support, their design differs from the traditional construction of argumentation frameworks. The main goal of ADFs is to be able to express arbitrary relations and avoid the need of extending AFs by a new relation sets each time they are needed. This is achieved by the means of acceptance conditions, which define what sets of arguments related to a given argument should be present for it to be accepted or rejected. In a certain sense, this form of representation can be seen as dual to the one normally seen in argumentation frameworks. Instead of focusing on separate links and saying "a supports b" or "c attacks b" and then checking if e.g. there are no attacks in a given set of arguments, we look at collections of arguments related to the one we are interested in and just evaluate the acceptance condition to say "with respect to this set of arguments, this argument can(not) be accepted". Only by checking the outcomes of the conditions we can later say that "the relation between a and b is attacking" and so on.

Definition 2.113. An **abstract dialectical framework** (ADF) is a tuple (A, L, C), where A is a set of abstract **arguments** (nodes, statements), $L \subseteq A \times A$ is a set of **links** (edges) and $C = \{C_a\}_{a \in A}$ is a set of **acceptance conditions**, one condition per each argument. An acceptance condition is a total function $C_a : 2^{par(a)} \rightarrow \{in, out\}$, where $par(a) = \{p \in A \mid (p, a) \in L\}$ is the set of parents of an argument a.

Within ADFs, we distinguish a particular subclass called bipolar. It is particularly valuable due to the fact bipolar ADFs appear to be of lower complexity than general ones [87].

Definition 2.114. Let $D = (A, L, C)$ be an ADF. A link $(r, s) \in L$ is:

- **supporting** iff for no $R \subseteq par(s)$ we have that $C_s(R) = in$ and $C_s(R \cup \{r\}) = out$.
- **attacking** iff for no $R \subseteq par(s)$ we have that $C_s(R) = out$ and $C_s(R \cup \{r\}) = in$.

An ADF is **bipolar** (BADF for short) iff it contains only links that are supporting or attacking.

Remark. Please note that links can be both attacking and supporting (in which they are also often called redundant), or neither – ADFs are able to express more than attack and support.

We can also represent the acceptance conditions by propositional formulas over arguments instead of "Boolean" functions [44]. In this case the condition C_a for an argument $a \in A$ is a propositional formula φ_a over the parents of a. Moreover, it is easy to see that links L are somewhat redundant and can be extracted from the conditions. Thus, we will use of shortened notation and assume an ADF $D = (A, C)$ through the rest of this paper. In order to recall the ADF semantics, we need to explain some basic notions first.

2.3.1 Interpretations and Decisiveness

Interpretations will be equally important both in labeling and extension–based semantics. While in the first case the interpretations will be returned instead of sets of arguments, in the latter they will be used to store accepted and rejected arguments in order to determine their acceptability.

Please note that particularly in the propositional descriptions of ADFs, we can occasionally observe a certain inconsistency in the notation, where the condition outcomes in and out are interchangeably used with truth values $\mathbf{t}$ and $\mathbf{f}$ of the propositional formulas. The reason why the conditions were not assigned the truth values from the very beginning was the need to distinguish between the status of the condition of an argument and the value a given argument is assigned in e.g. a labeling. However, since for any semantics the truth assignment has to be in accordance with the condition (i.e. in paired with $\mathbf{t}$, out with $\mathbf{f}$), this abuse of notation is not overly problematic.

A two (three–valued) interpretation is simply a mapping that assigns truth values (respectively $\{\mathbf{t}, \mathbf{f}\}$ and $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$) to arguments. We will be making use both of partial (i.e. defined only for a subset of A) and full ones (defined for all elements of A). The truth values can be compared with respect to truth ordering, i.e. $\mathbf{f} \leq_t \mathbf{u} \leq_t \mathbf{t}$, or precision (information) ordering: $\mathbf{u} \leq_i \mathbf{t}$ and $\mathbf{u} \leq_i \mathbf{f}$. The latter will be used in the context of labeling semantics. The pair $(\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}, \leq_i)$ forms a complete meet–semilattice with the meet operation $\sqcap$ assigning values in the following way: $\mathbf{t} \sqcap \mathbf{t} = \mathbf{t}$, $\mathbf{f} \sqcap \mathbf{f} = \mathbf{f}$ and $\mathbf{u}$ in all other cases. It can naturally be extended to interpretations: given two interpretations v and v' on A, we say that v' contains more information, denoted $v \leq_i v'$, iff for every argument $s \in A$, $v(s) \leq_i v'(s)$. In the case v is three and v' two–valued, we say that v' extends v. This means that the elements mapped originally to $\mathbf{u}$ are now assigned either $\mathbf{t}$ or $\mathbf{f}$. The set of all two–valued interpretations extending v is denoted $[v]_2$. The meet operation can be adjusted in a similar fashion as the information ordering, i.e. the meet of two interpretations $v \sqcap v'$ is an interpretation obtained by assigning to a given argument a the value $v(a) \sqcap v'(a)$.

Example 18. Let $v = \{a : \mathbf{t}, b : \mathbf{t}, c : \mathbf{f}, d : \mathbf{u})$ be a three–valued interpretation. We have two extending interpretations, namely $v' = \{a : \mathbf{t}, b : \mathbf{t}, c : \mathbf{f}, d : \mathbf{t})$ and $v'' = \{a :$

$\mathbf{t}, b : \mathbf{t}, c : \mathbf{f}, d : \mathbf{f})$. Clearly, it holds that $v \leq_i v'$ and $v \leq_i v''$. However, v' and v'' are incomparable w.r.t. $\leq_i$.

Let now $w = \{a : \mathbf{f}, b : \mathbf{f}, c : \mathbf{f}, d : \mathbf{t})$ be another three–valued interpretation. The meet of v and w gives us a new interpretation $w' = \{a : \mathbf{u}, b : \mathbf{u}, c : \mathbf{f}, d : \mathbf{u})$: as the assignments of a, b and d differ between v and w, the resulting value is $\mathbf{u}$. On the other hand, c is in both cases $\mathbf{f}$ and thus retains its value.

We will use v^x to denote a set of arguments mapped to x by v, where x is a given truth–value.

The notion of decisiveness is a key concept in our extension–based semantics for abstract dialectical frameworks. Let us assume an ADF $D = (A, C)$. Given an acceptance condition C_s for an argument $s \in A$ and an interpretation v, we define a shorthand $v(C_s)$ as $C_s(v^{\mathbf{t}} \cap par(s))$. For a given propositional formula φ and an interpretation v defined over all of the atoms of the formula, $v(\varphi)$ will just stand for the value of the formula under v. However, apart from knowing the "current" value of an acceptance condition for a given interpretation, we would also like to know if this interpretation is "final". By this we understand that no new information will cause the value to change. For example, given a condition $\varphi_s = a \wedge \neg b$ for an argument s dependent on a and b, knowing that b is true is enough to map φ_s to out in a way that no matter the value of a, it will always stay out. In order to verify whether our interpretation is decisive for a given argument, we will explore how the interpretations "filling in" the missing values evaluate the argument's condition. We will refer to them as completions:

Definition 2.115. Let A be a collection of elements, $E \subseteq A$ its subset and v a two–valued interpretation defined on E. A **completion** of v to a set Z where $E \subseteq Z \subseteq A$, is an interpretation v' defined on Z in a way that $\forall a \in E \; v(a) = v'(a)$. v' is a $\mathbf{t}/\mathbf{f}$ completion of v iff all arguments in $Z \setminus E$ are mapped respectively to $\mathbf{t}/\mathbf{f}$.

Remark. By the abuse of notation we will also talk about $\mathbf{u}$–completions when comparing extension and labeling–based approaches. It should be understood as a three–valued interpretation that assigns $\mathbf{u}$ to the "missing" mappings of a given two–valued interpretation.

We would like to draw the attention to the similarity between the concepts of completion and extending interpretation. Basically, given a three–valued interpretation v defined over A, the set $[v]_2$ corresponds precisely to the set of completions to A of the two–valued part of v. However, if we used the notion of an extension instead of a completion in a two–valued setting, it could be easily mistaken for the extension understood as set of arguments, not as an interpretation. Therefore, we will use our notation to avoid such collisions.

Definition 2.116. Let $D = (A, C)$ be an ADF, $E \subseteq A$ a set of arguments and v a two–valued interpretation defined on E. v is **decisive** for an argument $s \in A$ iff for any two completions $v_{par(s)}$ and $v'_{par(s)}$ of v to $E \cup par(s)$, it holds that $v_{par(s)}(C_s) = v'_{par(s)}(C_s)$. s is **decisively out/in** w.r.t. v if v is decisive and all of its completions evaluate C_s to respectively out, in.

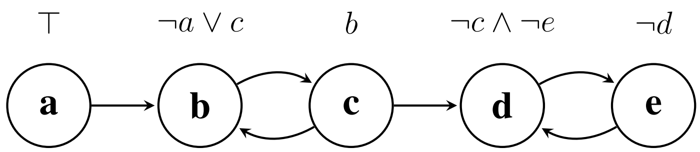


Figure 16: Sample ADF

Example 19. Let $(\{a, b, c, d, e\}, \{C_a = \top, C_b = \neg a \vee c, C_c = b, C_d = \neg c \wedge \neg e, C_e = \neg d\})$ be the ADF in Figure 16. Examples of decisively in interpretations for b include $v_1 = \{c : \mathbf{t}\}$. This means that knowing that c is true, we know that the whole disjunction (and thus the acceptance condition) are satisfied. Formally speaking, v_1 is decisive as both of its completions $\{c : \mathbf{t}, a : \mathbf{f}\}$ and $\{c : \mathbf{t}, a : \mathbf{t}\}$ satisfy the condition.

Remark. Please note that the existence of an interpretation that satisfies the acceptance condition of an argument a (i.e. there is a set of parents s.t. the condition is in) implies the existence of a decisively in interpretation for a and vice versa. Moreover, if an argument is decisively out/in w.r.t. an interpretation, it holds that its acceptance condition is out/in. It basically results from the definition of the completion and decisiveness. Finally, if an argument a is decisively in/out w.r.t. a given interpretation, then it is decisively out w.r.t. any of its completions, not necessarily the ones that are defined for all parents of a. For example, given a condition $C_a = b \vee c \vee d$ and an interpretation $v = \{b : \mathbf{t}\}$, then a is decisively in w.r.t. not only v, but $v' = \{b : \mathbf{t}, c : \mathbf{f}\}$ and $v'' = \{b : \mathbf{t}, c : \mathbf{f}, d : \mathbf{t}\}$ as well.

2.3.2 Evaluations and Acyclicity

Acceptance conditions tell us on what other arguments a given argument depends. We can see if they need to be accepted or rejected for the condition to be in our out and derive a range of decisively in interpretation based on it. We can then focus on the arguments in the condition and investigate them in a similar manner and continue this process until we have a full picture telling us when, how, and if at all, the arguments can be accepted or rejected, if they can be derived from initial arguments (i.e. those with acceptance condition equivalent to $\top$), include cyclic dependencies and so on. To this end, we have introduced the notions of positive dependency functions and evaluations [75]. We choose to call it positive dependencies rather than support in order not to confuse them with the notions of attack and support links from BADFs and not to point to any particular interpretation of support.

In the majority of the argumentation frameworks, the nature of a relation between the arguments is stated openly in the structure of the framework, i.e. R is the attack, N is the support and so on. This is not the case in ADFs and in order to obtain the arguments that are required or should be avoided for the acceptance of a given argument, we will make use of decisive interpretations. Naturally, it suffices to focus on the minimal ones, by which we understand that both $v^{\mathbf{t}}$ and $v^{\mathbf{f}}$ are minimal w.r.t. $\subseteq$. By $min\_dec(x, s)$ we will denote the set of minimal two–valued interpretations that are decisively x for s, where s is an argument and $x \in \{in, out\}$. We will explain this choice at the end of this section.

First of all, let us recall the concept of a positive dependency function. It basically maps every argument to one of its minimal decisively in interpretations contained in a given set:

Definition 2.117. Let $D = (A, C)$ be an ADF and $E \subseteq A$ a set of arguments. A **positive dependency function** on E is a function pd_E^D assigning every argument $a \in E$ an interpretation $v \in min\_dec(in, a)$ s.t. $v^t \subseteq E$ or $\mathcal{N}$ for null iff no such interpretation can be found. The function is **sound** iff no argument is mapped to $\mathcal{N}$. pd_E^D is maximally sound on E iff it is a sound function on $E' \subseteq E$ and there is no sound positive dependency function pd'^D_E on E'', where $E' \subset E'' \subseteq E$, s.t. $\forall a \in E'$, $pd_E^D(a) = pd'^D_E(a)$.

We will now trace the arguments that a given argument requires for its acceptance by the use of dependency evaluations, within which we can distinguish standard, acyclic and partially acyclic ones. While the last type might seem confusing, they will prove to be valuable in translating ADFs into AFs and SETAFs (see Translations 85, 86 and 87).

Definition 2.118. Let $D = (A, C)$ be an ADF, $X \subseteq A$ and pd_E^D a maximally sound positive dependency function of X defined over $E \subseteq X$. A **standard positive dependency evaluation** for an argument $e \in E$ in D based on pd_E^D is a pair (F, B), where $F \subseteq E$ is a set of arguments s.t. $e \in F$, and $\forall a \in F$, $pd_E^D(a)^{\mathbf{t}} \subseteq F$, and $B = \bigcup_{a \in F} pd_E^D(a)^{\mathbf{f}}$.

We will refer to F as the **pd–set** of the evaluation and to B as the **blocking set** of the evaluation.

Example 20. Let $(\{a, b, c, d, e\}, \{C_a = \bot, C_b = a \wedge c, C_c = d \wedge \neg e, C_d = d, C_e = \top\})$ be the ADF depicted in Figure 17. The argument a has no standard evaluation, as it possesses no decisively in interpretation to start with. Although the argument b has a decisively in interpretation $\{a : \mathbf{t}, c : \mathbf{t}\}$, it depends on a and thus there does not exist a sound pd–function from which we could construct an evaluation for b. For d we have a simple evaluation $(\{d\}, \emptyset)$, and based on it an evaluation $(\{c, d\}, \{e\})$ for c. Finally, e as an initial argument has a trivial evaluation $(\{e\}, \emptyset)$.

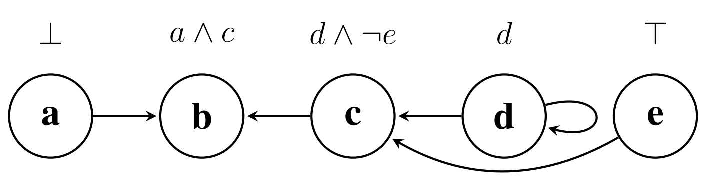


Figure 17: Sample ADF

While standard evaluations are already quite useful, we will also be interested in the more specialized types, dealing with the issue of positive dependency cycles. The informal understanding of a cycle is simply whether acceptance of an argument depends on this argument. First of all, we will consider the partially acyclic evaluations. They can be seen as refinement of the standard ones, where the arguments are separated into two groups; one that can be ordered into a sequence s.t. each argument depends only on the predecessors, and the other for which it is not possible, thus serving as a container for the cycles.

Definition 2.119. Let $D = (A, C)$ be an ADF, $X \subseteq A$ and pd_E^D a maximally sound positive dependency function of X defined over $E \subseteq X$.

A **partially acyclic positive dependency evaluation** based on pd_E^D for an argument $x \in E$ is a triple $(F, (a_0, ..., a_n), B)$, where $F \cap \{a_0, ..., a_n\} = \emptyset$, $(a_0, ..., a_n)$ is a sequence of distinct elements of E satisfying the requirements:

- if the sequence is non–empty, then $a_n = x$; otherwise, $x \in F$,
- $\forall_{i=1}^n$, $pd_E^D(a_i)^{\mathbf{t}} \subseteq F \cup \{a_0, ..., a_{i-1}\}$, $pd_E^D(a_0)^{\mathbf{t}} \subseteq F$,
- $\forall a \in F$, $pd_E^D(a)^{\mathbf{t}} \subseteq F$, and
- $\forall a \in F, \exists b \in F$ s.t. $a \in pd_E^D(b)$.

Finally, $B = \bigcup_{a \in F} pd_E^D(a)^{\mathbf{f}} \cup \bigcup_{i=0}^n pd_E^D(a_i)^{\mathbf{f}}$. The sequence part of the evaluation will be referred to as the **pd–sequence**.

We can now introduce the last type of evaluations: the acyclic ones, being a subclass of partially acyclic. It simply requires the “cycle container” to be empty.

Definition 2.120. Let $D = (A, C)$ be an ADF, $X \subseteq A$ and pd_E^D a maximally sound positive dependency function of X defined over $E \subseteq X$. A partially acyclic evaluation $(F, (a_0, ..., a_n), B)$ for an argument $x \in E$ is an **acyclic positive dependency evaluation** for x iff $F = \emptyset$. A set of arguments $E \subseteq A$ is **pd–acyclic** iff every argument $a \in E$ possesses an acyclic pd–evaluation on this set.

We will use the shortened notation $((a_0, ..., a_n), B)$ in order to denote the acyclic evaluations. We can also observe that the pd–acyclic sets are not unlike coherent and self–supporting ones; we will abbreviate them with pdc. Furthermore, we will simply write that an argument has a given type of evaluation on E if there is some pd–function on E from which we can produce such an evaluation.

We will say a standard evaluation (F, B) based on pd_E^D can be made acyclic for an argument $e \in F$ and w.r.t. pd_E^D iff there exists a way to order the elements of F into a sequence satisfying the pd–sequence requirements for e. It is also easy to see that any evaluation can be transformed into a standard one by joining the pd–set and the pd–sequence into a single pd–set.

Example 21. Let us come back to the framework $(\{a, b, c, d, e\}, \{C_a = \bot, C_b = a \wedge c, C_c = d \wedge \neg e, C_d = d, C_e = \top\})$ from Example 20 and Figure 17. The standard evaluation for e was $(\{e\}, \emptyset)$. Since e does not depend on any other argument, it can be easily moved into the pd–sequence and the partially acyclic representation of the standard evaluation is $(\emptyset, (e), \emptyset)$. This evaluation also happens to be acyclic. Although the evaluation for d looks similar, we can observe that the argument depends on itself, and thus the pd–sequence will be empty. The partial representation is thus $(\{d\}, (), \emptyset)$. Finally, let us look at the evaluation for c. The evaluation $(\{c, d\}, (), \emptyset)$ would not satisfy the partially acyclic requirements, since no argument in the pd–set depends on c. Consequently, we can "push" c into the sequence and obtain the evaluation $(\{d\}, (c), \{e\})$, which clearly shows where the actual cycle occurs. Neither c nor d possess acyclic evaluations.

There are two ways we can "attack" an evaluation. Either we accept an argument that needs to be rejected in order for the evaluation to hold (i.e. it is in the blocking set), or we are able to discard an argument from the pd–sequence or the pd–set. This leads to the following, more abstract formulation: [10]

Definition 2.121. Let $D = (A, C)$ be an ADF and $(F, (a_0, ..., a_n), B)$ a partially acyclic evaluation on a set $E \subseteq A$ for an argument $a \in E$. A two–valued interpretation v defined on a subset of A **blocks** $(F, (a_0, ..., a_n), B)$ iff $\exists b \in B$ s.t. $v(b) = \mathbf{t}$ or $\exists x \in \{a_0, ..., a_n\} \cup F$ s.t. $v(x) = \mathbf{f}$.

Remark. An evaluation can be self–blocking, i.e. some members of the pd–sequence or the pd–set are present in the blocking set. Although an evaluation like that will never be accepted in an extension, it can make a difference in what we consider a valid attacker.

Example 22. Recall the framework $(\{a, b, c, d, e\}, \{C_a = \top, C_b = \neg a \vee c, C_c = b, C_d = \neg c \wedge \neg e, C_e = \neg d\})$ from Example 19. For the argument b there exist two minimal decisively in interpretations: $v_1 = \{a : \mathbf{f}\}$ and $v_2 = \{c : \mathbf{t}\}$. The interpretations for a and c are respectively $w_1 = \emptyset$ and $z_1 = \{b : \mathbf{t}\}$. Therefore, on $E = \{a, b, c\}$ we have two pd–functions, namely $pd_1 = \{a : w_1, b : v_1, c : z_1\}$ and $pd_2 = \{a : w_1, b : v_2, c : z_1\}$. If we focus on acyclic evaluations with the minimal w.r.t. $\subseteq$ sequences, we obtain one acyclic evaluation for a: $((a), \emptyset)$, one for b: $((b), \{a\})$ and one for c: $((b, c), \{a\})$. Let us now analyze E. We can observe that accepting a "forces" a cycle between b and c; we thus look for a method that would detect the cycle. The acceptance conditions of all arguments are satisfied, thus this simple check is not enough to verify if it occurs. Also the pd–sequences of all arguments are contained in the set, thus the sequence check resembling the one in AFNs or EAFs is also insufficient (see Sections 2.2.2 and 2.2.3). Only looking at the whole evaluations shows us that b and c are both blocked by a through the blocking set. Although b and c are technically pd–acyclic in E, we see that their evaluations are in fact blocked and this type of conflict needs to be taken into account by the semantics.

We would now like to discuss the minimal interpretations and evaluations. Allowing every type of interpretation would not affect our semantics, as we are mostly interested in

[10] Since every standard evaluation can be made partially acyclic and every acyclic evaluation is also a partial one, we will only present the most general definition.

the existence of an unblocked evaluation of a given type or in blocking all evaluations. Existence of an unblocked evaluation built with arbitrary interpretations implies the existence of an evaluation built with minimal ones. We can always "remove" unnecessary elements from an interpretation in order to trim it to minimal one. Moreover, if all evaluations are blocked, then so are the ones constructed with the minimal interpretations. However, using non–minimal interpretations can introduce "fake cycles", i.e. show that a cycle exists even if it is not the case. Consequently, if we want to ensure that e.g. every evaluation that can be made acyclic (see Definition 2.171), minimality makes a difference.

Example 23. Let us consider a simple ADF $(\{a, b\}, \{C_a = \top, C_b = \top\})$. Both a and b possess a single minimal decisively in interpretation that is just empty. However, if we consider non–minimal ones, we would e.g. get interpretations $\{b : \mathbf{t}\}$ and $\{a : \mathbf{t}\}$ for a and b respectively. A standard evaluation constructed with them cannot be made acyclic and thus we get a false answer that there is a cycle in our framework. We can now argue that these interpretations go beyond the parents of the arguments. However, limiting ourselves to interpretations defined only for parents does not fix this issue. Consider a small modification of our ADF: $(\{a, b\}, \{C_a = \top \vee b, C_b = \top \vee a\})$. We get the same interpretations and evaluations as in the previous case, but we can observe that the links from b to a and a to b are redundant, i.e. presence of one argument never affects the outcome of the acceptance condition of the other. Although we can argue that there is a cycle on the links as such, it should clearly be disregarded due to its inability to affect the arguments.

We can introduce the concept of minimal evaluations. After all, not every evaluation may be of interest to us. For example, it may contain redundant elements on which the argument of interest does not really depend, or they may unnecessarily long. Let us consider an example.

Example 24. Let $(\{a, b, c, d\}, \{C_a = b \vee c, C_b = c, C_c = \top, C_d = \neg b \vee c\})$ be the ADF depicted in Figure 18. Let us focus on argument a and the following three acyclic evaluations for it: $((c, a), \emptyset)$, $((c, b, a), \emptyset)$ and $((d, c, a), \{b\})$. We can observe that a depends on c, but it can be reached either directly or though b. Although the "longer" part is perfectly fine, it can be seen as somewhat redundant due to the presence of a shorter route. Furthermore, the $((d, c, a), \{b\})$ evaluation contains data useless for a – it includes the analysis of argument d, which is not related to a at all.

Let us now consider argument d and its evaluations. It possesses an acyclic one $((d), \{b\})$, which can also be changed into standard, and a purely standard one $(\{d\}, \emptyset)$. While the first one is created with the decisively in interpretation $\{b : \mathbf{f}\}$, the other with $\{d : \mathbf{t}\}$. If we were to consider minimal evaluations based only on subset relations between pd–sets and blocking sets, we can observe that the standard evaluation corresponding to $((d), \{b\})$ would have been "lost". Thus, in this approach a minimal evaluation of one type may not necessarily be a minimal one of another type. While it does not create problems if we are trying to answer the question if all standard evaluations of an argument are blocked, it can make a difference if we distinguish between types of evaluations, like in the case of

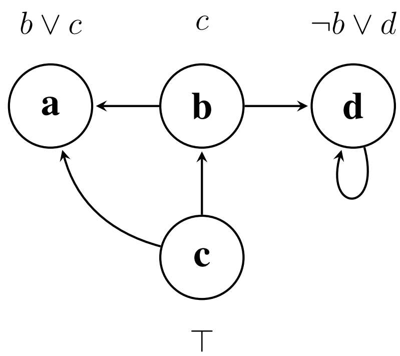


Figure 18: Sample ADF

Translation 83. Consequently, a safe approach to minimality should take the pd–function into account.

We close this section by formally defining minimal evaluations. Please note that we will focus on minimality w.r.t. a given argument, not for the whole framework itself. Due to the fact that from e.g. an acyclic evaluation we can always extract shorter evaluations for arguments earlier in the sequence, assuming global minimality would not be particularly informative.

Definition 2.122. Let $D = (A, C)$ be an ADF and pd_E^D a positive dependency function on a set $E \subseteq A$. Let $a \in E$ and (F, B) a standard evaluation for $a \in E$. (F, B) is a **minimal standard evaluation** for a w.r.t. pd_E^D if there is no other standard evaluation (F', B') for a based on pd_E^D s.t. $F' \subseteq F$ and $B' \subseteq B$.

Let (G, B) be an acyclic pd–evaluation for $a \in E$ based on pd_E^D. (G, B) is a **minimal acyclic pd–evaluation** for a w.r.t. pd_E^D if there is no other acyclic pd–evaluation (G', B') for a based on pd_E^D s.t. $B' \subseteq B$ and G' is a subsequence of G.

Let (F, G, B) be a partially acyclic pd–evaluation for $a \in E$ based on pd_E^D. (F, G, B) is a **minimal partially acyclic pd–evaluation** for a w.r.t. pd_E^D if there is no other partially acyclic pd–evaluation (F', G', B') for a based on pd_E^D s.t. $B' \subseteq B$, $F' \subseteq F$ and G' is a subsequence of G.

2.3.3 Standard, Acyclic and Partially Acyclic Range

Just like in the Dung's framework, the concept of range and the E^+ set also appears in ADFs. The original definition from [74] required the notion of conflict–freeness. We will recall it here and later show that with the use of evaluations, we can drop the conflict–freeness assumption. For more explanations and examples concerning this semantics, please refer to Section 2.3.5.

Definition 2.123. Let $D = (A, C)$ be an ADF. A set of arguments $E \subseteq A$ is a **conflict–free extension** of D if for all $s \in E$ we have $C_s(E \cap par(s)) = in$. E is a **pd–acyclic**

conflict–free extension of D iff for every argument $a \in E$, there exists an unblocked acyclic pd–evaluation on E w.r.t. v^E.

The basic concept of range is based on decisive outing. We start with the arguments we can accept and then look for ones that are decisively outed by our choice. Since discarding one argument can also discard another that depends on it via a chain reaction, we repeat this search until no further arguments can be found.

Definition 2.124. Let $D = (A, C)$ be an ADF, $E \subseteq A$ a conflict–free extension of D and v_E a partial two–valued interpretation built as follows:

1. let $M = E$ and for every $a \in E$ set $v_E(a) = \mathbf{t}$,
2. for every argument $b \in A \setminus M$ that is decisively out w.r.t. v_E, set $v_E(b) = \mathbf{f}$ and add b to M, and
3. now repeat the previous step until there are no new elements added to M.

By E^+ we understand the set of arguments $v_E^{\mathbf{f}}$ and we will refer to it as the **discarded set**. v_E now forms a **range interpretation** of E, where the usual range is denoted as E^{Ran} and equals $E \cup E^+$.

We can also redefine this notion by the use of standard evaluations, which limits the algorithm to a single iteration. Moreover, it allows us to find arguments decisively outed by a set of arguments without the conflict–freeness assumption.

Lemma 2.125. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a set of arguments and $X = \{a \in A \mid$ for every standard dependency evaluation (F, B) for a in D, $B \cap E \neq \emptyset\}$. If E is conflict–free, then $X = E^+$.*

The notions of the discarded set and the range are quite strong in the sense that they require an explicit "attack" on arguments that take part in dependency cycles. This is not always a desirable property. Depending on the approach we might not treat cyclic arguments as valid and hence want them "out of the way". The original definition is as follows:

Definition 2.126. Deprecated Let $D = (A, C)$ be an ADF, $E \subseteq A$ a conflict–free extension of D and v_E^a a partial two–valued interpretation built as follows:

1. let $M = E$. For every $a \in M$ set $v_E^a(a) = \mathbf{t}$,
2. for every argument $b \in A \setminus M$ s.t. every acyclic pd–evaluation of b in A is blocked by v_E^a, set $v_E^a(b) = \mathbf{f}$ and add b to M, and
3. repeat the previous step until there are no new elements added to M.

By E^{a+} we understand the set of arguments mapped to $\mathbf{f}$ by v_E^a and refer to it as **acyclic discarded set** of E. We refer to v_E^a as **acyclic range interpretation** of E.

However, it turns out that with this iterative approach, it suffices to focus on blocking an evaluation through the blocking set only:

Lemma 2.127. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ be a pd–acyclic conflict–free extension of D, v_E^a its acyclic range interpretation and $a \in A$ an argument s.t. it has at least one acyclic pd–evaluation $((a_0, ..., a_n), B)$ on A. The interpretation v_E^a blocks the evaluation iff $E \cap B \neq \emptyset$.*

The analysis above brings us to a conclusion that the algorithm from the original definition of the acyclic range in fact terminates after the first iteration. Consequently, we can rephrase it in the following way, similar to Lemma 2.125:

Lemma 2.128. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a set of arguments and $X = \{a \in A \mid$ for every acyclic pd–evaluation (F, B) for a, $B \cap E \neq \emptyset\}$. If E is pd–acyclic conflict–free, then $X \cap E = \emptyset$. If E is conflict–free, then $X \setminus E = E^+$.*

The last type of range we will consider, the partially acyclic one, will be used in one family of our semantics. It can be seen as a certain middle ground between the standard and acyclic range. We discard the arguments if we block all of its acyclic pd–evaluations, unless it is based on a "cycle" that we are ready to accept.

Definition 2.129. Let $D = (A, C)$ be an ADF and $E \subseteq A$ a set of arguments. The **partially acyclic discarded set** of E is $E^{p+} = \{a \in A \mid$ there is no partially acyclic evaluation (F', G', B') for a s.t. $F' \subseteq E$ and $B' \cap E = \emptyset\}$.

Lemma 2.130. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a set of arguments and E^{p+} its partially acyclic discarded set. If E is conflict–free in D, then $E \cap E^{p+} = \emptyset$.*

Definition 2.131. Let $D = (A, C)$ be an ADF, $E \subseteq A$ a conflict–free extension of D and E^{p+} its partially acyclic discarded set. The **partially acyclic range** of E in the interpretation v_E^p mapping to $\mathbf{t}$ all and only arguments in E and mapping to $\mathbf{f}$ all and only arguments in E^{p+}.

We can observe that there is a subset relation between the three versions of the discarded set:

Lemma 2.132. *Let $D = (A, C)$ be an ADF and $E \subseteq A$ a conflict–free extension of D. Then $E^+ \subseteq E^{p+} \subseteq E^{a+}$. If E is pd–acyclic conflict–free, then $E^{p+} = E^{a+}$.*

Example 25. Let us consider the framework $(\{a, b, c, d, e\}, \{C_a = a \land \neg b, C_b = a, C_c = \neg b, C_d = \neg a, C_e = d, C_f = f\})$ depicted in Figure 19 and focus on the conflict–free set $\{a\}$. We will now compute its standard range. First of all, the interpretation $v = \{a : \mathbf{t}\}$ decisively outs d. We update v and now have $\{a : \mathbf{t}, d : \mathbf{f}\}$. Our new interpretation now decisively outs e and we can extend it to $\{a : \mathbf{t}, d : \mathbf{f}, e : \mathbf{f}\}$. No further arguments can be falsified, as for both b and c the conditions are in w.r.t. $\{a\}$ and even though the condition of f is for now out, a completion of v mapping f to $\mathbf{t}$ can make it in. Let us now compute the standard range in the evaluation manner. For b we have an evaluation $(\{a, b\}, \{b\})$, for c $(\{c\}, \{b\})$, for d $(\{d\}, \{a\})$, $(\{d, e\}, \{a\})$ for e and finally $(\{f\}, \emptyset)$ for f. We can observe that only the evaluations for d and e are blocked by $\{a\}$. In any case, the standard range of the set $\{a\}$ is $v = \{a : \mathbf{t}, d : \mathbf{f}, e : \mathbf{f}\}$

Let us now consider the acyclic range. The evaluations for e and d can be made acyclic, and as their blocking sets contain a, it is easy to see that both of the arguments will also be falsified in the acyclic range. Since f possesses no acyclic evaluation, it will also be in the discarded set. Finally, the evaluation $(\{a, b\}, \{b\})$ for b cannot be made acyclic and the argument will be falsified for the same reason as f. Therefore, the acyclic range of $\{a\}$ is $w = \{a : \mathbf{t}, b : \mathbf{f}, d : \mathbf{f}, e : \mathbf{f}, f : \mathbf{f}\}$.

In the partially acyclic case, the arguments d, e and f will also be mapped to $\mathbf{f}$ by the range. However, even though argument b does not possess an acyclic evaluation, the partially acyclic representation $(\{a\}, (b), \{b\})$ of the standard one $(\{a, b\}, \{b\})$ has its pd–set contained in $\{a\}$. Consequently, the argument does not meet the partially acyclic range requirements.

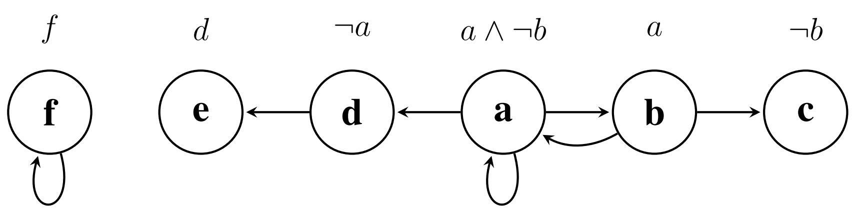


Figure 19: Sample ADF

2.3.4 Labeling–Based Semantics of ADFs

The two approaches towards labeling–based semantics of ADFs were developed in [21, 84]. They are based on the notion of a characteristic operator. While in the Dung's setting the operator worked with sets, here three valued interpretations are used.

Definition 2.133. Let $D = (A, C = \{\varphi_a\}_{a \in A})$ be an ADF and V_A be the set of all three–valued interpretations defined on A, a an argument in A and v an interpretation in V_S. The **three–valued characteristic operator** of D is a function $\Gamma_D : V_A \to V_A$ s.t. $\Gamma_D(v) = v'$ with $v'(a) = \prod_{w \in [v]_2} w(\varphi_a)$.

Recall that verifying the value of an acceptance condition under a set of extensions of a three–valued interpretation $[v]_2$ is just like testing its value against the completions of the two–valued part of v. Thus, an argument that is **t**/**f** in $\Gamma_D(v)$ is decisively in/out w.r.t. the two–valued sub–interpretation of v (see also Theorems 2.148 and 2.149).

Remark. It is easy to see that in a certain sense this operator allows self–justification and self–falsification. Take, for example, a self–supporter; if we generate an interpretation in which it is false then, obviously, it will remain false. Same follows if we assume it to be true. This results from the fact that the operator functions on interpretations defined on all arguments, thus allowing a self–dependent argument to affect its status. The same is true if we consider bigger positive dependency cycles.

The labeling–based semantics are now as follows:

Definition 2.134. Let $D = (A, C = \{\varphi_a\}_{a\in A})$ be an ADF and v be a three–valued interpretation for D and Γ_D its characteristic operator. v is:

- a **three–valued model** of D iff for all $a \in A$ we have that $v(a) \neq \mathbf{u}$ implies that $v(a) = v(\varphi_a)$.
- an **admissible** labeling of D iff $v \leq_i \Gamma_D(v)$.
- a **complete** labeling of D iff $v = \Gamma_D(v)$.
- a **preferred** labeling of D iff it is $\leq_i$–maximal admissible labeling of D.
- a **grounded** labeling of D iff it is the least fixpoint of Γ_D.

We will shorten the semantics to $mod3$, $lab{-}adm$, $lab{-}comp$, $lab{-}pref$ and $lab{-}grd$ for functional representation.

The stable semantics is a slightly different case. Although formally we receive a set, not an interpretation, this makes no difference for stability. As nothing is left undecided, there is a one–to–one correspondence between the extensions and labelings. The current state of the art definition, presented in [21, 84] is based on the concepts of reduct and grounded semantics:

Definition 2.135. Let $D = (A, L, C)$ be an ADF and $E \subseteq A$ a set of arguments. A **reduct** of D w.r.t. E is a framework $D^E = (E, L^E, C^E)$, where $L^E = L \cap (E \times E)$ and for $e \in E$ we set $C^E_e = \varphi_e[b/\mathbf{f} : b \notin E]$.

Definition 2.136. Let $D = (A, L, C = \{\varphi_a\}_{a\in A})$ be an ADF, $M \subseteq A$ be a model of D and $D^M = (M, L^M, C^M)$ a reduct of D w.r.t. M. Let gv be the grounded model of D^M. Model M is **stable** iff $M = gv^{\mathbf{t}}$.

Example 26. We will now show the extensions of all of the semantics and their sub–semantics on an example. Let $(\{a, b, c, d\}, \{C_a = \neg b, C_b = \neg a, C_c = b \wedge \neg d, C_d = d\})$ be an ADF, as depicted in Figure 20. Its possible labelings are visible in Table 2. As there are over twenty possible three–valued models, we will not list them.

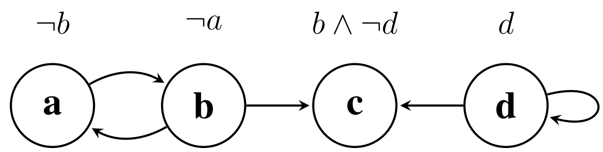


Figure 20: Sample ADF

Table 2: Labelings of the ADF from Figure 20.

ADM	$\{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{u}, d:\mathbf{t}\}, \{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{f}, d:\mathbf{t}\}, \{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{u}, d:\mathbf{f}\},$ $\{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{f}, d:\mathbf{t}\}, \{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{t}, d:\mathbf{f}\}, \{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{u}, d:\mathbf{u}\},$ $\{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{u}, d:\mathbf{t}\}, \{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{t}\}, \{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{f}\},$ $\{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{u}, d:\mathbf{u}\}, \{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{u}, d:\mathbf{t}\}, \{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{u}, d:\mathbf{f}\},$ $\{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{u}\}, \{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{u}, d:\mathbf{f}\}, \{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{u}, d:\mathbf{u}\}$
COMP	$\{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{f}\}, \{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{u}, d:\mathbf{u}\}, \{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{f}, d:\mathbf{t}\},$ $\{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{u}\}, \{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{u}, d:\mathbf{f}\}, \{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{t}, d:\mathbf{f}\},$ $\{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{t}\}, \{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{f}, d:\mathbf{t}\}, \{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{u}, d:\mathbf{u}\}$
PREF	$\{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{f}, d:\mathbf{t}\}, \{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{t}, d:\mathbf{f}\}, \{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{f}\},$ $\{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{t}\}$
STB	$\{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{f}\}, \{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{t}, d:\mathbf{f}\}$
GRD	$\{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{u}, d:\mathbf{u}\}$

2.3.5 Extension–Based Semantics of ADFs

In [74] we have developed a family of extension–based semantics and created a classification of them w.r.t. positive dependency cycles. We have distinguished four categories and used an $xy-$ prefixing system to denote them. The x stated whether only acyclic - a - arguments can be accepted in an extensions, or would cyclic - c - also do the trick. y then meant if we need to "defend" only from acyclic - again, a - arguments, or of this restriction is not necessary - c. In [75], we have further split the ca– type into two groups and referred to them as ca_1 and ca_2. While the first approach assumed that we do not need to defend from any cyclic argument, the latter made an exception for the arguments that are based on a cycle that we have accepted in an extension. We will now recall all of the semantics and refer the reader to the original work for proofs and further explanations.

In the Dung's setting, conflict–freeness meant that the elements of an extension could not attack one another. This is also the common interpretation in various other AF generalizations, including the bipolar ones such as AFNs and EASs [69, 72]. Providing an argument with the required support is then a separate condition. In ADFs, where we lose the set representation of relations in favor of abstraction, not including "attackers" and accepting "supporters" is combined into one notion. It basically takes the intuition of "arguments that can stand together" [12] to a higher level, which simply leads to satisfying the acceptance conditions. The pd–acyclic version of conflict–freeness needs to take into account also the attacks on the evaluation level (see Example 22). This brings us to the following definitions, previously briefly introduced in Section 2.3.3:

Definition 2.123. Let $D = (A, C)$ be an ADF. A set of arguments $E \subseteq A$ is a **conflict–free extension** of D if for all $s \in E$ we have $C_s(E \cap par(s)) = in$. E is a **pd–acyclic conflict–free extension** of D iff for every argument $a \in E$, there exists an unblocked acyclic pd–evaluation on E w.r.t. v^E.

The two semantics will be abbreviated with cf and $acy\text{-}cf$ respectively. Please note that conflict–free (pd–acyclic conflict–free) extensions can be also viewed as standard (acyclic) pd–evaluations that are not self–blocking. Every set for which an acceptance condition is in can be made into a trivial decisively in evaluation by assigning $\mathbf{f}$ to absent arguments. From it, a minimal interpretation can be extracted, and we can observe that its $\mathbf{t}$ part will be contained in the extension in question, and the $\mathbf{f}$ outside it. We can thus gather such interpretations for the arguments in an extension and obtain an unblocked standard evaluation. If we are dealing with a pd–acyclic conflict–free extensions, then as a result of Theorem 2.146, we can recombine the unblocked acyclic pd–evaluations for arguments in the set and obtain (at least) one that has a pd–sequence containing all and only arguments in the extension. Similarly, if a standard evaluation (F, B) is not self–blocking, then clearly $C_a(F \cap par(a)) = in$ for an $a \in F$ and F is conflict–free. If (F, B) is an acyclic evaluation, then we can clearly "trim" it down for a given a in the pd–sequence and obtain an unblocked acyclic pd–evaluation for every argument in F. Thus, we obtain our pd–acyclic conflict–free sets.

The concept of a model (short form mod) basically follows the intuition that if something can be accepted, it should be accepted. It was meant as a basis for the stable semantics, as could have already been observed in Section 2.3.4. However, we would like to note that there is more than one way to produce stable extensions and we do not need to make use of reducts:

Definition 2.137. Let $D = (A, C)$ be an ADF. A conflict–free extension $E \subseteq A$ of D is a **model** of D if $\forall\ s \in A,\ \ C_s(E \cap par(s)) = in$ implies $s \in E$.

Theorem 2.138. *Let $D = (A, C)$ be an ADF. A model $E \subseteq A$ of D is a* ***stable extension*** *of D iff it is pd–acyclic conflict–free in D.*

Lemma 2.139. *A set $E \subseteq A$ is stable in D iff it is a pd–acyclic conflict–free extension of D s.t. $E^{a+} = A \setminus E$.*

Let us now continue with the grounded and acyclic grounded semantics. Just like in the Dung's setting, they preserve the unique–status property. Moreover, the first one is defined in the terms of a special operator:

Definition 2.140. Let $\Gamma'_D(A, R) = (acc(A, R), reb(A, R))$, where $acc(A, R) = \{r \in S \mid A \subseteq S' \subseteq (S \backslash R) \Rightarrow C_r(S' \cap par(s)) = in\}$ and $reb(A, R) = \{r \in S \mid A \subseteq S' \subseteq (S \backslash R) \Rightarrow C_r(S' \cap par(s)) = out\}$. Then E is the **grounded model** of D iff for some $E' \subseteq S$, (E, E') is the least fix–point of Γ'_D.

Although it might look complicated at first, acc and reb are nothing more than means of retrieving decisively in/out arguments via a set representation (see [75] for more details). Therefore, there exists an alternative way to compute the grounded extension, in line with Proposition 2.7:

Proposition 2.141. *Let $D = (A, C)$ be an ADF and v an empty interpretation. For every argument $a \in A$ that is decisively in w.r.t. v, set $v(a) = \mathbf{t}$ and for every argument $b \in A$ that is decisively out w.r.t. v, set $v(b) = \mathbf{f}$. Repeat the procedure until no further assignments can be done. The* ***grounded extension*** *of D is then $v^{\mathbf{t}}$.*

The acyclic version is very similar; however, instead of working with the standard range construction, it uses the acyclic version.

Definition 2.142. Let $D = (A, C)$ be an ADF and v an empty interpretation. For every argument $a \in A$ that is decisively in w.r.t. v, set $v(a) = \mathbf{t}$. For every argument $b \in A$ s.t. all of its acyclic pd–evaluations are blocked by v, set $v(b) = \mathbf{f}$. Repeat the procedure until no further assignments can be done. The **acyclic grounded extension** of D is then $v^{\mathbf{t}}$.

Just like in the conflict–free case, we will shorten the semantics to grd and $acy\text{-}grd$. The rest of the extension–based semantics will be abbreviated in the usual manner with an appropriate xy– prefix.

Let us now focus on admissible, preferred and complete semantics. What is important to understand is the fact that even though there are significant differences between the aa, ac, cc and ca families, the core concept remains the same – admissibility representing a defensible stand, preferred extensions being maximally admissible, and complete accepting whatever they defend. By replacing defense with decisiveness w.r.t. range, we basically obtain the ADF semantics. The differences lie in which range should be chosen, and if acyclicity of the extension is also desired.

Definition 2.143. Let $D = (A, C)$ be an ADF and $E \subseteq A$ a set of arguments. E is a:

- a **cc–admissible** extension of D iff it is conflict–free in D and every $e \in E$ is decisively in w.r.t. standard range v_E.
- an **aa–admissible** extension of D iff it is pd–acyclic conflict–free in D and every $e \in E$ is has an acyclic pd–evaluation $((a_0, .., a_n, B)$ on E s.t. all members of B are mapped to $\mathbf{f}$ by the acyclic range v_E^a.
- a **ac–admissible** extension of D iff it is pd–acyclic conflict–free in D and every $e \in E$ has an acyclic pd–evaluation $((a_0, .., a_n, B)$ on E s.t. all members of B are mapped to $\mathbf{f}$ by the standard range v_E.
- a **ca$_1$–admissible** extension of D iff it is conflict–free in D and every $e \in E$ is decisively in w.r.t. acyclic range v_E^a.

- a **ca$_2$–admissible** extension of D iff it is conflict–free ind D and every $e \in E$ is decisively in w.r.t. partially acyclic range v_E^p.

Please note that even though the usage of pd–acyclicity in the case of aa and ac families was to be expected, when it comes to semantics acyclic on the inside we have to deal with the "second" level of conflict visible in the blocking sets of acyclic evaluations. This gives rise to another level of "defense", where not only we check if arguments are decisively in w.r.t. range, but also need to protect their evaluations.

Example 27. Recall the framework $(\{a, b, c, d, e\}, \{C_a = \top, C_b = \neg a \vee c, C_c = b, C_d = \neg c \wedge \neg e, C_e = \neg d\})$ from Example 19. The acyclic evaluation for b was $((b), \{a\})$ and $((b, c), \{a\})$ for c. Consider the set $E = \{b, c\}$. Its discarded set is just $\{d\}$, independently of the type. Technically speaking, E is pd–acyclic conflict–free. Moreover, both arguments are decisively in w.r.t. (any) range interpretation. However, again we can observe that accepting a will force a cycle between b and c, even though the conditions of arguments will remain satisfied. Consequently, acyclicity requires a separate level of "defense".

Remark. It is worth mentioning that if an argument possesses an evaluation (of any type) s.t. the pd–sequence is in the set and the blocking set is in the discarded set of appropriate type, it is decisively in w.r.t. the respective range – this is due to the fact that range simply becomes a completion of the decisively in interpretation used in the construction of the evaluation. Thus, explicitly requiring decisiveness in case of aa and ac–admissible semantics is redundant – even though decisiveness does not imply a protected acyclic evaluation, a protected evaluation does lead to decisiveness.

A stronger relation can be observed between standard evaluations and decisiveness. Every argument in a conflict–free extension will have a standard evaluation on this set s.t. the blocking set is disjoint from the extension. The extension itself can be seen as a single standard evaluation in which the pd–set is disjoint from the blocking set. A ca or cc–admissible extension can also be seen as a standard evaluation for which the blocking set is falsified by the range. While decisiveness of an argument did not imply acyclicity due to the fact that not every decisively in interpretation for an argument will be used to create a acyclic pd–evaluation, we are less restricted in the standard case. Any decisively in interpretation s.t. its true part is contained in a conflict–free extension is good enough for a standard evaluation.

Definition 2.144. Let $D = (A, C)$ be an ADF. A set $E \subseteq A$ is an **xy–preferred** extensions of D, where $x, y \in \{a, c\}$, iff it is maximal w.r.t. set inclusion xy–admissible extension of D.

Definition 2.145. Let $D = (A, C)$ be an ADF and $E \subseteq A$ a set of arguments. E is a:

- a **cc–complete** extension of D iff it is cc–admissible in D and every $a \in A$ that is decisively in w.r.t. v_E, is in E.

- a **ac–complete** extension of D iff it is ac–admissible in D and every $a \in A$ that is decisively in w.r.t. v_E, is in E.

- an **aa–complete** extension of D iff it is aa–admissible in D and every $a \in A$ that is decisively in w.r.t. v_E^a, is in E.

- a **ca$_1$–complete** extension of D iff it is ca$_1$–admissible in D and every $a \in A \setminus E^{a+}$ that is decisively in w.r.t. v_E^a, is in E.

- a **ca$_2$–complete** extension of D iff it is ca$_2$–admissible in D and every $a \in A$ that is decisively in w.r.t. v_E^p, is in E.

The final theorem we want to recall in this section is valuable not for the relation between the pd–acyclic conflict–free and grounded extensions, but for it consequences and the way it is proved (see [75]). The meaning behind it is that while every argument in a pd–acyclic conflict–free extension has an unblocked evaluation and can be assigned different decisively in interpretations by the related pd–functions, we can create a single "big" acyclic pd–evaluation for the whole extension:

Theorem 2.146. *Let $D = (A, C)$ be an ADF and $E \subseteq A$ a set of arguments. E is pd–acyclic conflict–free iff it is the grounded extension of the reduct $D^E = (E, C^E)$ of D w.r.t. E.*

Example 28. Let us look at the ADF $(\{a, b, c, d, e\}, \{C_a = e, C_b = d \lor (c \land e), C_c = \neg e, C_d = \top, C_e = a \land b\})$ depicted in Figure 21. $\emptyset$, $\{c\}$, $\{d\}$, $\{b, d\}$, $\{c, d\}$, $\{b, c, d\}$ and $\{a, b, d, e\}$ are its conflict–free extensions, with the acyclic ones being $\emptyset$, $\{c\}$, $\{d\}$, $\{b, d\}$, $\{c, d\}$ and $\{b, c, d\}$.

From the available conflict–free extensions only two are models. $\emptyset$ and $\{c\}$ do not qualify as they do not include d, which has an acceptance condition that is always satisfied. Presence of d evaluates the condition of b to in, and thus the $\{d\}$ and $\{c, d\}$ conflict–free extensions are also not models. Also the condition of c is satisfied under $\{b, d\}$ and we need to exclude this set as well. We are thus left with $\{b, c, d\}$ and $\{a, b, d, e\}$ and as no arguments outside the sets have satisfied acceptance conditions w.r.t. them, we obtain our two models. The first extension is also pd–acyclic conflict–free and as a result, the single stable model of our framework.

The easy ac– and cc–admissible extensions are $\emptyset$, $\{d\}$ and $\{b, d\}$. Since d is an initial argument, it can be accepted without any restrictions. The presence of d makes b acceptable independently of what happens to c and e, thus we do not have to analyze the conflict between them in this context. The last cc–admissible extension is $\{a, b, d, e\}$ and again, since d is present, the conflict can be disregarded. This is also the only cc–admissible extension that is not ac–admissible.

Let us now move to semantics acyclic on the "outside", starting with the aa approach. The ac– and cc–admissible extensions $\emptyset$, $\{d\}$ and $\{b, d\}$ are also aa–admissible. However, we can observe a cyclic positive dependency between a and b and $\{a, b, d, e\}$ cannot be aa–admissible. Since we only have to defend against acyclic attackers, $\{c\}$, $\{b, c, d\}$ and $\{c, d\}$ are additional aa–extensions. Finally, all of those sets, including $\{a, b, d, e\}$, are ca_1 and ca_2–admissible.

The extension $\{b, d\}$ will be cc and ac–complete, but not aa, ca_1 and ca_2–complete as a and e will be automatically in the acyclic range. On the other hand, $\{b, c, d\}$ will be aa, ca_1 and ca_2–complete, but not cc and ac–complete. Finally, $\{a, b, d, e\}$ will be ca_1, ca_2 and cc–complete.

The set $\{a, b, d, e\}$ is our only cc–preferred extension, $\{b, d\}$ is ac–preferred and $\{b, c, d\}$ is aa–preferred. Finally, the ca_1 and ca_2–preferred extensions are $\{b, c, d\}$ and $\{a, b, d, e\}$.

For the grounded cases, assume an empty interpretation v. It is easy to see that only d is decisively in w.r.t. v and that there are no decisively out arguments. However, now that we have the $d : \mathbf{t}$ assignment, b can be also decisively assumed. Again, no decisive outing occurs, and next round returns us no new assignments. Thus, the grounded extension is $\{b, d\}$. When it comes to acyclic case, we can again trivially accept d. However, since a and e have no acyclic pd–evaluations, they are mapped to $\mathbf{f}$. By accepting d we can assume b, and from the rejection of e follows c. Consequently, our acyclic grounded extension will be $\{b, d, c\}$ and contains the standard one $\{b, d\}$.

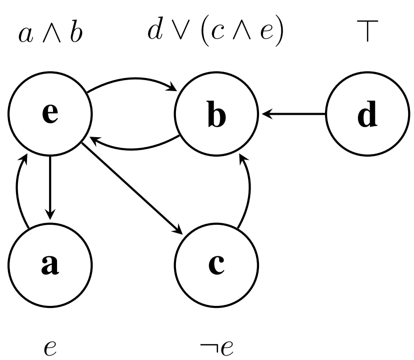


Figure 21: Sample ADF

2.3.6 Properties of ADF Semantics

In this section we will recall the properties of the ADF semantics [21, 74, 75], which will prove useful in proving a number of translations.

Theorem 2.147. *Let $D = (A, C)$ be an ADF. The following holds:*

- *each preferred labeling is a complete labeling, but not vice versa.*
- *the grounded model is the $\leq_i$–least complete labeling.*

- *the complete labelings of D form a complete meet–semilattice w.r.t. $\leq_i$.*

Theorem 2.148. *Let $D = (A, C)$ be an ADF, v be a three–valued interpretation on A and v' its (maximal) two–valued sub–interpretation. v is admissible iff all arguments mapped to $\mathbf{t}$ are decisively in w.r.t. v' and all arguments mapped to $\mathbf{f}$ are decisively out w.r.t. v'.*

Theorem 2.149. *Let $D = (A, C)$ be an ADF, v an admissible labeling and v' its (maximal) two–valued sub–interpretation. v is complete iff all arguments decisively out w.r.t. v' are mapped to $\mathbf{f}$ by v and all arguments decisively in w.r.t. v' are mapped to $\mathbf{t}$ by v.*

Proposition 2.150. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a standard and $S \subseteq A$ a pd–acyclic conflict–free extension of D, with v_E, v_E^p, v_E^a, v_S, v_S^p and v_S^a as their corresponding standard, partially acyclic and acyclic range interpretations. Let $s \in A$ be an argument. The following holds:*

- *if $v_E(s) = \mathbf{f}$, then s is decisively out w.r.t. v_E. The same holds or v_E^p, but not for v_E^a.*
- *if $v_S(s) = \mathbf{f}$, then s is decisively out w.r.t. v_S. The same holds for v_E^p and v_E^a.*
- *if $v_E(s) = \mathbf{f}$, then $C_s(E \cap par(s)) = out$. The same holds or v_E^p, but not for v_E^a.*
- *if $v_S(s) = \mathbf{f}$, then $C_s(S \cap par(s)) = out$. The same holds for v_E^p and v_E^a.*

Lemma 2.151. *Let $D = (A, C)$ be an ADF and E and E' two conflict–free extensions s.t. $E \subseteq E'$. It follows that $v_{E'}$ is a completion of v_E to some set $A' \subseteq A$.*

Let E and E' be two pd–acyclic conflict–free extensions s.t. $E \subseteq E'$. It follows that $v_{E'}^a$ is a completion of v_E^a to some set $A' \subseteq A$ and that $v_{E'}^p$ is a completion of v_E^p to some set $A'' \subseteq A$.

Lemma 2.152. *Let $D = (A, C)$ be an ADF. The following holds:*

- *every ac–admissible extension of D is cc–admissible in D.*
- *every ac–admissible extension of D is aa–admissible in D.*
- *every aa–admissible extension of D is ca$_2$–admissible in D.*
- *every cc–admissible extension of D is ca$_2$–admissible in D.*
- *every ca$_2$–admissible extension of D is ca$_1$–admissible in D.*
- *not every ca$_1$–admissible extension of D is ca$_2$–admissible in D.*

Lemma 2.153. *Let $D = (A, C)$ be an ADF. Let xy and x'y' be two admissible sub–semantics, where $x, x', y, y' \in \{a, c\}$, s.t. every xy–admissible extension is also x'y'–admissible (see Lemma 2.152). Then every xy–preferred extension of D is contained in some x'y'–preferred extension of D.*

Lemma 2.154. ***CC/AC/AA Fundamental Lemma**: Let $D = (A, C)$ be an ADF, E a cc(ac)–admissible extension of D, v_E its range interpretation and $a, b \in A$ two arguments decisively in w.r.t. v_E. Then $E' = E \cup \{a\}$ is cc(ac)–admissible in D and b is decisively in w.r.t. v'_E.*

Let E be an aa–admissible extension of D, v^a_E its acyclic range interpretation and $a, b \in A$ two arguments decisively in w.r.t. v^a_E. Then $E' = E \cup \{a\}$ is aa–admissible in D and b is decisively in w.r.t. v'^a_E.

Lemma 2.155. ***Weak CA$_1$ Fundamental Lemma**: Let $D = (A, C)$ be an ADF, $E \subseteq A$ a ca$_1$–admissible extension, v^a_E its acyclic range interpretation and $a, b \in A \setminus E^{a+}$ arguments decisively in w.r.t. v_E. Then $E' = E \cup \{a\}$ is ca$_1$–admissible in D, b is decisively in w.r.t. $v_{E'}$, but it is not necessarily in $A \setminus E'^{a+}$.*

Lemma 2.156. ***CA$_2$ Fundamental Lemma** Let $D = (A, C)$ be an ADF, $E \subseteq A$ an ca$_2$–admissible extension of D, v^p_E its partially acyclic range interpretation and $a, b \in A$ two arguments decisively in w.r.t. v^p_E. Then $E' = E \cup \{a\}$ is ca$_2$–admissible in D and b is decisively in w.r.t. v'^p_E.*

Lemma 2.157. *Let $D = (A, C)$ be an ADF. It holds that:*

- *every ac–complete extension of D is cc–complete in D.*
- *every aa–complete extension of D is ca$_1$–complete in D.*
- *every aa–complete extension of D is ca$_2$–complete in D.*
- *not every ca$_1$–complete extension of D is ca$_2$–complete in D and vice versa.*

We can now continue with an ADF version of Theorem 2.10 from the Dung's setting:

Theorem 2.158. *Let $D = (A, C)$ be an ADF. The following holds:*

- *every xy–preferred extension of D is an xy–complete extension of D for $x, y \in \{a, c\}$, but not vice versa.*
- *the grounded extension of D might not be an aa–, ca$_1$– or ca$_2$–complete extension of D.*
- *the grounded extension of D is the least w.r.t. set inclusion ac(cc)–complete extension of D.*
- *the acyclic grounded extension of D is the least w.r.t. set inclusion aa–complete extension of D and a minimal ca$_1$(ca$_2$)–complete extension of D.*
- *the cc–, ac– and aa–complete extensions of D form complete meet–semilattices w.r.t. set inclusion.*

- *the ca_1 and ca_2–complete extensions of D may not form complete meet–semilattices w.r.t. set inclusion.*

Lemma 2.159. *Let $D = (A, C)$ be an ADF and $E \subseteq A$ a model of D. Then $E^{a+} = A \setminus E$ and $E^{p+} = A \setminus E$.*

Lemma 2.160. *Let $D = (A, C)$ be an ADF. Every model of D is ca_1/ca_2–complete in D, but not necessarily ca_1/ca_2–preferred in D.*

Lemma 2.161. *Let $D = (A, C)$ be an ADF. Every stable extension of D is an aa–preferred in D, but not vice versa. It is not necessarily a cc/ac/ca_1/ca_2–preferred extension of D.*

Lemma 2.162. *Let $D = (A, C)$ be an ADF. Every xy–preferred extension of D is a maximal w.r.t. $\subseteq$ xy–complete extension of D for $x, y \in \{a, c\}$.*

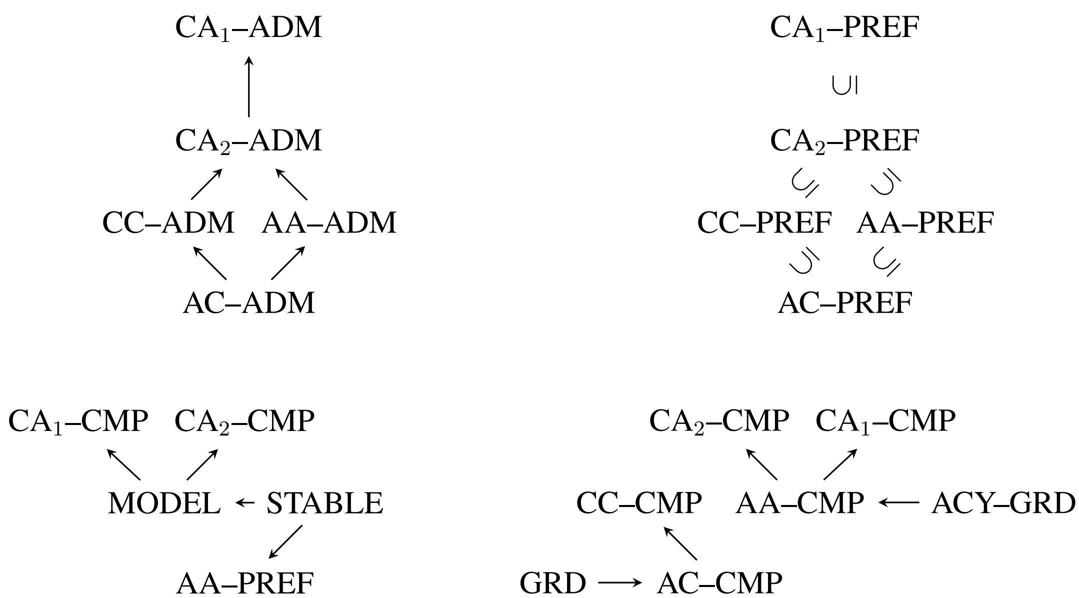


Figure 22: The relations between given extension–based sub–semantics. $x \to y$ should be read as extensions of x are extensions of y. $x \subseteq y$ should be read as any extension of x is contained in some extension of y.

2.3.7 Comparison of Extension–Based and Labeling–Based Semantics

In this section we will briefly recall the relation between the families of ADF semantics and refer the reader for further details and proofs to [75]. In order to compare extensions and labelings we will use the notion of correspondence:

Definition 2.163. Let $D = (A, C)$ be an ADF, v a three–valued interpretation over A and $E \subseteq A$ a set of arguments. v and E **correspond** iff $v^{\mathbf{t}} = E$.

By the abuse of notation we will also use the notion of a $\mathbf{u}$–completion, which should be understood as a three–valued interpretation that assigns $\mathbf{u}$ to the "missing" mappings of a given two–valued interpretation.

Theorem 2.164. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a conflict–free and $S \subseteq A$ a pd–acyclic conflict–free extension of D. The $\mathbf{u}$–completions of v_E, v_E^p, v_S, v_S^p and v_S^a to A are three–valued models of D. The $\mathbf{u}$–completion of v_E^a might not be a three–valued model of D.*

Theorem 2.165. *Let $D = (A, C)$ be an ADF and v be a three–valued model of D. Then $v^{\mathbf{t}}$ is a conflict–free set of D.*

Theorem 2.166. *Let $D = (A, C)$ be an ADF. The following holds:*

- *let E be a cc–admissible extension of D. Then the $\mathbf{u}$–completion of v_E is an admissible labeling of D.*
- *let E be an ac–admissible extension of D. Then the $\mathbf{u}$–completion of v_E is an admissible labeling of D.*
- *let E be an aa–admissible extension of D. Then the $\mathbf{u}$–completion of v_E^a is an admissible labeling of D.*
- *let E be an ca$_2$–admissible extension of D. Then the $\mathbf{u}$–completion of v_E^p is an admissible labeling of D.*
- *let v be an admissible labeling of D. Then $v^{\mathbf{t}}$ is a ca$_1$ and ca$_2$–admissible extension of D.*
- *let E be ca$_1$–admissible extension of D. There might be no admissible labeling of D corresponding to E.*

Remark. Please note that although every admissible labeling has a corresponding ca$_2$–admissible extension and vice versa, just like in the Dung's framework it does not need to be a one–to–one relation. The $\mathbf{u}$–completion of a given range produces only one of many admissible labelings that have common $\mathbf{t}$ mappings.

Theorem 2.167. *Let $D = (A, C)$ be an ADF. The following holds:*

- *let $E \subseteq A$ be an aa–, ac–, cc– or ca$_1$–preferred extension of D. There might not exist a corresponding preferred labeling of D.*
- *let $E \subseteq A$ be a ca$_2$–preferred extension of D. The $\mathbf{u}$–completion of v_E^p to A is a preferred labeling of D.*

- *let v be a preferred labeling of D. Then $v^{\mathbf{t}}$ is a ca$_2$–complete extension of D, but it does not have to be aa–, ac–, cc– or ca$_1$–complete or aa–, ac–, cc–, ca$_1$– or ca$_2$–preferred in D.*

Theorem 2.168. *Let $D = (A, C)$ be an ADF. The following holds:*

- *let E be a cc–complete extension of D. The $\mathbf{u}$–completion of v_E is a complete labeling of D.*
- *let E be an ac–complete extension of D. The $\mathbf{u}$–completion of v_E is a complete labeling of D.*
- *let E be an aa–complete extension of D. The $\mathbf{u}$–completion of v_E^a is a complete labeling of D.*
- *let E be a ca$_2$–complete extension of D. The $\mathbf{u}$–completion of v_E^p is a complete labeling of D.*
- *let v be a complete labeling of D. The set v^t might not be a cc–, ac–, aa–, ca$_1$– or ca$_2$–complete extension of D.*
- *let v be a complete labeling of D. There exists a ca$_2$–complete extension E of D s.t. $v^{\mathbf{t}} \subseteq E$.*
- *let E be a ca$_1$–complete extension of D. There might be no corresponding complete labeling of D.*

As the grounded semantics has a very clear meaning, it is no wonder that both available approaches coincide, as already noted in [21].

Theorem 2.169. *Let $D = (A, C)$ be an ADF. Given the two–valued grounded extension $E \subseteq A$ of D and the grounded labeling v of D, it holds that $v^{\mathbf{t}} = E$.*

However, the best we can find for the acyclic grounded extension is an associated complete labeling. It will of course not be the least one, since that corresponds to the standard grounded semantics.

Theorem 2.170. *Let $D = (A, C)$ be an ADF and E its acyclic grounded extension. The $\mathbf{u}$–completion of the acyclic range of E is a complete labeling of D.*

The results are summarized in Figure 23. Please note we do not include the transitive relations here, such as grounded extension having a corresponding complete labeling due to grounded labeling being complete.

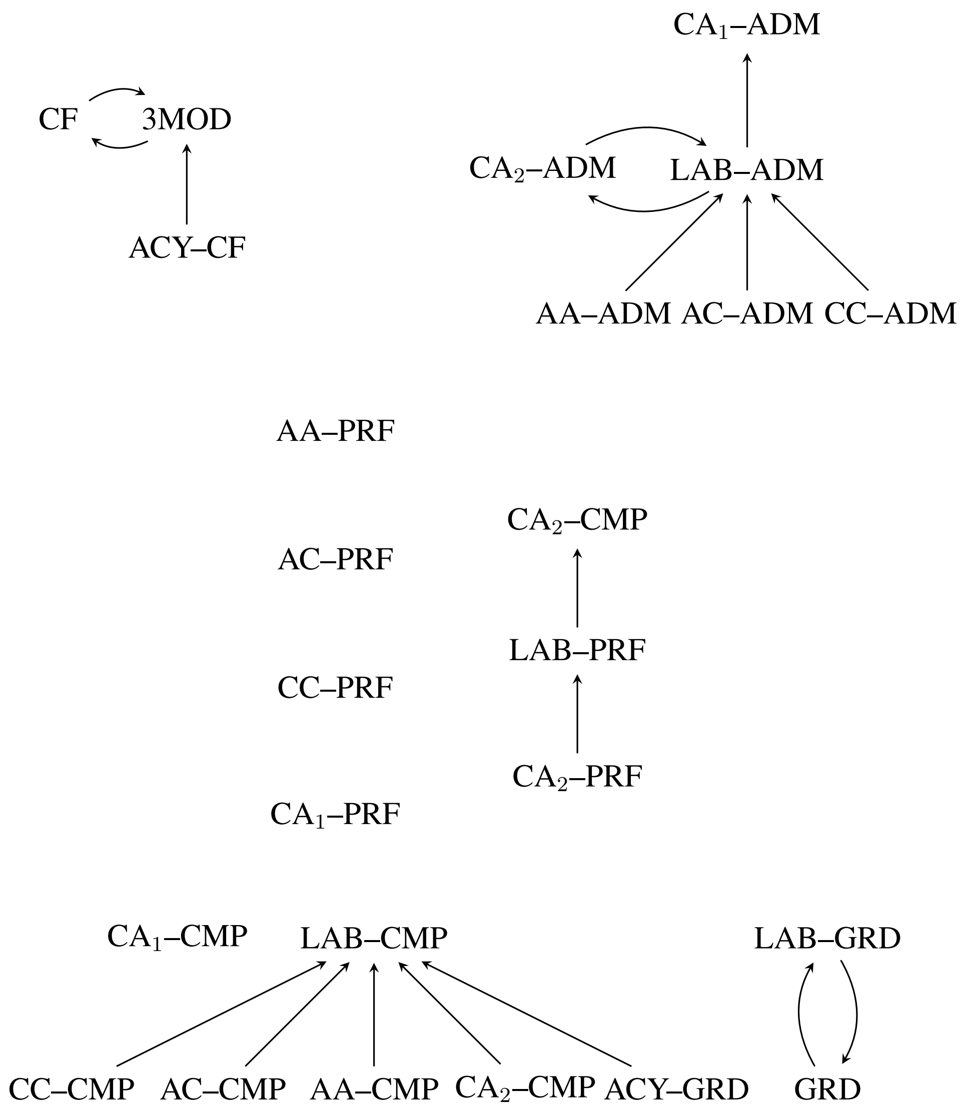


Figure 23: The relations between extension–based and labeling–based semantics. $x \rightarrow y$ should be read as every extension/labeling of type x has a corresponding labeling/extensions of type y.

Example 29. Let us consider a simple framework $(\{a, b\}, \{C_a = a, C_b = b\})$ depicted in Figure 24. Its cc,ca$_1$ and ca$_2$–complete extensions are $\emptyset$, $\{a\}$, $\{b\}$ and $\{a, b\}$, while the aa and ac one is just $\emptyset$. Thus, we obtain one cc, ca$_1$ and ca$_2$–preferred extension $\{a, b\}$ and a single aa and ac–preferred one – $\emptyset$.

The complete labelings for this framework are $\{a : \mathbf{u}, b : \mathbf{u}\}$, $\{a : \mathbf{u}, b : \mathbf{f}\}$, $\{a : \mathbf{f}, b : \mathbf{u}\}$, $\{a : \mathbf{f}, b : \mathbf{f}\}$, $\{a : \mathbf{t}, b : \mathbf{u}\}$, $\{a : \mathbf{t}, b : \mathbf{f}\}$, $\{a : \mathbf{u}, b : \mathbf{t}\}$, $\{a : \mathbf{f}, b : \mathbf{t}\}$ and finally $\{a : \mathbf{t}, b : \mathbf{t}\}$. The first four correspond to $\emptyset$, then both $\{a\}$ and $\{b\}$ have two labelings, and finally we receive $\{a, b\}$. In this case, our results in compliance with the cc, ca$_1$ and ca$_2$–

complete extensions. The preferred labelings are $\{a : \mathbf{f}, b : \mathbf{f}\}$,$\{a : \mathbf{t}, b : \mathbf{f}\}$, $\{a : \mathbf{f}, b : \mathbf{t}\}$ and $\{a : \mathbf{t}, b : \mathbf{t}\}$, again producing the sets $\emptyset$, $\{a\}$, $\{b\}$ and $\{a, b\}$. We can observe that $\{a\}$ and $\{b\}$ are not preferred extensions of any family.

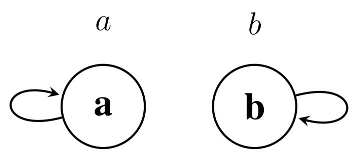


Figure 24: Sample ADF

Example 30. Let us consider a simple framework $(\{a, b, c, d\}, \{C_a = \neg c, C_b = \neg d, C_c = c, C_d = d\})$ depicted in Figure 25. Its extensions and labelings will be listed in Tables 3 and 4. Although there are many admissible labelings, in the end they produce the following sets: $\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, d\}, \{b, c\}$ and $\{c, d\}$. Similar follows for the complete labelings. The preferred ones correspond to $\{a, b\}, \{a, d\}, \{b, c\}$ and $\{c, d\}$.

We can observe that in our example, every admissible labeling will produce a ca$_1$ and ca$_2$–admissible extension and vice versa. However, even though every aa, cc and ac–admissible extension will have a corresponding labeling, it does not hold in the other direction. Although every complete extension of a given type will have a corresponding complete labeling, the sets $\{a\}$ and $\{b\}$ produced by some of the complete labelings are not complete extensions in any of the families. Finally, we can see that the ac–preferred extension $\emptyset$ has no corresponding preferred labeling.

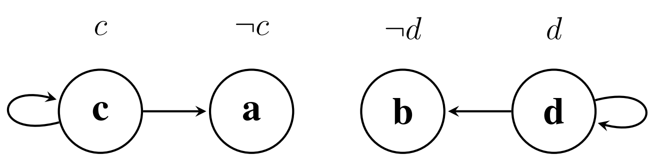


Figure 25: Sample ADF

2.3.8 Sub–Semantics Coincidence: the AADF$^+$ Subclass

With this amount of different families of semantics available in ADFs, it is natural to ask what are the conditions under which all xy–subtypes of a given semantics coincide, e.g. when is every aa–admissible extension is also cc–admissible and so on. In this section we will describe a subclass of ADFs for which our classification system collapses. Moreover, this class will also provide a more precise correspondence between the extension and labeling–based approaches. We will refer to the frameworks in this subclass as the positive dependency acyclic abstract dialectical frameworks and denote them as AADF$^+$s.

Table 3: Extensions of the ADF from Figure 25.

ADM	$CA_{1,2}$	$\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,d\}, \{b,c\}, \{c,d\}$
	CC	$\emptyset, \cancel{\{a\}, \{b\}}, \{c\}, \{d\}, \cancel{\{a,b\}, \{a,d\}, \{b,c\}}, \{c,d\}$
	AA	$\emptyset, \{a\}, \{b\}, \cancel{\{c\}, \{d\}}, \{a,b\}, \cancel{\{a,d\}, \{b,c\}, \{c,d\}}$
	AC	$\emptyset, \cancel{\{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,d\}, \{b,c\}, \{c,d\}}$
COMP	$CA_{1,2}$	$\cancel{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}}, \{a,b\}, \{a,d\}, \{b,c\}, \{c,d\}$
	CC	$\emptyset, \cancel{\{a\}, \{b\}}, \{c\}, \{d\}, \cancel{\{a,b\}, \{a,d\}, \{b,c\}}, \{c,d\}$
	AA	$\cancel{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}}, \{a,b\}, \cancel{\{a,d\}, \{b,c\}, \{c,d\}}$
	AC	$\emptyset, \cancel{\{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,d\}, \{b,c\}, \{c,d\}}$
PREF	$CA_{1,2}$	$\cancel{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}}, \{a,b\}, \{a,d\}, \{b,c\}, \{c,d\}$
	CC	$\cancel{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,d\}, \{b,c\}}, \{c,d\}$
	AA	$\cancel{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}}, \{a,b\}, \cancel{\{a,d\}, \{b,c\}, \{c,d\}}$
	AC	$\emptyset, \cancel{\{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,d\}, \{b,c\}, \{c,d\}}$

Table 4: Labelings of the ADF from Figure 25.

ADM	$\{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{u}, d:\mathbf{u}\}, \{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{f}, d:\mathbf{u}\}, \{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{u}, d:\mathbf{f}\},$ $\{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{f}, d:\mathbf{f}\}, \{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{t}, d:\mathbf{u}\}, \{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{u}, d:\mathbf{t}\},$ $\{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{t}, d:\mathbf{t}\}, \{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{f}, d:\mathbf{t}\}, \{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{t}, d:\mathbf{f}\},$ $\{a:\mathbf{u}, b:\mathbf{f}, c:\mathbf{u}, d:\mathbf{t}\}, \{a:\mathbf{u}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{t}\}, \{a:\mathbf{u}, b:\mathbf{f}, c:\mathbf{t}, d:\mathbf{t}\},$ $\{a:\mathbf{u}, b:\mathbf{t}, c:\mathbf{u}, d:\mathbf{f}\}, \{a:\mathbf{u}, b:\mathbf{t}, c:\mathbf{f}, d:\mathbf{f}\}, \{a:\mathbf{u}, b:\mathbf{t}, c:\mathbf{t}, d:\mathbf{f}\},$ $\{a:\mathbf{f}, b:\mathbf{u}, c:\mathbf{t}, d:\mathbf{u}\}, \{a:\mathbf{f}, b:\mathbf{u}, c:\mathbf{t}, d:\mathbf{f}\}, \{a:\mathbf{f}, b:\mathbf{u}, c:\mathbf{t}, d:\mathbf{t}\},$ $\{a:\mathbf{f}, b:\mathbf{f}, c:\mathbf{t}, d:\mathbf{t}\}, \{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{t}, d:\mathbf{f}\}, \{a:\mathbf{t}, b:\mathbf{u}, c:\mathbf{f}, d:\mathbf{u}\},$ $\{a:\mathbf{t}, b:\mathbf{u}, c:\mathbf{f}, d:\mathbf{f}\}, \{a:\mathbf{t}, b:\mathbf{u}, c:\mathbf{f}, d:\mathbf{t}\}, \{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{t}\},$ $\{a:\mathbf{t}, b:\mathbf{t}, c:\mathbf{f}, d:\mathbf{f}\}$
COMP	$\{a:\mathbf{u}, b:\mathbf{u}, c:\mathbf{u}, d:\mathbf{u}\}, \{a:\mathbf{f}, b:\mathbf{u}, c:\mathbf{t}, d:\mathbf{u}\}, \{a:\mathbf{u}, b:\mathbf{f}, c:\mathbf{u}, d:\mathbf{t}\},$ $\{a:\mathbf{u}, b:\mathbf{t}, c:\mathbf{u}, d:\mathbf{f}\}, \{a:\mathbf{t}, b:\mathbf{u}, c:\mathbf{f}, d:\mathbf{u}\}, \{a:\mathbf{t}, b:\mathbf{t}, c:\mathbf{f}, d:\mathbf{f}\},$ $\{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{t}\}, \{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{t}, d:\mathbf{f}\}, \{a:\mathbf{f}, b:\mathbf{f}, c:\mathbf{t}, d:\mathbf{t}\},$
PREF	$\{a:\mathbf{t}, b:\mathbf{t}, c:\mathbf{f}, d:\mathbf{f}\}, \{a:\mathbf{t}, b:\mathbf{f}, c:\mathbf{f}, d:\mathbf{t}\}, \{a:\mathbf{f}, b:\mathbf{t}, c:\mathbf{t}, d:\mathbf{f}\},$ $\{a:\mathbf{f}, b:\mathbf{f}, c:\mathbf{t}, d:\mathbf{t}\},$

Definition 2.171. Let $D = (A, C)$ be an ADF. D is an **AADF$^+$** iff for every standard evaluation (F, B) of D and the pd–function pd^D it was created with, we can construct an acyclic pd–evaluation $((a_0, ..., a_n), B)$ based on pd^D s.t. $F = \{a_0, ..., a_n\}$.

In other words, a framework is an AADF$^+$ if we can make every standard evaluation in it acyclic. Since every standard evaluation can be represented as a partially acyclic one, we can also say that we are dealing with an AADF$^+$ if every partially acyclic evaluation is in fact acyclic. The following example shows possible frameworks satisfying and not satisfying our definition:

Example 31. Let $D_1 = (\{a, b, c, d, e\}, \{C_a = \top, C_b = \neg a \vee c, C_c = b, C_d = \neg c \wedge \neg e, C_e = \neg d\})$ be the ADF previously analyzed in Example 19 and visible in Figure 16. For every

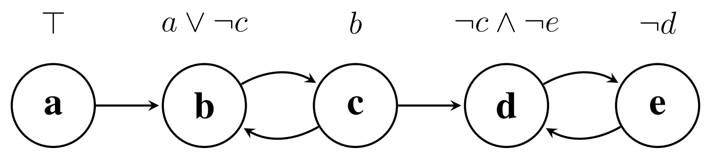


Figure 26: Sample AADF$^+$

argument we can create an acyclic pd–evaluation. However, in some cases we also have a standard evaluation that cannot be made acyclic. We can consider the decisively in interpretations $v_b = \{c : \mathbf{t}\}$ and $v_c = \{b : \mathbf{t}\}$ for b and c respectively and use them to construct a standard evaluation $(\{b, c\}, \emptyset)$. There is no way to order the pd–set w.r.t. v_b and v_c s.t. the pd–sequence requirements would be satisfied. Therefore, D_1 is not an AADF$^+$. Neither are the frameworks depicted in Figures 24 and 25.

We can now consider a modification of D_1 depicted in Figure 26. Our framework is now $D_2 = (\{a, b, c, d, e\}, \{C_a = \top, C_b = a \vee \neg c, C_c = b, C_d = \neg c \wedge \neg e, C_e = \neg d\})$. The minimal decisively in interpretations for our arguments are now as follows: $v_a = \emptyset$, $v_b^1 = \{a : \mathbf{t}\}$, $v_b^2 = \{c : \mathbf{f}\}$, $v_c = \{b : \mathbf{t}\}$, $v_d = \{c : \mathbf{f}, e : \mathbf{f}\}$, $v_e = \{d : \mathbf{f}\}$. We can observe that all of the interpretations with the exception of v_b^1 and v_c satisfy the a_0 requirements of a pd–sequence in an acyclic evaluation. Let us consider a standard evaluation containing c. We can observe it would have to contain b as well; if the associated interpretation is v_b^1, then a needs to be in the pd–set as well. It is easy to see that $\{a, b, c\}$ clearly satisfies the requirements of a pd–sequence. If the interpretation associated with b is v_b^2, then $\{b, c\}$ is a pd–sequence as well. Therefore, given any subset of arguments, if we order it in a way that a precedes b and b precedes c (if they are present), then all of the resulting sequences would meet the pd–sequence restrictions. Thus, the framework is an AADF$^+$.

Theorem 2.172. *Let $D = (A, C)$ be an AADF$^+$. The following holds:*

- *every conflict–free extension of D is pd–acyclic conflict–free in D,*
- *every model of D is stable in D,*
- *given a conflict–free set of arguments $E \subseteq A$, $E^+ = E^{p+} = E^{a+}$,*
- *the aa–, cc–, ac–, ca$_1$ and ca$_2$–admissible extensions of D coincide,*
- *the aa–, cc–, ac–, ca$_1$ and ca$_2$–complete extensions of D coincide,*
- *the aa–, cc–, ac–, ca$_1$ and ca$_2$–preferred extensions of D coincide, and*
- *the grounded and acyclic grounded extensions of D coincide.*

Theorem 2.173. *Let $D = (A, C)$ be an AADF$^+$. The following holds:*

- *every admissible labeling of D has a corresponding aa–, cc–, ac–, ca_1 and ca_2–admissible extension of D and vice versa.*
- *every complete labeling of D has a corresponding aa–, cc–, ac–, ca_1 and ca_2–complete extension of D and vice versa.*
- *every preferred labeling of D has a corresponding aa–, cc–, ac–, ca_1 and ca_2–preferred extension of D and vice versa.*

Finally, it is natural to ask what is the relation between the AADF$^+$ and BADF subclasses. The answer is that while there exist frameworks belonging to both, there are also some belonging to one, but not the other. Let us look at an example.

Example 32. Let $(\{a,b,c\},\{C_a = \top, C_b = \top, C_c = (a\vee b)\wedge(\neg a\vee\neg b)\})$ be a simple ADF. We can observe that c has a condition that is simply an xor on the remaining two arguments. This framework is not a BADF; the links from a and b to c are neither supporting nor attacking. The condition of c is out w.r.t. $\emptyset$, and will turn to in for $\{a\}$ and $\{b\}$. However, it will then turn to out again for $\{a,b\}$. Nevertheless, this simple framework is an AADF$^+$.

Let $(\{a\},\{C_a = a\})$ be another simple framework. There is only one link in the framework – namely, (a,a) – and it is easy to show that it is a supporting one. Thus, our structure is in fact a BADF. However, the only minimal decisively in interpretation for a is $v_a = \{a : \mathbf{f}\}$, and we cannot use it to construct an acyclic pd–evaluation for a. Therefore, we are clearly not dealing with an AADF$^+$.

To show that the subclasses are not disjoint, the easiest way is to take a Dung–style ADF. A structure where only attacks are present, e.g. $(\{a,b,c\},\{C_a = \top, C_b = \neg a, C_c = \neg b\})$, is both a BADF and an AADF$^+$.

2.3.9 Conceptual Differences Between ADFs and Different Argumentation Frameworks

Among all of the argumentation frameworks that we have recalled in this work, the abstract dialectical frameworks clearly stand out. Their definition is visibly different from all the other structures. In this section we will compare acceptance conditions and the relations between the arguments defined explicitly in the framework structure. We will also discuss the problems arising from the differences between them, as they will have an impact on translating other argumentation frameworks into ADFs.

However, before we continue, we will first discuss the falsum arguments in ADFs. One of the things special about ADFs is their ability contain the “non existent” or “false” arguments. In a Dung’s framework, an unattacked argument represents the knowledge we are certain of and as such, will be included in a grounded extension. In an ADF, this argument will be assigned a condition equivalent to $\top$. However, we also have the arguments with $\bot$ condition, which stand for statements we know not to be true and that neither they nor anything building upon them should ever be accepted. We refer to them as *falsum* arguments. Please observe that although self–attacking arguments do not appear in extensions as well, they do not represent knowledge we know to be false – a self–attacker is always assigned an undecided status and as such needs to be defended from. Consequently, its handling is quite different. Falsum arguments in behavior resemble invalid arguments in AFNs and EAFs with the difference that not being valid (i.e. not having a powerful or evidential sequence) is a semantical property, while having a condition that can never be satisfied is more of a structural one. In what follows we will show that the differences between ADFs and other bipolar frameworks can cause undesirable falsum arguments to be created during a translation if certain consistency prerequisites are not met (see also Section 4.4).

The more direct descendants of the Dung's framework, including BAFs, EASs and AFNs, explicitly state "this is a supporter", "this is an attacker" and so on. Thus, in order to know if an argument can be accepted along with other arguments, i.e. whether it is not attacked by them or receives sufficient support, we need to go through all the relations it is a target of. ADFs work somehow the other way around. Acceptance conditions "zoom out" from singular relations and given a set of arguments, they tell us whether the argument can be accepted or not. A condition speaks in terms of requirements and tells us what arguments need to be present or cannot be assumed for it to be satisfied, and whether it is *always* the case, happens only *sometimes*, or if it needs to be like this because *certain different arguments were assumed*. Thus, the focus is put on what would usually be seen as a target of a relation, while other frameworks put the source in center. In ADFs, in order to say if a parent of argument is a supporter, attacker or none of these, we analyze all the models and countermodels of the formula (or the mappings, if the condition is in a functional form), not its structure, i.e. this argument appears as a positive literal, this as a negative and so forth. This is especially visible in the definitions of an ADF–attacker and ADF–supporter (see Definition 2.114). This is also one of the reasons why finding support cycles in ADFs is more difficult than in other support frameworks. Finally, since the role of parent is derived from how it affects the behavior of an argument, not whether it is in e.g. the support relation N, E, or however it is designated in other structures, an attacker or a supporter in a different framework may not necessarily have the same role in a corresponding ADF. The following examples will make this issue more visible.

Example 33. Let us consider an AFN $FN_1 = (\{a,b\}, \{(b,a)\}, \{(\{b\}, a)\})$ depicted in Figure 27a, where argument a is at the same time supported and attacked by b. In a certain sense, a is somewhat difficult to describe – although it depends on b, this dependency cannot be classified as positive or negative, and thus the status of a is more or less undecided. Of course, a can never appear in any extension. However, in most of the bipolar frameworks it is still treated as a valid attacker that one needs to defend from. In the ADF setting, there is no set of arguments that would in the acceptance condition of a – whether we include or exclude b, we are always either attacked or missing support. It can also be seen as a $b \wedge \neg b$ formula. Basically speaking, we receive a falsum argument, which is interpreted as *I do not exist*. This nonexistence is visible in the extensions of the framework, i.e. whether the argument is present or not, it does not affect the extensions. This also means that we do not have to defend from such arguments and thus, there is an important difference between the design of ADFs and other argumentation frameworks.

At this point one might want to say "let us assume this situation does not occur and we do not produce falsum arguments". However, simply adding a restriction that there are e.g. no two arguments a and b s.t. bRa and $\{b\}Na$ in an AFN is not sufficient to bypass this problem and obtain a desirable translation to ADFs. The problem can occur when an attacking argument is just a part of the supporting set, not the set itself.

Example 34. Let us analyze an AFN $FN_2 = (\{a, b, c, d\}, \{(a, d), (b, a), (d, c)\}, \{(\{b, c\}, a)\})$ depicted in Figure 27b. We have that a is again attacked by b, and either b or c needs to be present in order to accept a. If we were to defend from a, we would need to either accept its attacker or cut off its support. In the AFN setting, we would need to either accept b in the set, or attack both b and c, since the two coherent sets that we need to attack are $\{a, b\}$ and $\{a, c\}$. In ADF, only the set $\{c\}$ (or, equivalently, interpretation $\{b : \mathbf{f}, c : \mathbf{t}\}$) can satisfy the acceptance condition of a, thus assuming b or discarding c is sufficient. This can be seen as considering b primarily as an attacker. As a result, $\{d\}$ is ADF, but not AFN admissible.

(a) FN_1 (b) FN_2

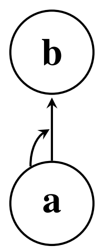
Figure 27: Sample AFNs

A similar issue appears in extended argumentation frameworks, though normally they do not qualify as bipolar. This is due to the fact that defense attack can be translated into a particular form of support in ADFs (see Section 8.6.1), to which we normally refer to as overpowering. Consequently, the problem of consistency needs to be handled here as well. In this case, if an argument carries out an attack on another argument and at the same time defense attacks this conflict, we can obtain an initial argument, i.e. one with acceptance condition being equivalent to verum. Let us look at an example:

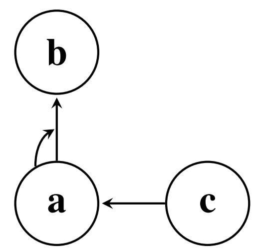
Example 35. Let $EF_1 = (\{a, b\}, \{(a, b)\}, \{(a, (a, b))\})$ be the EAF depicted in Figure 28a. Its admissible extensions are $\emptyset, \{a\}$ and $\{a, b\}$. The combination of attack and defense attack can be read as "b cannot be accepted if a is accepted, unless a is accepted" which propositionally, gives us a tautology. This is not a desirable reading – not only it removes the relations, but also produces different extensions. The (arbitrary) admissible extensions of the corresponding ADF $(\{a, b\}, \{C_a = \top, C_b = \top\})$ would be $\emptyset, \{a\}, \{b\}$

and $\{a, b\}$. Since in our EAF, b requires a in an extension, we can consider modifying our ADF by changing the acceptance condition of b from $\top$ to a. It would now produce the desirable extensions, namely $\emptyset$, $\{a\}$ and $\{a, b\}$. Unfortunately, this is not a long–term method.

Let $EF_2 = (\{a, b, c\}, \{(a, b), (c, a)\}, \{(a, (a, b))\})$ be a modification of our EAF, visible in Figure 28b; we basically include a new argument attacking a. Its admissible extensions are $\emptyset$, $\{c\}$ and $\{c, b\}$. However, if we include the attack into our ADF, the admissible extension of $(\{a, b, c\}, \{C_a = \neg a, C_b = a, C_c = \top\})$ are $\emptyset$ and $\{c\}$. Since a is set to support b, an attack on it renders b unacceptable.



(a) EF_1



(b) EF_2

Figure 28: Sample EAFs

Please note this does not in any way imply that arguments can be assigned only a single permanent "role" in ADFs, e.g. "attacker" or "supporter". The framework supports changes and an argument can on one occasion be required, while quite the opposite on another (XOR and XNOR acceptance conditions are very simple examples). A more accurate description is that an argument should have a defined role "at a point", i.e. w.r.t. a given set of arguments. ADFs ensure consistency, not constancy.

Although our examples used AFNs and EAFs for comparison, similar situations occur when we consider EASs and current BAFs, where the consistency constraint between the relations was dropped. Moreover, due to their advanced systems of indirect attacks, further discussion on them in context of ADFs will be given in Section 9.4. We do not consider the difference in treatment of arguments that both attack and support at the same time to be an error on either side. It is more a side effect of varying intuitions, design choices and permitting or rejecting such behaviors falls into the category with odd and even attack cycles, support cycles or self–attackers. If we were to represent the situation as a propositional formula, it is like comparing atom based and literal based evaluation. The same issue arises when we consider standard and ultimate versions of logic programming semantics, as already noted in [84]. Our answer to this problem is to use a method adapting the inconsistent frameworks to ADFs. We refer to it as the bypass method; in case an argument a is directly supported and attacked by the same argument b, we introduce an auxiliary bypass argument b' for b in a way that b supports b' and b' support a in place of b. Therefore, in our work we will identify subclasses or normal forms of the mentioned frameworks that can be shifted straightforwardly into ADFs and ones that require additional modifications. Although the method will be described more in detail in appropriate sections (see Sections 10.6 and 11.5), we will close the section with showing a sample of it on the considered examples.

Example 36. Let us come back to AFN $FN_1 = (\{a,b\}, \{(b,a)\}, \{(\{b\},a)\})$ from Example 33. By introducing a bypass argument for b and making it support a, we obtain another AFN $(\{a,b,b^b\}, \{(b,a)\}, \{(\{b\},b^b), (\{b^b\},a)\})$. The corresponding ADF $D_1 = (\{a,b,b^b\}, \{C_a = \neg b \wedge b^b, C_b = \top, C_{b^b} = b\})$ depicted in Figure 29a no longer turns a into a falsum argument.

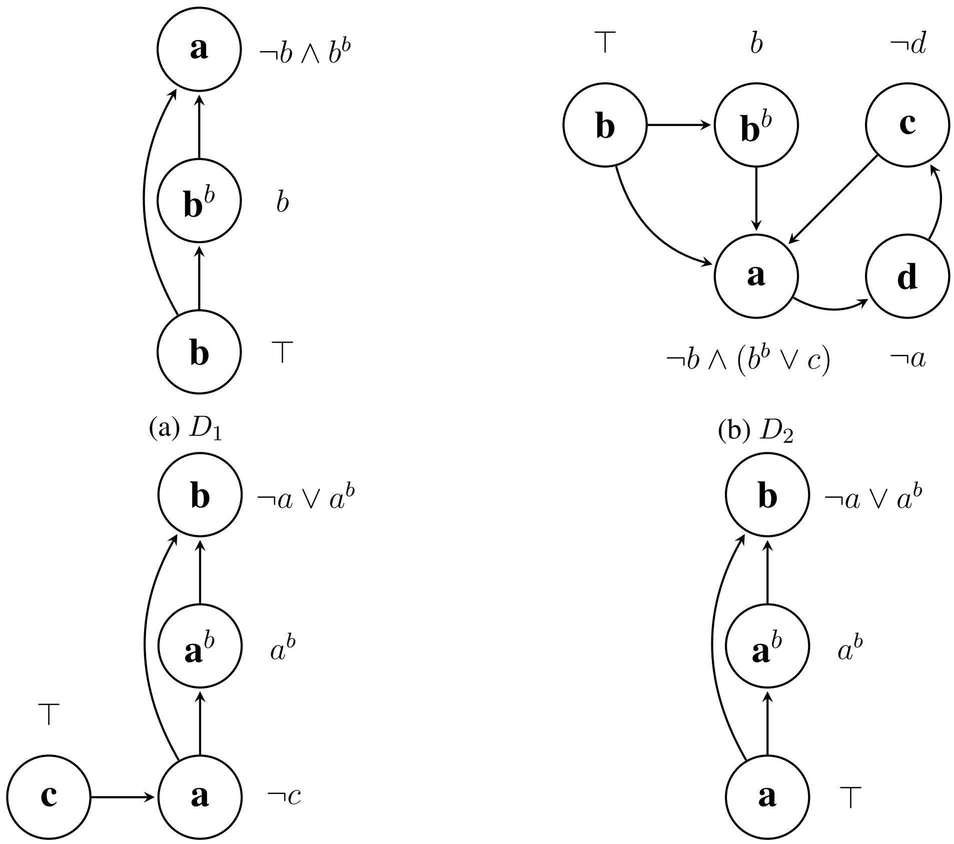
Example 37. Now let us consider $FN_2 = (\{a,b,c,d\}, \{(a,d),(b,a),(d,c)\}, \{(\{b,c\},a)\})$ from Example 34. We again introduce a bypass for b and obtain a framework $D_2 = (\{a,b,c,d,b^b\}, \{(a,d),(b,a),(d,c)\}, \{(\{b\},b^b), (\{b^b,c\},a)\})$ depicted in Figure 29b. It still holds that $\{d\}$ is not AFN admissible. The corresponding ADF is now $(\{a,b,c,d,b^b\}, \{C_a = \neg b \wedge (b^b \vee c), C_b = \top, C_c = \neg d, C_d = \neg a, C_{b^b} = b\})$ and this time $\{d\}$ is no longer admissible; its range interpretation falsifies c only and cannot prevent acceptance of b and b^b, and thus not of a.

Example 38. Finally, we will show the bypass method applied to EAFs from Example 35. For the EAF $(\{a,b,c\}, \{(a,b),(c,a)\}, \{(a,(a,b))\})$, we create a bypass ADF $D_3 = (\{a,b,c,a^b\}$, $\{C_a = \neg c$, $C_b = \neg a \vee a^b$, $C_c = \top$, $C_{a^b} = a\})$ depicted in Figure 29c. It produces the following (arbitrary) admissible extensions - $\emptyset$, $\{c\}$, $\{b,c\}$. We can observe that they are in agreement with the EAF ones. If we go back to the original EAF $(\{a,b\}$, $\{(a,b)\}$, $\{(a,(a,b))\})$, the corresponding bypass ADF $D_4 = (\{a,b,a^b\}$, $\{C_a = \top$, $C_b = \neg a \vee a^b$, $C_{a^b} = a\})$ from Figure 29d gives us sets $\emptyset$, $\{a\}$, $\{a,a^b\}$, $\{a,a^b,b\}$, which after removing a^b return to the desired collection.

2.4 Signatures and Realizability of Argumentation Semantics

The majority of research dedicated to a given argumentation framework is often focused on creating new semantics, analyzing their complexity and producing new algorithms for their computation. However, there is also another line of study, focusing on the expressive power of argumentation frameworks and their semantics [13,37,38,40,43,82,86]. Within this group is the research on realizability, which tries to answer the question whether, given a set of desired extensions, it is possible to create a framework producing exactly this set under a given semantics. Although this line of study has a number of important applications, from the point of view of intertranslatability of argumentation frameworks it is particularly relevant for establishing whether a translation is possible or if an existing transformation can be improved.

Unfortunately, the research on realizability is still quite new. It involves mostly Dung's argumentation frameworks [37,38,43,50] and abstract dialectical frameworks [59,82] and



(a) D_1 (b) D_2 (c) D_3 (d) D_4

Figure 29: Bypass ADFs

is an ongoing line of research. For example, sufficient conditions for realizing complete semantics are still an open research problem. Moreover, the majority of the available studies are focused on single semantics only. By this we understand that while we can establish whether there exists a framework in which a set of extensions E can be realized under a σ–semantics and a framework realizing a set of extensions E' under a different semantics σ', it would be also valuable to know if there is a framework meeting both of these requirements. The multi–dimensional signatures would be very useful in our study due to our focus on generic translations; so far, the only work dealing with this topic is [38]. Finally, the majority of the available studies are, in a certain sense, very "precise". They check whether exactly the given collection of extensions, not something "similar" to it, can be realized under the semantics in question. Although these results are important when comparing the power of two frameworks, from the intertranslatability perspective we would be interested in a more relaxed approach. By this we understand that even if a given set of extensions is not "good enough", by introducing auxiliary arguments (or in the worst case, also extensions) we can derive a new extension set that is. The only exception here is the study in [43], which shows that given enough auxiliary arguments, we can realize any collection of preferred or semi–stable labelings in the Dung's framework. For us this means that due to the relation between the extensions and labelings in AFs stated in Theorem 2.15, with the use of auxiliary arguments any framework can be translated into the Dung's framework under the preferred and semi–stable semantics.

Let us now recall the AF signature and realizability research done in [37]. The signature of a given AF semantics σ is understood as a collection of all sets of σ–extensions that can ever be produced by a Dung's framework. We will now assume that we are working with an argument domain $\mathcal{U}$ and a collection of argumentation frameworks $AF_{\mathcal{U}}$ s.t. for every $X = (A, R) \in AF_{\mathcal{U}}$, $A \subseteq \mathcal{U}$ and A is non-empty and finite.

Definition 2.174. Let $\sigma \in \{$conflict–free, admissible, preferred, complete, stable, grounded$\}$ be an AF semantics. The **signature** of σ is defined as $\Sigma_{\sigma}^{AF} = \{\sigma(F) \mid F \in AF_{\mathcal{U}}\}$.

In what follows we will describe the necessary and sufficient conditions for a set of extensions to belong to the signature of a given AF semantics σ.

Definition 2.175. Let $S \subseteq 2^U$ be a collection of sets of arguments. Then $Arg_S = \bigcup_{E \in S} E$ stands for the collection of all arguments occurring in the sets in S and $Pair_S = \{(a, b) \mid \exists E \in S\, s.t.\, \{a, b\} \subseteq E\}$ is the collection of pairs of arguments that occur in any set in S.

Definition 2.176. Let $S \subseteq 2^U$. The **downward–closure** of S is defined as $dcl(S) = \{E' \subseteq E \mid E \in S\}$. Given a set of arguments $E \subseteq U$, the completion–sets $C_S(E)$ of E in S is the collection of $\subseteq$–minimal sets $E' \in S$ where $E \subseteq E'$. Then S is:

- **downward closed** if $S = dcl(S)$.
- **incomparable** if for each $E, E' \in S$, $E \subseteq E'$ implies $E = E'$.
- **tight** if for all $E \in S$ and $a \in Arg_S$ it holds that if $E \cup \{a\} \notin S$ then there exists an $s \in E$ s.t. $(a, s) \notin Pair_S$.
- **adm-closed** if for each $E, E' \in S$ it holds that if $(a, b) \in Pair_S$ for each $a, b \in E \cup E'$, then $E \cup E' \in S$.
- **com-closed** if for each $T \subseteq S$ the following holds: if $(a, b) \in Pair_S$ for each $a, b \in Arg_T$, then there exists a unique completion–set $E \in C_S(Arg_T)$.

Downward closure means that if we accept a set of arguments, we accept every of its subset as well. Although most of the semantics do not produce sets that are downward closed, conflict–freeness does – it is easy to see that if there are no conflicts in a set of arguments, then there are no conflicts in any of its subsets. The tightness requirement grasps the idea that if an argument does not occur in some extension then there must be a reason for that, the simplest one in the case of AFs being the presence of some conflict. The adm–closed property means that if a union of two extensions is not an extension on its own, then there must be some conflict between the two sets in question. The com–closed requirement is simply meant to ensure the least upper bound property for the semilattice structure of complete semantics (see Theorem 2.10).

Example 39. Taken from [37]. Let us consider the extension set $S_1 = \{\{a,b\}, \{a,c\}, \{b,c\}\}$. It is not downward closed due to the absence of sets $\emptyset$, $\{a\}$, $\{b\}$ and $\{c\}$. It is also not tight, since there is no reason to, for instance, exclude c from extension $\{a,b\}$ ((a,c) and (b,c) are both contained in $Pair_{S_1}$). On the other hand, the set $\{\{a,b\},\{a,c\},\{b,d\},\{c,d\}\}$ is easily checked to be tight.

Let $S_2 = \{\{a,b\},\{a,d,e\},\{b,c,e\}\}$ be a collection of extensions. We can observe that S is adm–closed, since for each pair of extensions, there exists a pair of arguments not contained in $Pair_{S_2}$: $b,d \in \{a,b\} \cup \{a,d,e\}$ and $(b,d) \notin Pair_{S_2}$; $a,c \in \{a,b\} \cup \{b,c,e\}$ and $(a,c) \notin Pair_{S_2}$; $c,d \in \{a,d,e\} \cup \{b,c,e\}$ and $(c,d) \notin Pair_{S_2}$. However, we also observe that S is not tight; the set $\{a,b\} \cup \{e\}$ is not in S, but both (a,e) and (b,e) are contained in $Pair_{S_2}$.

Finally, let us consider $S_3 = \{\emptyset, \{a\}, \{b\}, \{a,b,c\}\}$. The collection is com–closed, in particular, as $C_{S_3}(\{a\} \cup \{b\}) = \{\{a,b,c\}\}$. Observe that since $\{a,b\} \notin S_3$, but $(a,b) \in Pair_{S_3}$, S_3 is not adm–closed.

Although the work in [37] is much more in–depth and shows a number of relations between properties and explains the construction of an argumentation framework for a given set of extensions that meets the desired requirements, we are only interested in the following results:

Theorem 2.177. *The signatures for the considered semantics are given by the following collections of extension–sets:*

- $\Sigma^{AF}_{cf} = \{S \neq \emptyset \mid$ *S is downward–closed and tight*$\}$.
- $\Sigma^{AF}_{stable} = \{S \mid$ *S is incomparable and tight*$\}$.
- $\Sigma^{AF}_{adm} = \{S \neq \emptyset \mid$ *S is adm–closed and contains* $\emptyset\}$.
- $\Sigma^{AF}_{pref} = \{S \neq \emptyset \mid$ *S is incomparable and adm–closed*$\}$.

Although no sufficient conditions are known for the complete semantics, the necessary ones will still be useful to us.

Proposition 2.178. *For each $F \in AF_U$, the set of complete extensions $com(F)$ of F is a non–empty, com–closed extension–set with $(\bigcap_{E \in com(F)} E) \in com(F)$.*

While the last condition might at first seem confusing, it basically describes the grounded extension – the least complete one. Finally, we will recall the results from [43], where in realizing preferred and semi–stable labelings, auxiliary arguments were allowed. Given a labeling v defined over A and a set of arguments $A' \subseteq A$, by $v|_{A'}$ we will denote the subinterpretation of v defined over A'.

Theorem 2.179. *Let $A \subseteq \mathcal{U}$, Lab_A a collection of three–valued labelings over A and $L \subseteq Lab_A$. There exists a finite AF F s.t. for $\sigma \in \{$preferred, semi–stable$\}$, $L = \{v|_A \mid v \in \sigma(F)\}$.*

With the following example we close the background section of our work.

Example 40. Let us come back to Example 39. We could have observed that the set $\{\{a, b\}, \{a, c\}, \{b, d\}, \{c, d\}\}$ was tight. As it is also incomparable, it fits the stable signature. By adding the extensions $\emptyset$, $\{a\}$, $\{b\}$, $\{c\}$ and $\{d\}$, we create a collection that is still tight, but additionally downward closed. Thus, there is a framework producing exactly the conflict–free and stable extensions that we have listed. In particular, the AF $(\{a, b, c, d\}, \{(a, d), (b, c), (c, b), (d, a)\})$ satisfies these requirements.

Let us come back to the adm–closed collection $S_2 = \{\{a, b\}, \{a, d, e\}, \{b, c, e\}\}$. As it is already incomparable, it fits the preferred signature. For example, the framework $(\{a, b, c, d, e, f\}, \{(a, c), (c, a), (b, d), (d, b), (c, f), (d, f), (f, e), (f, f)\})$ produces S_3 under the preferred semantics.

We can now consider the collection $S_3 = \{\emptyset, \{a\}, \{b\}, \{a, b, c\}\}$. By adding the set $\{a, b\}$, we can make it adm–closed. As it already contains $\emptyset$, it now fits the admissible signature. One of the frameworks producing such a set of admissible extensions is $(\{a, b, c, d, e\}, \{(a, d), (b, e), (d, d), (e, e), (d, c), (e, c)\})$.

3 Introduction to Translations

A translation is, simply speaking, a way to convert one formalism into another. It can also be seen as just a function, where to an element A of a given type we assign an element B of another, though not necessarily different, type. In our case we will be interested in translations between argumentation frameworks and assume framework types $\{AF, SETAF, AFRA, EAS, BAF, AFN, EAFC, EAF, ADF\}$, as explained in Section 2. Moreover, unless stated otherwise, we will focus on those frameworks that have a finite set of arguments.

The research on the translations between argumentation frameworks can be roughly split into three main strands. One concerns the analysis of argumentation semantics; for example, we can transform a given AF into another AF s.t. the stable extensions of the former coincide with the preferred ones of the latter. We are thus working with a single framework type, but different source and target semantics. This is referred to as the *intertranslatability of argumentation semantics* and has been studied only in the context of AFs [42]. Another strand is somewhat dual; we translate a framework of one type into another so that the source extensions are in a relation to the target ones under a given semantics. For example, we might want to transform a given EAS into an AF s.t. the stable extensions of the latter can be transformed into the stable extensions of the former. Thus, we deal with different framework types, but use similar semantics. This is in general what we will understand as the *intertranslatability of argumentation frameworks* and what will be the primary focus of the remainder of this work (Sections 5 to 12). To the last case, in which we work with the same framework type and similar semantics, we will refer to as the *normal form translations*. The purpose of these approaches is to obtain a framework that is either "equivalent" or as close to the original one as possible, but has some desirable structural properties. For example, we can transform a given AFN into an AFN in which every argument possesses a powerful sequence. We will analyze these methods in Section 4.

Although translations can be seen as functions mapping one framework to another, we are usually not interested in just some random assignments. We use and create translations for a *reason*, and depending on this reason we want them to have certain *properties*. Moreover, there is often more than one way to transform one framework into another. It is thus natural to ask what are the differences between the available approaches both on the conceptual and the "physical" level. The majority of this section will be devoted to addressing this issue. We will give an abstract definition of a translation, describe the groups of properties that will be of interest to us in the context of this work, and provide a rough classification of the transformation types. Please note that in some cases it will be difficult to give concrete definitions due to differences between the frameworks. However, we hope that the provided discussion and analysis will clearly show our intents and motivate the design choices.

Following this introduction to our translation classification system, starting from Section 4 we will be introducing both new and existing translations. When necessary, we will also extend and complete the results available in the literature. We will also provide the property analysis of all the approaches and possible ways of improving them. Additionally, we will discuss the normal forms that the translated framework might possess, both depending and irrelevant of the forms of the source framework. We therefore create a comprehensive compendium on the intertranslatability of argumentation frameworks that consists of almost ninety methods. Please note that we will use two schemes in this work for describing translations and their associated semantics theorems. Along the classical "if then" definitions, we will use our own system which explicitly states the relevant domains and properties. Thanks to this distinction, the readers looking for a particular translation do not have to familiarize themselves with the entirety of our work and the big picture is still there for those interested in it. Moreover, it also makes the comparison with the existing results much easier. However, what also has to be mentioned is that the usual definitions can sometimes contain methods both for retrieving source extensions from target ones and producing target extensions from the source ones. Our redefinitions will focus solely on the target–to–source directions.

3.1 Translation & Casting Function

In this section we will give the definition of a framework translation, define the similarity relation between the argumentation semantics and introduce the notion of a semantics casting function that will be used to transform the target extensions (or labelings) into the source ones. However, before we do so, we need to discuss a certain issue first.

In Section 2 we have explained that all argumentation frameworks share the set of abstract arguments in their definition. The keyword here is *abstract*. Since no content of the arguments is stored, this means that two argument domains become somewhat indistinguishable due to their abstractness. However, it is quite often the case that the target arguments can in fact represent structures built from the source arguments and possibly other elements of the original framework (see for example Sections 6.1, 10.1, 11.1 and 12). Therefore, we need to be able to say if and how one domain is different from another. Without it, the comparison of the answers produced by the source and target semantics can become impossible.

This means that in some ways, our work is not entirely contained in the field of abstract argumentation. Some of the approaches we will present can be qualified as meta–level argumentation [18, 28, 30, 64]. However, many of them are also not dissimilar to instantiation–based argumentation methods [24, 26]. In this context, a translation can be perceived as an additional "loop" in the argumentation process. An argumentation framework, which is instantiated with a given knowledge base, now itself becomes a source for instantiating the target framework. The resulting target extensions then need to be reinterpreted w.r.t. the source arguments before both of them can be compared, similarly as the source extensions would be reinterpreted w.r.t. the underlying knowledge base. In such a case, the difference between abstract domains can be seen as the difference between the underlying knowledge bases or other data sources with respect to which the extensions need to be recast. This implies that we have e.g. some auxiliary data structure storing the content of target arguments. We leave it to the reader to decide how he or she wants to view the argument domains. It is only important that we can mark given domains as different and have some means of shifting between them

We can now define what we understand as a framework translation. Bearing in mind the previous discussion, we will make it explicit with what types of framework we work with and what are the argument domains they are built on.

Definition 3.1. Let T, T' be two distinct types of abstract argumentation frameworks, $\mathcal{U}^T$,$\mathcal{U}^{T'}$ two domains of arguments and $Fr^T, Fr^{T'}$ the collections of all frameworks of type T and T' on the respective domains. A **framework translation** is a function $Tr : D^T \to D^{T'}$, where $D^T \subseteq Fr^T$ and $D^{T'} \subseteq Fr^{T'}$.

Previously, we have mentioned that we look for translations s.t. the target and source extensions are related and the semantics on both sides are similar. We will now specify what *similar* and *related* mean. Concerning the first issue, not all of the semantics are named the same in all frameworks, best example being the presence of subtypes in ADFs. Moreover, not all of the semantics defined for one framework are also introduced in another – while we have e.g. strongly coherent (i.e. coherent and conflict–free) extensions in AFNs, we only have conflict–freeness in AFs. Therefore, to this end we introduce the similarity relation between semantics of different frameworks. Please note that this relation is in principle not transitive, though it is symmetric.

Definition 3.2. Let T, T' be two distinct types of abstract argumentation frameworks and σ_T and $\sigma_{T'}$ their semantics. We define the **similarity relation** between the semantics, denoted $\sigma_T \sim \sigma_{T'}$, the following way:

- if $\sigma_T \sim \sigma_{T'}$, then $\sigma_{T'} \sim \sigma_T$.
- for $T, T' \in \{AF, SETAF, AFRA, AFN, EAS, EAF, EAFC\}$ and $\sigma_T, \sigma'_T \in$ {conflict–free, admissible, complete, preferred, grounded, stable}, $\sigma_T \sim \sigma_{T'}$ if $\sigma_T = \sigma'_T$.
- for $T \in \{AF, SETAF, AFRA\}$, $cf_T \sim str\text{-}coh_{AFN}$.
- for $T \in \{AF, SETAF, AFRA\}$, $cf_T \sim cf\text{-}sup_{EAS}$.
- $str\text{-}ssup_{EAS} \sim str\text{-}coh_{AFN}$.
- for $T \in \{AF, SETAF, AFRA\}$:
  - $cf_T \sim cf_{ADF}$ and $cf_T \sim acy\text{-}cf_{ADF}$,
  - $stb_T \sim stb_{ADF}$ and $stb_T \sim mod_{ADF}$,
  - $grd_T \sim grd_{ADF}$ and $grd_T \sim acy\text{-}grd_{ADF}$, and
  - for $\sigma_T \in$ {admissible, complete, preferred} and $x, y \in \{a, c\}$, $\sigma_T \sim xy\text{-}\sigma_{ADF}$.

- for $T \in \{AFN, EAS\}$:
  - $coh_{AFN} \sim pdc_{ADF}$ and $ssup_{EAS} \sim pdc_{ADF}$,
  - $str\text{-}coh_{AFN} \sim acy\text{-}cf_{ADF}$ and $str\text{-}ssup_{EAS} \sim acy\text{-}cf_{ADF}$,
  - $grd_{AFN} \sim acy\text{-}grd_{ADF}$ and $grd_{EAS} \sim acy\text{-}grd_{ADF}$,
  - $stb_{AFN} \sim stb_{ADF}$ and $stb_{EAS} \sim stb_{ADF}$, and
  - for $\sigma_T \in \{$admissible, complete, preferred$\}$, $\sigma_T \sim aa\text{-}\sigma_{ADF}$.
- for $T \in \{EAF, EAFC\}$:
  - $cf_T \sim cf_{ADF}$,
  - $stb_T \sim mod_{ADF}$,
  - $grd_T \sim acy\text{-}grd_{ADF}$, and
  - for $\sigma_T \in \{$admissible, complete, preferred$\}$, $\sigma_T \sim ca_2\text{-}\sigma_{ADF}$.

We can observe that the BAF semantics were not included in this listing. This is due to the fact that they can be parametrized in various ways and depending on the used indirect conflicts, the similarity might or might not be there. Thus, in this case we will depend more on the naming convention, and the precise parametrization will be provided in the relevant sections.

We can now focus on the relation between the answers produced by the semantics in the target and source frameworks. Now, it can happen that this connection is trivial. For example, the produced answers can perfectly coincide. This is typically the relation we aim for if we intend to compare the expressive power of given formalisms. Due to the nature of the research on the semantics signatures, it is also the relation we focus on when we are concerned with establishing whether a given translation is possible or not.

However, if we want to use the target framework as a "solver" for the source one, then we want to have a working translation, even if it is not a simple one. In many cases, the argument domains of the source and target structures might be different and the relation between the extensions much more complicated. If we are lucky, we might only have to remove some auxiliary arguments. Nevertheless, the target arguments might also represent e.g. sets of source arguments (see Section 3.3) or inference trees [81]. Additionally, they can also carry pieces of information that are relevant for the construction of the translated framework, but are not important from the semantics point of view (see Section 12), which complicates the matters even further.

This wide scope of the relations between the extensions calls for some sort of a function that would allow us to cast the target answers into the source ones. One could thus say that the extensions are "related" if it is possible to define a function SC s.t. $\sigma(X) = SC(\sigma(Tr(X))$, where Tr is a translation, X a source framework and σ a semantics. We will refer to it later as the *collective* approach. Unfortunately, this definition allows more than we would like, as it permits addition and removal of certain extensions.

In our study we want to focus on manipulating single extensions to retrieve the desired ones. Thus, the formula $\sigma(X) = \{SC(E) \mid E \in \sigma(Tr(X))\}$ is closer to what we want the "relation between extensions" to stand for. We will refer to this as the *singular* approach. Please note that casting an extension may require access to the source or target framework, for example in case some arguments need to be removed. Thus, if we want to use SC for computing the actual extensions, labelings or any sort of answers produced by the semantics, we need to take the frameworks into account as well.

Definition 3.3. Let T, T' be two distinct types of abstract argumentation frameworks and $Tr : D^T \to D^{T'}$ a translation between them, where D^T and $D^{T'}$ are collections of frameworks of given types. Let σ^T and $\sigma^{T'}$ be two similar semantics and $X \in D^T$ a source framework.

The **singular semantics casting function** for X under translation Tr and semantics σ^T is a function $SC^X_{\sigma^T} : \sigma^{T'}(Tr(X)) \to \sigma^T(X)$ mapping target answers to source answers.

The **collective semantics casting function** for X under translation Tr and semantics σ^T is a function $CSC^X_{\sigma^T} : \{\sigma^{T'}(Tr(X))\} \to \{\sigma^T(X)\}$ mapping the set of all target answers to a set of source answers.

By $SC^{Tr}_{\sigma^T}$ ($CSC^{Tr}_{\sigma^T}$ respectively) we will denote the assignment of a singular (collective) semantics casting function to every $X \in D^T$.

Please note that one can define the collective function in a simpler manner, as in its current state it can be seen as just a single pair rather than a function. However, we prefer to use consistent notation between the approaches. From now on, by semantics casting function we will understand the singular one, unless stated otherwise.

Let us now introduce the concept of the strength of a translation. Not in all cases casting the target extensions will give us all and only the answers we wanted. It can happen that we end up with too many or not enough extensions. This brings us to the following definition:

Definition 3.4. Let T, T' be two distinct types of abstract argumentation frameworks and $Tr : D^T \to D^{T'}$ a translation between them, where D^T and $D^{T'}$ are collections of frameworks of given types. Let σ^T and $\sigma^{T'}$ be two similar semantics and $SC^{Tr}_{\sigma^T}$ the singular semantics casting functions for D^T. The translation Tr is then a:

- **$\subseteq$–weak** translation under $(\sigma, SC^{Tr}_{\sigma^T})$ if for every $X \in D^T$, $\sigma^T(X) \subseteq SC^X_{\sigma^T}[\sigma^{T'}(Tr(X))]$.[11]
- **$\supseteq$–weak** translation under $(\sigma, SC^{Tr}_{\sigma^T})$ if for every $X \in D^T$, $\sigma^T(X) \supseteq SC^X_{\sigma^T}[\sigma^{T'}(Tr(X))]$.
- **strong** translation under $(\sigma, SC^{Tr}_{\sigma^T})$ if for every $X \in D^T$, $\sigma^T(X) = SC^X_{\sigma^T}[\sigma^{T'}(Tr(X))]$.

[11] Recall that for a function $f : D \to C$, $f[D]$ denotes the image of D, i.e. the set $\{c \in C \mid c = f(d)$ for some $d \in D\}$.

We will of course be interested in the collections of extension functions that are "of the same type". Although in certain cases the SC function might be a bit more involved, in majority it will be *identity*, *removal* or *union* casting, or possibly a combination of them. In the case of labeling–based semantics, we will also consider the *addition* function.

The first (and the simplest) type – the identity casting – basically returns the extensions the way they are, without any further modifications. This of course implies that our semantics domains are the same, though please note it does not necessarily mean that the argument domains are the same as well. Although it is normally a safe assumption, there are exceptions. Not in all frameworks semantics produce answers dependent on arguments only. For example, AFRA semantics return sets of arguments and attacks. Consequently, while an AF obtained from an AFRA has a different domain of arguments than the source framework, the semantics domains is the same and the identity casting function can be used (see Section 7.1.1).

The removal casting function is used when we have to "filter out" auxiliary elements showing up in the target extensions or labelings. These elements can be additional arguments, conflicts, and more. For the removal to work, we have to assume that the source and target semantics domains are related. We believe a subset relation between the two to be adequate. However, formally defining what "filtering out" means can be a bit tricky. If we work with extension–based semantics, we simply remove arguments. Therefore, the casted target extension is in fact a subset of the target set before casting. If we deal with labelings, we need to remove assignments. Hence, the casted labeling is a subinterpretations of the target one. Semantics can produce answers of various types and researching the more complex ones might force us to use additional notions. We will thus introduce a new operator $\subseteqq$ to denote this general containment relation. For extensions and labelings, it will have the meaning we have just explained. Instead of redefining the casting function in the future, we can simply extend this operator to handle other types of answers.

The addition casting function is dual to removal and will primarily be used in the context of the labeling–based semantics. On various occasions, a translation can delete certain arguments from the source framework (see e.g. Section 4.3). Very often the arguments qualifying for removal are those that would not appear in any extensions, which means that most of the semantics are not affected by this modification. However, in the labeling–based semantics, the produced interpretation is defined on all arguments in framework. Therefore, the casting function has to revert the removal caused by the translation, and thus the target results need to be "extended back" to the original ones.

Finally, we have the union casting function, which is particularly useful in e.g. coalition translations. In this approach we assume that the target arguments are sets of source arguments. As a result, the target extensions are now sets of sets of source arguments. Thus, performing a union of all the elements in the target set gives us the source extension. Please note that in the case of labelings, the situation is a bit more complicated; a given source argument can appear in a number of target set arguments, including some being accepted and some being rejected. Consequently, while the sets of arguments mapped to in, out and $undec$ in the target are disjoint, the $\bigcup in$, $\bigcup out$ and $\bigcup undec$ do not have to be. Retrieval of the original labeling becomes more complicated and since the only labeling related translations we will deal with concerns ADFs, we will come back to this problem in Section 12.1.5. For now, we will give a simplified definition suitable only in the extension–based approaches. The main types of casting functions are illustrated in Example 41.

Definition 3.5. The **sub relation** $\subseteq$ is defined the following way. For two sets A and B, $A \subseteq B$ iff $A \subseteq B$. For two labelings v and v' on sets A, A' respectively, where $A \subseteq A'$, $v \subseteq v'$ iff $\forall a \in A, v(a) = v'(a)$.

Definition 3.6. Let T, T' be two distinct types of abstract argumentation frameworks, Tr a translation between them, σ^T and $\sigma^{T'}$ two similar semantics on domains U_{σ^T} and $U_{\sigma^{T'}}$ respectively, X an argumentation framework of type T and $SC^X_{\sigma^T}$ its semantics casting function. The function is:

- an **identity** casting if $U_{\sigma^T} = U_{\sigma^{T'}}$ and for any $E \in \sigma^{T'}(Tr(X))$, $SC^X_{\sigma^T}(E) = E$.
- a **removal** casting if $U_{\sigma^T} \subseteq U_{\sigma^{T'}}$ and for any $E \in \sigma^{T'}(Tr(X))$, $SC^X_{\sigma^T}(E) \subseteq E$.
- an **addition** casting if $U_{\sigma^{T'}} \subseteq U_{\sigma^T}$ and for any $E \in \sigma^{T'}(Tr(X))$, $E \subseteq SC^X_{\sigma^T}(E)$.
- a **union** casting if $U_{\sigma^{T'}} = 2^{U_{\sigma^T}}$ and for any $E \in \sigma^{T'}(Tr(X))$, $SC^X_{\sigma^T}(E) = \bigcup E$.

Please observe that identity can be seen as a special type of a removal and addition casting. This is by design; just because for a given translation the extension casting functions will be in majority strictly removal (or strictly additions), there will always be some framework for which filtering out or adding arguments will not be necessary. We will say that the casting functions for a translation are removals (or additions) if there are some frameworks for which the functions are not identities.

Example 41. Let us consider a very simple translation from AF to SETAF (Translation 17). It only involves changing binary attacks to single–element set attacks. The SETAF representation of an AF $(\{a, b, c\}, \{(a, b), (b, a), (b, c)\})$ is $(\{a, b, c\}, \{(\{a\}, b), (\{b\}, a), (\{b\}, c)\})$. The admissible extensions of both frameworks are the same, namely $\emptyset$, $\{a\}$, $\{b\}$ and $\{a, c\}$. By Theorem 5.2 the extensions will agree for all AF–produced SETAFs. This is thus a strong translation under admissible semantics and identity casting.

Let us now consider the same AF as an AFRA (Translation 18). The translated framework is identical its source, we only label the attacks: $(\{a, b, c\}, \{r_1 = (a, b), r_2 = (b, a), r_3 = (b, c)\})$. The complete extensions of the AF are $\emptyset$, $\{b\}$ and $\{a, c\}$. However, the AFRA ones are $\emptyset$, $\{b, r_2, r_3\}$ and $\{a, c, r_1\}$. We can thus observe that our semantics casting function is a removal and we need to get rid of attacks from the AFRA extensions in order to retrieve the AF ones.

Let us now consider a simple AFN $(\{a, b, c, d\}, \{(c, d), (d, a)\}, \{(\{a, b\}, c\})$. In the coalition translation for AFNs to AFs (Translation 61), for every AFN argument we create

AF arguments representing its minimal coherent sets. Since a, b and d require no support, we create the set arguments $\{a\}, \{b\}$ and $\{d\}$ for them. The argument c has two minimal coherent sets; $\{a, c\}$ and $\{b, c\}$. Finally, any argument containing c will have to carry out an attack on $\{d\}$; similarly for the (d, a) attack. Thus, our corresponding AF is $(\{\{a\}, \{b\}, \{d\}, \{a, c\}, \{b, c\}\}, \{(\{a, c\}, \{d\}), (\{b, c\}, \{d\}), (\{d\}, \{a\}), (\{d\}, \{a, c\}\})$. We can observe $\{a, c\}$ is an admissible extension of our AFN, while the AF produces $\{\{a, c\}\}$ and $\{\{a\}, \{a, c\}\}$ for it. We use union casting to retrieve our original extension.

In our research we will consider one more type of semantics casting function, to which we will refer as *extraction*. This type is specific to transformations from ADFs to other frameworks and will be described in Section 12. The intuition behind it is that the arguments in the target structure contain information relevant for the construction of a framework, but not for its evaluation. Consequently, the data important from the semantics perspective needs to be extracted from the arguments first and only later analyzed in the context of the source extensions or labelings.

3.2 Properties of Translations

In this section we will describe the properties of translations. We will distinguish four main attribute groups – functional, syntactical, semantical and computational. The functional properties concern looking at a translation more as a function, analyzing its domain, codomain and the uniqueness of the mappings. The syntactical properties focus on changes in the structure of the framework during translation. Please note that by this we understand something more than just "this framework has support, this one does not" or "this one has group attack, the other only binary". Our focus is on problems like the change of argument domain, the loss or gain of arguments and relations, and whether the elements of the source framework are uniquely represented in the target structure or not. The next group of properties concerns the semantics of the frameworks, i.e. whether all extensions can be retrieved or too many/not enough are produced by the target framework, for how many semantics does the translation "work", is it exact, faithful and so on. This analysis will be most important to us. The final group of properties covers the translation difficulty, the knowledge required to perform the transformation, its computational complexity, modularity, and difference in size of the source and target framework. Although this group is important for practical purposes, we will mostly focus on other types of properties.

3.2.1 Functional Properties

Let us start with the functional properties of a translation. First of all, a transformation can be defined for all the frameworks of a given type, or only for a group of them meeting certain requirements. There can be various reasons for such limitations, one of them being that a particular framework subclass can be transformed more easily. The most trivial example involves a situation when one framework generalizes the other; we can shift an AF to an AFN easily, while going back from arbitrary AFNs to AFs requires a bit of work. However, if we considered only AFNs with the empty support relation, which basically represent Dung's frameworks (see Sections 5 and 10.1), then the problem would be nonexistent. We will thus refer to translations that can handle any framework of a given type as *full*, and to ones that can not, as *partial* or *source–subclass*. In contrast, we can now ask ourselves whether the translation can produce any framework of the target type, or will they all belong to a particular subclass. Again, to give the most basic example, an AF translated to a SETAF (see Section 5) will not need to use sets of attacks of size bigger than 1. Should it be possible to obtain any target framework, then the translation will be basically a *surjection*. In any other case, we will refer to it as the *target–subclass*. Since the subclass descriptions, both source and target, depend on the frameworks we are dealing with, we will not define any abstract properties they might have here and refer the reader to the appropriate sections.

Although already the target–subclass property can tell us a bit about the relative strength of the frameworks, another interesting question concerns whether it is possible to obtain the same target framework with more than one source structure. A positive answer can point to loss of some data contained in the initial structure. Moreover, in this case we would also like to know if there is any particular relation between the source frameworks that would allow us to pinpoint the cause for this similarity. We will refer to translations in which a target framework can be obtained only from a single source structure as *injective*. Otherwise, we will say that the translation is *overlapping*. This brings us to the following definition:

Definition 3.7. Let T, T' be two distinct types of abstract argumentation frameworks and $Fr^T, Fr^{T'}$ the collections of all frameworks of type T and T'. A framework translation $Tr : D^T \to D^{T'}$, where $D^T \subseteq Fr^T$ and $D^{T'} \subseteq Fr^{T'}$, is:

- **full** if $D^T = Fr^T$.
- **partial** or **source–subclass** if $D^T \subseteq Fr^T$.
- **surjective** if $D^{T'} = Fr^{T'}$.
- **target–subclass** if $D^{T'} \subseteq Fr^{T'}$.
- **injective** if for all $X, X' \in D^T$, if $Tr(X) = Tr(X')$ then $X = X'$.
- **overlapping** if there exists $X, X' \in D^T$ s.t. $X \neq X'$ and $Tr(X) = Tr(X')$.

3.2.2 Syntactical Properties

This group of properties concerns the syntactical changes the source undergoes during the transformation to the target framework. First of all, the type of arguments in the target framework might not be the same as in the original one. Moreover, even if they are, we might want to have access to additional arguments s.t. we know they cannot appear in the source domain. Due to the fact that the structure of arguments is in no way stored by an abstract argumentation framework, it is important to note on the domain change, since it implies we need to additionally store the change somewhere. Typically, frameworks with support are translated into AFs by the use of coalitions, i.e. the arguments in a resulting AF correspond to sets of arguments of the source structure. Although their content is related to the elements in the initial framework, each source argument can be represented by multiple ones in the target structure. Similarly, certain arguments may not be represented at all, for example arguments not possessing coherent sets will not appear in any AF argument translated from an AFN (see Translation 61). Independently of the argument types, there might also be a need for additional arguments, possibly not related to the source ones, even if they do have some meaning attached. Similar analysis can be done in case of relations, however, it is difficult to give any concrete definitions without taking into account the frameworks in question. Consequently, the provided definitions will not be very precise at this point, but more details will be given when dealing with particular translations.

Definition 3.8. Let T, T' be two distinct types of abstract argumentation frameworks, D^T, $D^{T'}$ collections of the frameworks of the respective types and U^T, $U^{T'}$ their argument domains. A framework translation $Tr : D^T \to D^{T'}$ is:

- **argument domain preserving** if $U^T = U^{T'}$.
- **argument domain altering** if $U^T \neq U^{T'}$.
- **weakly domain altering** if $U^T \subseteq U^{T'}$ or $U^T \supseteq U^{T'}$ and elements in U^T and $U^{T'}$ are of the same type.
- **argument introducing** if not every argument in the framework $Tr(X)$ represents an argument from a framework $X \in D^T$ or arguments in X can be represented by more than one argument in $Tr(X)$.
- **argument removing** if there is an argument in a framework $X \in D^T$ that is not represented by any argument in the framework $Tr(X)$.
- **relation introducing** if not every relation of a given type occurring between given arguments in the framework $Tr(X)$ represents a relation in framework $X \in D^T$ or relation in X can be represented by more than one relation in $Tr(X)$.
- **relation removing** if not every relation of a given type occurring between given arguments in a framework $X \in D^T$ is represented by a relation in the framework $Tr(X)$.
- **induced relation removing** if a relation is removed only if one of its arguments is removed.
- **induced relation adding** if it is argument introducing an a relation is added only if one of its arguments is also an added one.

- **structure preserving** if no elements are removed or added during the translation.

Whenever a translation is removing some elements, we will just refer to it as **lossy**. Please note that while preserving and altering a domain are mutually exclusive properties, it is not necessarily the case with removal and introduction of arguments and relations. One can remove arguments or relations that are e.g. not valid due to support cycles, but still introduce auxiliary elements later in order to be able to translate the framework into a different one. For example, translating an ADF into an AF w.r.t. the cc–family of semantics (see Translation 85) introduces auxiliary arguments due to support cycles, but also arguments not possessing standard evaluations (i.e. falsum arguments and ones building up on them) do not appear in the target framework.

Finally, please note that our list is by no means exhaustive. In [42] properties such as covering, embedding and monotonicity were introduced. Covering and embedding are related to how removing and introducing argument elements works. However, since they were defined with AFs as source and target frameworks, their definitions are not easily shifted into our setting and thus we believe that the current notions are sufficient for our purposes. Although monotonicity of a translation is an interesting notion, its study will be left for future work.

3.2.3 Semantical Properties

The point of semantical properties is to describe how the semantics behave between the target and source framework. Therefore, they depend not only on the translation itself, but also on the semantics casting function we decide to use along with it. This means that a single translation, when associated two different casting functions, may exhibit different semantical properties. Additionally, we can observe that this group of attributes cannot be clearly separated from syntactical properties, since the structure of the framework clearly affects the extensions it produces. The answer to the question of how we need to *manipulate* the extensions of the target framework to retrieve the ones from the source framework depends on how the target framework *looks like*, particularly compared to the initial one. A change in the arguments domain and using auxiliary elements can have a major impact on the extensions. Nevertheless, there are a number of properties that can be clearly separated from the syntactical ones.

The first of the semantical properties we will take into account is whether a given translation and its semantics casting functions are specialized to work (i.e. be strong) for a particular semantics, or whether they can be more general and hold for a number of them. For example, in order to retrieve SETAF extension from an EAS, we only need to remove the evidence argument from the produced sets. Such a translation works for all the usual semantics (see Section 6.4). If we transform an AFN into an AF (see Section 10.1), we have one easy translation aimed at conflict–freeness and another, somewhat more complicated and expensive, preserving a number of semantics. Thus, in our work, we will distinguish between *specialized* and *generic* translations, and focus on the latter. However, please note that the definition of these properties will not be very formal. From the technical side, the extension casting functions defined for e.g. the admissible and complete semantics in the SETAF–EAS example would be different due to the change of domains and codomains caused by the change of semantics. However, it is easy to see that they follow the same principle (removal of evidence) and since every complete extension is admissible for both frameworks, the complete semantics casting functions would be "contained" in the admissible ones. Thus, by saying that a translation is generic, we will assume that the casting functions associated to the semantics follow the same principles. Finally, we will set the border between the specialized and generic approach at two, i.e. translations handling two or less semantics will be qualified as specialized, while three or more as generic. This choice is more a matter of taste and observation rather than an established rule. Our translations emerge from analyzing the differences between the target and source frameworks and the design choices behind their semantics, not by having a single given semantics in mind. Thus, even if they can be sometimes complicated, in general they work for at least three major semantics, namely complete, preferred and grounded (usually also stability is included). Consequently, this is where we decided to set the border.

Another property concerns the semantics domains, which can agree or be different independently of the arguments domains. For example, one can use different source and target semantics families, such as extension–based on one side and labeling–based on the other. Thus, the semantics domains can differ even if the argument ones do not. Although in our approach we will use the same source and target semantics classes, special cases can be found even under this restriction. While in general it does hold that a different semantics domain implies a different argument domain if we stay within the same semantics class, the AFRA framework is an exception to this rule. In AFRA, the extensions contain both arguments and attacks, not just arguments. When translated into AFs, the attacks form new arguments and thus the corresponding AF will have a semantics domain the same with AFRA, but a different argument one (see Section 7.1). Therefore, the change in the semantics domain needs to be distinguished from the change in the argument domain.

The strength of a translation and the nature of the semantics casting functions can also be seen as semantical properties. The strength tells us if all extensions (or any other answers), too many or not enough are produced by the target framework, while the use of a particular casting function can tell us whether e.g. there are auxiliary arguments showing up in the extensions. Please note that the fact that the framework itself can use auxiliary arguments does not imply that the casting function will be a removal. For example, if the auxiliary arguments are just self–attackers, they will never show up in the extensions (see Section 12.1.3 for an example). Consequently, just like in the case of semantics domain, this question is separate from the syntactical counterpart.

Although the properties mentioned above are interesting and should not be neglected, one of the most important semantical properties concerns faithfulness and exactness of the translation. Faithfulness, along with modularity and polynomiality, is one of the most studied properties of translations and appears also in research on intertranslatability of various nonmonotonic reasoning formalisms, including default logic, autoepistemic logic and more [52, 57, 58]. As a consequence, there is also no single definition of faithfulness. The notion is often overloaded or specialized for a given formalism, giving rise to formulations that are not always equivalent:

> *Faithful translations can be defined in two ways, which are equivalent when new variables are not allowed. In particular, a translation is faithful if each theory T_1 is translated into a theory T_2 such that either:*
>
> 1. *between the extensions of T_1 and the extensions of T_2 there is a bijection such that the associated extensions of T_1 and T_2 are equivalent, or*
> 2. *for every extension of T_1 there exists an equivalent extension of T_2 and vice versa.*
>
> *These two definitions can be adapted to the case in which new variables are allowed by replacing "equivalence" with "var–equivalence". However, they no longer coincide. Indeed, the second definition allows a single extension of T_1 to be associated to several extensions of T_2.* ([58])

By combining the two approaches with various notions of equivalence, we can obtain a number of definitions of faithfulness. In what follows we will use two of them and show that what we understand as faithfulness recreates the research done on intertranslatability of semantics [42]. In this approach, a translation in which both the source and target extensions were precisely the same was called *exact*, while a bijective one where we could retrieve the source extensions by removing auxiliary arguments from target ones was referred to as *faithful*.

We may notice that first and foremost a faithful translation has to be strong – we can obtain every desired source extension and there are no unrelated target ones produced. Further restrictions that we can add concern the semantics domain, argument domain and the semantics casting function. In [57], faithfulness requires that the extensions of the initial theory, which is based on some language L, and the extensions of the translated one coincide up to L. This means that even if the used language extends L or we have some auxiliary elements showing up, the extensions can be projected w.r.t. L. However, since we are working with abstract argumentation, no actual content of the arguments is stored by the framework and thus it is difficult to speak about faithfulness without assuming that the argument domains are similar between the source and the target frameworks. Any domain projection would require us to have the access to some auxiliary data structure in which contents of arguments are stored, and allowing such access for projection purposes would go against the meaning of faithful translations. As the semantics domain depends on (at least) the argument domain, we will have sufficient means to compare the extensions between our structures and no further assumptions need to be done for now.

Let us now analyze the issue of equivalence. What the semantics produce can be equivalent in a number of ways, without even going into auxiliary arguments. For example, we can work with the complete extensions on one framework and assume the complete labelings on the other. Even though this would be a one–to–one relation in case of e.g. AFs, we cannot put a simple "=" between the two approaches. However, since in this work we are not interested in mixing extension–based and labeling–based semantics, we will provide our definitions under the assumption that the source and target semantics belong to the same class. This means that for "pure" equivalence we can simply use the identity extension casting function, which basically brings us to the definition of *exact* translations [42]. If we allow auxiliary variables, we can use removal casting in the definition. The bijective relation between the extensions can be handled by requiring the semantics casting functions to be bijective. By joining removal and bijection, we obtain the faithful translations as defined in [42].

Definition 3.9. Let T, T' be two distinct types of abstract argumentation frameworks, D^T, $D^{T'}$ collections of the frameworks of the respective types and U^T, $U^{T'}$ their argument domains. Let σ^T and $\sigma^{T'}$ be two similar semantics on domains U_{σ^T} and $U_{\sigma^{T'}}$, $Tr : D^T \to D^{T'}$ a framework translation and $SC^{Tr}_{\sigma^T}$ the semantics casting functions for Tr and semantics σ^T. We say that Tr is:

- **generic** if the translation and the semantics casting functions can be applied to three or more semantics in a strong manner.
- **specialized** if the translation and the semantics casting functions can be applied to two or less semantics in a strong manner.
- **semantics domain preserving** for σ^T if $U_{\sigma^T} = U_{\sigma^{T'}}$.
- **weakly semantics domain altering** for σ^T if $U_{\sigma^T} \subseteq U_{\sigma^{T'}}$ or $U_{\sigma^T} \supseteq U_{\sigma^{T'}}$ and elements in U_{σ^T} and $U_{\sigma^{T'}}$ are of the same type.
- **semantics domain altering** for σ^T if $U_{\sigma^T} \neq U_{\sigma^{T'}}$.
- **semantics bijective** under $(\sigma^T, SC^{Tr}_{\sigma^T})$ iff it is strong under $(\sigma^T, SC^{Tr}_{\sigma^T})$ and every semantics casting function in $SC^{Tr}_{\sigma^T}$ is bijective.
- **faithful** under $(\sigma^T, SC^{Tr}_{\sigma^T})$ iff it is semantics bijective and the semantics casting functions are removals or additions.
- **exact** under $(\sigma^T, SC^{Tr}_{\sigma^T})$ iff it is semantics bijective and the semantics casting functions are identities.

Please note that [42] also introduces the weak versions of exactness and faithfulness. In this approach for a given translation there is a certain predefined collection of extensions referred to as the remainder sets. Such sets are always removed from the collection of extensions a given semantics produces for a framework and whatever is left has to conform to the normal exact and faithful definitions. Clearly, this manipulation can no longer be handled by the singular semantics casting functions and qualifies for collective approach. Therefore, we will not focus on these properties further.

3.2.4 Computational Properties

Last, but not least, we have the group of properties concerning how difficult and expensive to actually compute the translation for a given framework. Although this group is important from the practical perspective, many properties here will be only sketched and analyzing the complexity of translations will be left mostly for future work.

We have already mentioned that the faithfulness, modularity and polynomiality are among the most studied properties of translations. While the first one qualified as a semantical property, the other two fall into complexity category. *Modularity* [55] tells us whether the target framework can be obtained by joining separately translated parts of the source framework. While normally this property allows more efficient translation of a framework that had to undergo some update (i.e. addition of new elements), it also gives us the opportunity to create parallel algorithms for a given translation. In case of AFs, we "joined" the frameworks back together through the union of their corresponding parts, e.g. two AFs (A_1, R_1) and (A_2, R_2) would produce $(A_1 \cup A_2, R_1 \cup R_2)$. However, when we move to the structures permitting group relations, this is clearly not the only way we can proceed. Consider two AFNs $(\{a, b, c\}, \emptyset, \{(\{a\}, c)\})$ and $(\{a, b, c\}, \emptyset, \{(\{b\}, c)\})$. They can be the parts of a framework $(\{a, b, c\}, \emptyset, \{(\{a\}, c), (\{b\}, c)\})$ if we follow the union–based approach. The support given to c is read as "both a and b need to be present to accept a". Nevertheless, the result $(\{a, b, c\}, \emptyset, \{(\{a, b\}, c)\})$ also makes a lot of sense. In this case, either a or b needs to be present to assume c, and splitting this framework into our two initial ones can be used in analyzing powerful sequences for c. Consequently, a given argumentation framework can be separated and put back together in various ways. In our work we will focus on the union–based modularity that we described in the very beginning. For most of the argumentation frameworks, the union of two structures will be just the framework obtained by joining their respective elements. However, since ADFs use acceptance conditions, not separate sets of relations, we need to describe how two conditions can be made into one. The problem has been previously studied in [49] and two operators have been proposed, one that can be seen as "conjunctive" and the other as "disjunctive".

Definition 3.10. Let $D_1 = (A_1, L_1, C_1)$ and $D_2 = (A_2, L_2, C_2)$ be two propositional ADFs, i.e. $C_1 = \{\varphi_1^a\}_{a \in A_1}$ and $C_2 = \{\varphi_2^a\}_{a \in A_2}$, where φ_i^a is a propositional formula over a subset of A_i. Let $A = A_1 \cup A_2$ and $L = L_1 \cup L_2$. We define $D_1 \otimes D_2 = (A, L, C^{\otimes})$ and $D_1 \oplus D_2 = (A, L, C^{\oplus})$, where:

$$C^{\otimes} = \{C_1 \otimes_s C_2\}_{s \in A} \text{ and } C_1 \otimes_s C_2 = \begin{cases} \varphi_s^1 \land \varphi_s^2 & \text{if } s \in A_1 \cap A_2 \\ \varphi_s^1 & \text{if } s \in A_1 \setminus A_2 \\ \varphi_s^2 & \text{otherwise} \end{cases}$$

$$C^{\oplus} = \{C_1 \oplus_s C_2\}_{s \in A} \text{ and } C_1 \oplus_s C_2 = \begin{cases} \varphi_s^1 \lor \varphi_s^2 & \text{if } s \in A_1 \cap A_2 \\ \varphi_s^1 & \text{if } s \in A_1 \setminus A_2 \\ \varphi_s^2 & \text{otherwise} \end{cases}$$

We can observe that if $A_1 \cap A_2 = \emptyset$, then $D_1 \otimes D_2 = D_1 \oplus D_2$. Which operator will be used will depend on the translation in question.

Definition 3.11. Let $D_1 = (A_1, L_1, C_1)$ and $D_2 = (A_2, L_2, C_2)$ be two functional ADFs, i.e. $C_1 = \{C_1^a\}_{a \in A_1}$ and $C_2 = \{C_2^a\}_{a \in A_2}$, where C_i^a is a total function from the sets of parents of a in framework D_i to $\{in, out\}$. Let $A = A_1 \cup A_2$ and $L = L_1 \cup L_2$. The joint parent set $jpar(s)$ of an argument $s \in A$ is defined as:

$$jpar(s) = \begin{cases} par_1(s) \cup par_2(s) & \text{if } s \in A_1 \cap A_2 \\ par_1(s) & \text{if } s \in A_1 \setminus A_2 \\ par_2(s) & \text{otherwise} \end{cases}$$

We define $D_1 \otimes D_2 = (A, L, C^{\otimes})$ and $D_1 \oplus D_2 = (A, L, C^{\oplus})$, where $C^{\otimes} = \{C_1 \otimes_s C_2\}_{s \in A}$, $C^{\oplus} = \{C_1 \oplus_s C_2\}_{s \in A}$ and for a given subset of joint parents $X \subseteq jpar(s)$, the condition $C_1 \otimes_s C_2$ ($C_1 \otimes_s C_2$ respectively) is defined as:

$$C_1 \otimes_s C_2(X) = \begin{cases} in & \text{if } C_s^i(X \cap par_i(s)) = in \text{ for all } i \in \{1,2\} \text{ and } s \in A_1 \cap A_2 \\ out & \text{if } C_s^i(X \cap par_i(s)) = out \text{ for any } i \in \{1,2\} \text{ and } s \in A_1 \cap A_2 \\ C_s^1(X) & \text{if } s \in A_1 \setminus A_2 \\ C_s^2(X) & \text{otherwise} \end{cases}$$

$$C_1 \oplus_s C_2(X) = \begin{cases} out & \text{if } C_s^i(X \cap par_i(s)) = out \text{ for all } i \in \{1,2\} \text{ and } s \in A_1 \cap A_2 \\ in & \text{if } C_s^i(X \cap par_i(s)) = in \text{ for any } i \in \{1,2\} \text{ and } s \in A_1 \cap A_2 \\ C_s^1(X) & \text{if } s \in A_1 \setminus A_2 \\ C_s^2(X) & \text{otherwise} \end{cases}$$

Polynomiality can stand for two things; running time of the translation or the size of the produced framework[12]. To distinguish between the two, following [57] we will refer to the other property as *polysize*. Although the size of the framework is usually understood as the number of arguments, please note that in certain cases this will not be an adequate view. This is particularly visible in the translations from frameworks with support to SETAFs (see Sections 10.2 and 11.2). While the increase in the number of arguments is at worst linear, the amount of created group attacks can be overwhelming. For example, if we did not assume minimality in the defender translation from AFN to SETAF (Translation 65), the amount of added group attacks form the coherent would be bounded by 2^n with n being the number of source arguments. While assuming minimality reduces this number (it is no more than $\binom{n}{\lfloor n/2 \rfloor}$ by Sperner's theorem), it naturally increases the running time of the translation. As the number of attacks in the Dung's framework is at most quadratic w.r.t. the number of arguments, it is not really necessary to differentiate between the polysize increase of attacks and the polysize increase of arguments. A SETAF in which every (nonempty) set of arguments attacks every argument has $(2^n - 1) * n$ attacks and the need to distinguish between the two elements is more apparent. Consequently, on certain

[12]Please note that [42] also introduces efficient translations which run in logarithmic time.

occasions it needs to be stated with respect to what the polysize property does or does not hold.

Finally, we will analyze what does the translation actually *need* in order to proceed: is it just the structure of the framework that is required, or do we need to take into account some semantical aspects? Answering this question is of practical value, particularly if we use translations in order to reuse the solver of the target structure for working with the source one. In the first case we just need to program a way to take framework as an input, basically a parsing method. Otherwise, we might need to do some pre–solving of the structure, which can require some effort. Please note that the border between structural and semantical properties is not very clear. For example, shifting an AF (A, R) into an AFN (Translation 21) requires only the addition of an empty support relation $N = \emptyset$. The produced triple (A, R, N) is a fully functioning AFN that can be used to produce AF extensions. This is clearly a purely "structural" translation. When we transform a SETAF into an AF, then a set of arguments that carries out some attack ends up added as a new argument in the resulting AF (Translation 25). This still is a structural approach. Let us now assume we want to translate a support framework such as BAF into an AF [30]. The AF arguments are sets of source arguments just like in the SETAF–AF translation; however, while in SETAF they represented groups that carried out attacks, in this case they correspond to groups that are connected by support. Now we are not so certain anymore whether this can really count as a structural approach. To create the support sets we only "scan" through the relation, but the required analysis puts it closer to e.g. conflict–free semantics rather than just adding the attack sets already defined in the framework. Thus, this is more of a "semi–structural" approach. Finally, let us now assume that the support framework in question is actually an AFN or EAS (see Translations 61 and 71). In this case the we cannot create arguments from just arbitrary support sets. We need to make sure that the support connections are *valid*, i.e. if they are acyclic or rooted in evidence. This basically brings us to creating coherent or self–supporting sets. This is now clearly a "semantical" translation and requires knowledge about semantical notions in the framework, such as what sort of support is valid, in order to proceed. However, please note that what we understand by a semantical translation is not as extreme as in [40], where the extensions of a given semantics, such as grounded, need to be computed for the transformation. In our approach, the semantical aspects rarely go beyond validity of framework components (i.e. sequences and acyclic evaluations) and indirect conflicts. Thus, we still stay in the "lower class" of semantical transformations.

Definition 3.12. Let T, T' be two distinct types of abstract argumentation frameworks, D^T, $D^{T'}$ collections of the frameworks of the respective types and $Tr : D^T \to D^{T'}$ a framework translation. By fs we will understand the type of framework size. We say that Tr is:

- **modular** if $T, T' \neq ADF$ and for every $X, X' \in D^T$ s.t. $X \cup X' \in D^T$, it holds that $Tr(X) \cup Tr(X') = Tr(X \cup X')$.
- **δ–modular**, where $\delta \in \{\otimes, \oplus\}$, if:

- $T \neq ADF, T' = ADF$, and for every $X, X' \in D^T$ s.t. $X \cup X' \in D^T$, it holds that $Tr(X) \,\delta\, Tr(X') = Tr(X \cup X')$, or
- $T, T' = ADF$, and for every $X, X' \in D^T$ s.t. $X \,\delta\, X' \in D^T$, it holds that $Tr(X) \,\delta\, Tr(X') = Tr(X \,\delta\, X')$, or
- $T = ADF, T' \neq ADF$, and for every $X, X' \in D^T$ s.t. $X \,\delta\, X' \in D^T$, it holds that $Tr(X) \cup Tr(X') = Tr(X \,\delta\, X')$.

- **polynomial** (polytime) w.r.t. fs if the running time of Tr is polynomial w.r.t. fs size of the frameworks.
- **polysize** w.r.t. fs if for every $X \in D^T$, the fs size of $Tr(X)$ is polynomially large w.r.t. the fs size of X.
- **semantical** if translating the framework strongly depends on the semantical notions of the source framework or target framework.
- **structural** if translating the framework does not depend on any semantical notions of the source or target framework.
- **semi–structural** if translating the framework can depend on or be similar to certain weak semantical notions of the source or target framework.

Please note that when necessary, we might introduce more specialized notions of modularity. Nevertheless, they will still follow the idea of splitting and composing the frameworks under a translation. We also stress the fact that we speak in terms of modularity only w.r.t. those framework for which their union is still in the assumed framework domain. This will be particularly important in the case of source–subclass translations.

3.3 Classification of Translations

In the previous section we have described various properties that a translation can have. However, we have not said much about the methods for creating translations. The translations in this work are not created at random; they follow certain concepts and ideas which have resulted from the current approaches in the field and the analysis of the framework types in question. In this section we will distinguish four main types of translations - basic, coalition, attack propagation and defender, though please note that they of course do not account for all possible approaches. The separation is also rough at best; we will later observe that there are translations that can be assigned more than one type.

The basic approach will concern the simple transformations, which are quite often just generalizations and concern moving between comparable or from less to more structurally complicated frameworks. On average they do not require auxiliary arguments and do not change the domains. For example, in order to transform an AF into an AFN, we only need to include the empty set for the support relation (see Translation 21). These are also the most desirable approaches from the practical point of view, since they will be in general structural, computationally cheap and at least faithful.

The coalition, attack propagation and defender approaches are somewhat opposite to basic and are used when the target structure cannot represent certain elements of the source one. In the coalition case, we assume a particular structure of the arguments in the target framework (very often just the sets of source arguments) and the "not handled" elements are hidden away in these arguments. For example, AFs do not have a support relation in their structure. Therefore, when transforming an EAS, we create self–supporting sets and feed them to a Dung's framework as arguments, thus hiding the evidential support from the target structure (see Translation 71). The same is done to the group attack present in EASs. The coalition approach is close to meta–level argumentation [64] and has the problem of assuming that the arguments we are dealing with possess some structure. Since, by principle, no abstract argumentation framework stores such data, one has to bear in mind that it is necessary to remember such content in some auxiliary structure. Please note that although the approach is named after the research in [28, 29], our coalition construction is more relaxed and closer to [30, 69, 73] rather than the original.

In the attack propagation and defender methods, instead of hiding problematic relations, we try to transform them into something that the target framework can handle. For example, in the first case, we simulate the effect of support by combinations of attacks (e.g. in Translation 73), while in the other we turn a supporter of an argument into its defender from an auxiliary attacker (see Translation 75). The attack propagation idea has been taken from [30], though as we will see in Sections 10.2 and 11.2, the original research had certain technical deficiencies and did not take into account the various design choices of the frameworks in question, in particular the problem of handling the support cycles. The defender method was inspired by the discussion regarding the difference between support and defense done in [28], though recently it has separately appeared in [31]. Although we still believe that there are types of support which cannot be handled by defense, in this work we will show that with the use of auxiliary arguments, the necessary and evidential types are not among them.

What is also worth mentioning is that although this classification was created with translations from and to framework with support in mind, similar behaviors can be found in conversions between attack–based structures. Semantics of e.g. AFRAs take not only direct, but also indirect defeats into account. Consequently, this conflict needs to be propagated when we attempt to transform AFRA into an AF (see Section 7.1). Similarly, the flattening translations, though performed only for attack–based frameworks [18, 64], turn out to follow a similar principle as the defender method. The types of conflicts that cannot be handled by the target framework are changed into arguments, which are later defended by the arguments carrying them out. Therefore, we will classify those two approach as one.

The most important thing to note about the defender and propagation approaches, when used in bipolar frameworks, is the fact that we assume that the acceptability of an argument depends on the presence of its supporters. Since not all types of support have this property, one has to make this assumption explicit. When an argument cannot be accepted without the sufficient support, the so–called "cutting off the support" qualifies as an attack suitable for defense. This means that we can defeat an argument not just by directly attacking it, but also by attacking its supporters – since without them the argument will not be accepted, defeating supporters leads to defeating the argument in an indirect manner. What attack propagation does is adding such indirect attacks to direct ones; the consequences of support are then simulated by attacks and it can be safely removed from the framework. The downside of this approach is that it detaches supporters from supported arguments and as a side effect, this method is best used for semantics that are at least complete.

While the attack propagation focuses on the "if supporters are not accepted, the argument is not accepted" aspect of the necessary and evidential supports, the defender approach is a slightly more positive approach. What ties arguments together in any type of argumentation is defense; an argument needs it defenders to be present in an admissible extension. Since we have the assumption that it also needs its supporters, transforming support into defense is a natural idea. In the defender approach, we introduce additional attackers claiming that the given arguments are unsupported; they are then in turn attacked by the supporters and thus defense simulates the desired behavior. While the method can be used for semantics that are at least admissible, it does introduce new arguments into the framework. Please note that in a certain sense, the defender approach borders meta–argumentation. Although we do not really "store" the meaning of the auxiliary arguments and we do not need it to retrieve the desired extensions unlike in e.g. the coalition approach, what they do is make statements about the support status of other arguments. Without being aware of this, the method behind the translation would be harder to understand.

What has to be clearly stated is that in the coalition, attack propagation and defender approaches the validity of support links makes a huge difference. For example, in necessary and evidential cases, self–supporters are not valid arguments and they will not be represented by any coalition. Since the attacks carried out by them are ignored, or depending on how we want to view it, automatically defended from, we need to flush them out from the attack propagation approach. Although we can leave them just as arguments that will never be defended from the auxiliary attackers, a given argument has to be defended not only by direct, but also indirect supporters – basically speaking, the whole evidential or powerful sequence. This stems from the fact that not every "support path" is valid and if we are not careful, we might end up transforming a support cycle into self–defense. While the AFN, EAS and ADF semantics would handle the cycles, self–defense is not an issue in majority of the semantics in any framework. Consequently, unlike in the basic translation, these approaches will be semantical and relatively expensive in frameworks with support. We close the section with charts showing where given translations types occur and refer the reader to the provided sections for further details and examples. Please note that by the notion of a chained translation we will understand a translation between two frameworks that uses another framework as an intermediate step.

Table 5: Translation Classification

	AF	SETAF	AFRA	EAF	EAFC	BAF	AFN	EAS	ADF
AF	-	Bas	Bas	Bas	Ch	Bas	Bas	Bas	Bas
SETAF	Bas- Cl, Bas-Def	-	Ch	Ch	Ch	Ch	Bas-Cl	Bas	Bas
AFRA	Bas-AP, Bas-Def	Ch	-	Ch	Ch	Bas	Bas	Ch	Ch
EAF	Bas-Def	Bas	Bas	-	Bas	Ch	Bas	Ch	Bas
EAFC	Bas-Def	Bas	Ch	Bas-Def	-	Ch	Bas	Ch	Bas
BAF	Bas-AP, Bas-AP-Def Bas-Cl	Ch	Ch	Ch	Ch	-	Bas	Bas	Ch
AFN	Cl, Bas	AP, Def	Ch	Ch	Ch	Bas	-	Bas	Bas
EAS	Cl	AP, Def, Cl	Ch	Ch	Ch	Bas	Bas, Bas-Cl	-	Bas
ADF	Cl, Cl-Def	Cl-Def, AP	Ch	Ch	Ch	Ch	?	?	-

Legend: Bas- basic translation, Cl- coalition translation, AP– attack propagation translation, Def- defender translation, Ch- chained translation, ? - unclassified translation

4 Framework Normal Forms & Subclasses

When analyzing argumentation frameworks, one can sometimes identify their subclasses which have some desirable properties, such as agreement of semantics or lower computational complexity. The majority of this research concerns the Dung's framework [10, 11, 34, 36], though certain subclasses have also been identified for EAFs (see Section 2.1.4) and for ADFs (Sections 2.3 and 2.3.8). However, the normal forms of a framework are something more than subclasses. Again, their purpose is to have frameworks exhibiting properties desirable from the point of view of e.g. computational complexity or seman-

Table 6: Translation Chart of Argumentation Frameworks

<table>
<tr><th></th><th>AF</th><th>SETAF</th><th>AFRA</th><th>EAF</th><th>EAFC</th><th>BAF</th><th>AFN</th><th>EAS</th><th>ADF</th></tr>
<tr><td>AF</td><td>x</td><td>17</td><td>18</td><td>19</td><td>Sec. 5.3</td><td>20</td><td>21</td><td>22</td><td>23</td></tr>
<tr><td>SETAF</td><td>25, 26, 27</td><td>x</td><td colspan="3">Sec. 6.6</td><td>Sec. 6.2</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>AFRA</td><td>32, 33</td><td>Sec. 7.6</td><td>x</td><td colspan="2">Sec. 7.2</td><td>34</td><td>35</td><td>Sec. 7.6</td><td>Sec. 7.5</td></tr>
<tr><td>EAF</td><td>38</td><td>40</td><td>42</td><td>x</td><td>36</td><td>Sec. 8.7</td><td>43, 44</td><td>Sec. 8.7</td><td>47, 48</td></tr>
<tr><td>EAFC</td><td>39</td><td>41</td><td>Sec. 8.4</td><td>37</td><td>x</td><td>Sec. 8.7</td><td>45, 46</td><td>Sec. 8.7</td><td>49, 50</td></tr>
<tr><td>BAF</td><td>51,52, 53, 54, 55, 57, 58</td><td colspan="4">Sec. 9.5</td><td>x</td><td>59</td><td>60</td><td>Sec. 9.4</td></tr>
<tr><td>AFN</td><td>61, 62</td><td>63, 65, 66</td><td>Sec. 10.7</td><td colspan="2">Sec. 10.3</td><td>67</td><td>x</td><td>68</td><td>69, 70</td></tr>
<tr><td>EAS</td><td>71</td><td>72, 73, 75, 76</td><td colspan="3">Sec. 11.6</td><td>77</td><td>78,79</td><td>x</td><td>80, 81</td></tr>
<tr><td>ADF</td><td>82, 83, 85, 86</td><td>87, 88</td><td colspan="4">Sec. 12.4</td><td>89</td><td>Sec. 12.4</td><td>x</td></tr>
</table>

tics. Consequently, the collections of normal forms of frameworks can form subclasses of their own. The difference lies in the fact that while a given framework can belong to a subclass or not, a normal form is something we can transform the structure into. We create a new framework which might look differently, but is in a sense equivalent to the initial one – in a similar manner one generates conjunctive or disjunctive normal forms of propositional formulas. To the best of our knowledge, only the work in [35] also studies the normal forms of frameworks, though only in the context of AFs. Moreover, the results are relevant for computational and not translation purposes.

In what follows we will discuss minimal, consistent, and various validity normal forms (or subclasses, when applicable). Moreover, we will also introduce the cleansed form for ADFs. Please note that although we follow the "to the subclass you belong, to normal form you transform" distinction, in some cases we will not provide a translation to a normal form. Depending on whether we believe that such a translation exists, certain notions we might count as normal forms and certain as subclasses. In such ambiguous cases we allow the reader to refer to them or reclassify them as he or she pleases.

In principle, we have normal forms that add (replace) or remove certain elements in order to simplify the framework structure or computation of the semantics. While replacement can create a structure that does not exactly qualify as "bigger" or "smaller" than the original, removal produces a so–called subframework:

Definition 4.1. Let $F = (A, R)$ be a Dung's framework. We say that $G_F = (A', R')$ is a **subframework** of F, denoted $G_F \sqsubseteq F$, if $A' \subseteq A$ and $R' \subseteq R \cap (A' \times A')$. G_F is a **full subframework** induced by $A' \subseteq A$ if $R' = R \cap (A' \times A')$.

The notion of a (full) subframework can naturally be extended to other argumentation frameworks. We can observe that if we are dealing with group relations, then following this definition, only those relations that are carried out by sets fully contained in the new argument set may carry over to the subframework. However, in the case of support in AFNs, another way of obtaining N' can be considered more intuitive. Given that c is supported by the set $\{a, b\}$, which is read as "a or b needs to be accepted in order to assume c", a subframework not containing b would completely remove this support. However, one would rather expect the support to be trimmed down to "a is necessary for c". This brings us to the notion of the trimmed subframework:

Definition 4.2. Let $FN = (A, R, N)$ be an AFN. We say that $G_{FN} = (A', R', N')$ is a **trimmed subframework** of FN, denoted $G_{FN} \overset{T}{\sqsubseteq} FN$, if $A' \subseteq A$, $R' \subseteq R \cap (A' \times A')$, and $N' \subseteq \{(C', a) \mid a \in A', C' \neq \emptyset, \exists (C, a) \in N \wedge C' \subseteq C \cap A'\}$. G_{FN} is a **full trimmed subframework** induced by $A' \subseteq A$ if $R' = R \cap (A' \times A')$ and $N' = \{(C', a) \mid a \in A', C' \neq \emptyset, \exists (C, a) \in N \wedge C' = C \cap A'\}$.

The subframework notion is related to the notion of a reduct in ADFs. Let us recall the original definition from Section 2.3.4:

Definition 2.135. Let $D = (A, L, C)$ be an ADF and $E \subseteq A$ a set of arguments. A **reduct** of D w.r.t. E is a framework $D^E = (E, L^E, C^E)$, where $L^E = L \cap (E \times E)$ and for $e \in E$ we set $C^E_e = \varphi_e[b/\mathbf{f} : b \notin E]$.

Please note that this definition uses the propositional representation of ADFs. Thus, we will also define an equivalent, functional version (see Definition 2.113 of ADFs):

Definition 4.3. Let $D = (A, L, C)$ be an ADF and $E \subseteq A$ a set of arguments. The **reduct** of D w.r.t. E is a framework $D^E = (E, L^E, C^E)$, where $L^E = L \cap (E \times E)$ and for $e \in E$, $C_e^E = C_e \cap (2^E \times \{in, out\})$.

In other words, we just remove those mappings which contain arguments from $A \setminus E$ from a given acceptance condition. Thus, one can see an ADF subframework as a framework containing some of the arguments and a subset of the assignments of a given acceptance condition s.t. the subset is still the powerset of the arguments appearing in the new condition.

It is easy to see that in many ways, the notion of a subframework is a broad one, and not all subframeworks of a given framework are interesting for our purposes. By restricting this concept, we will obtain most of our normal forms, such as minimal, weakly valid, relation valid and cleansed. Let us now start with the first one.

4.1 Minimal and Redundancy–Free Forms

The *minimal normal form* of a framework is relevant to SETAFs, EAFCs, AFNs and EASs – basically speaking, any frameworks that permit group relations. Similar in its purpose is the redundancy–free form for ADFs. When faced with non binary relations, in certain situations we can view some information as excessive. For example, when we know that arguments a and b jointly attack c, the group attack from $\{a, b, d\}$ on c is, in this case, not relevant from the point of view of the standard semantics – defending from the first would imply defense from the latter. In some cases, removing such redundancies might even be necessary [48]. In what follows we will define the minimal normal form for the listed frameworks, show that they in fact preserve the behavior of the semantics, and put them into our translation system.

Please note that what we will describe is only one of the ways minimality can be understood. Our approach is purely structural based and "local", i.e. imposes minimality on the incoming relations. However, one can also consider imposing minimality on support paths. Consider an AFN $(\{a, b, c\}, \emptyset, \{(\{a\}, b), (\{a\}, c), (\{b\}, c)\})$, in which a supports b, and c requires both a and b to hold. However, we can observe that removing the $(\{a\}, c)$ support from the framework would not change much – after all, a will always be present in an extension containing c due to the relation of both arguments to b. Thus, in case one argument supports both directly and indirectly another argument, we can decide to remove the first one. Nevertheless, for now we will consider only the local minimality and leave this approach for our future work.

4.1.1 SETAF Minimal Form

We will start with the minimal normal form of SETAFs. Basically speaking, we remove non–minimal attacks from the framework. We can observe that this type of a structural change does not affect the semantics:

Definition 4.4. Let $SF = (A, R)$ be a SETAF. The subframework $SF^{min} = (A, R')$ is the **minimal form** of SF, denoted $SF^{min} \overset{min}{\sqsubseteq} SF$, iff $R' \subseteq R$ consists of all and only elements (T, a) in R s.t. $\nexists T' \subset T, (T', a) \in R$.

Theorem 4.5. *Let $SF = (A, R)$ be a SETAF and $SF^{min} = (A, R')$ its minimal form. A set of arguments $E \subseteq A$ is a σ–extension of SF, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, iff it is a σ–extension of SF^{min}.*

By putting the normal form transformation into our system, we obtain the following notions:

Translation 1. *Let Fr^{SETAF} be the collection of all SETAFs and Min^{SETAF} the collection of those SETAFs that are in minimal normal form, both based on argument domain $\mathcal{U}$. The minimal form translation $min\text{-}Tr^{SETAF} : Fr^{SETAF} \rightarrow Min^{SETAF}$ is defined as $min\text{-}Tr^{SETAF}((A, R)) = (A, R')$, where $R' \subseteq R$ consists of all and only elements (T, a) in R s.t. $\nexists T' \subset T, (T', a) \in R$.*

Redefinition of Theorem 4.5: *Let $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC_{σ}^{Tr} the identity casting functions for σ. The translation $min\text{-}Tr^{SETAF}$ is strong and semantics bijective under $(\sigma, SC_{\sigma}^{Tr})$.*
Analysis of Translation 1: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and identity casting functions, the translation $min\text{-}Tr^{SETAF}$ is:

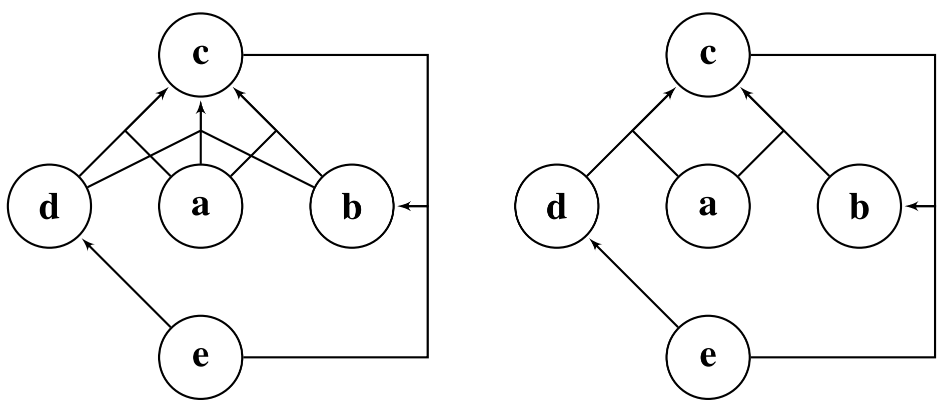
- full, target–subclass and overlapping
- argument domain preserving and attack relation removing
- generic, semantics domain preserving and exact
- structural

The translation $min\text{-}Tr^{SETAF}$ is not modular.

Explanation. Since every SETAF can be transformed to the minimal normal form and the frameworks that are already in that form are only a subclass of Fr^{SETAF}, the translation is classified as full and target–subclass. As two different SETAFs can have the same minimal normal form, for example both $(\{a, b, c\}, \{(\{a\}, c), (\{a, b\}, c)\})$ and $(\{a, b, c\}, \{(\{a\}, c), (\{a, b\}, c), (\{a, b, c\}, c)\})$ are mapped to $(\{a, b, c\}, \{(\{a\}, c)\})$, our assignment is overlapping. We can also easily observe that the translation is removing certain relations; that was after all the whole point of this method. The domain preserving and generic properties are a result of the definition of the translation and Theorem 4.5. The strength and exactness of the translation follow easily from the redefinition of this theorem as well. The approach is also clearly structural; the modifications done to the attack relation do not take any semantical notions into account.

The loss of modularity comes from the fact that we look for the "minimal" attacks in the set. Consequently, we can split a framework in a way that the minimal attacks of its subframeworks will be in fact comparable. Thus, the union of their translations might be in fact outside Min^{SETAF}. Consider the framework $(\{a,b,c\},\{(\{a\},c),(\{a,b\},c)\})$ and its subframeworks $(\{a,b,c\},\{(\{a\},c)\})$ and $(\{a,b,c\},\{\{a,b\},c)\})$. They are already in a minimal form, however, their union is not, and it does not coincide with the minimal form $(\{a,b,c\},\{(\{a\},c)\})$ of the original structure. ■

Example 42. Let $SF = (\{a,b,c,d,e\}, \{(\{a,b\},c), (\{a,d\},c), (\{a,b,d\},c), (\{e\},d), (\{c,e\},b)\})$ be a SETAF. Its minimal form is $SF^{min} = (\{a,b,c,d,e\}, \{(\{a,b\},c), (\{a,d\},c), (\{e\},d), (\{c,e\},b)\})$. Both are depicted in Figure 30. We can observe that in SF, the set $\{c,e\}$ has the power to defend c; as the set attacks both d and b, it prevents all three attacks on c. Since attacks formed by the subsets of $\{a,b,d\}$ are taken care of, so is the one carried out by $\{a,b,d\}$ itself and removing $(\{a,b,d\},c)$ from the attack relation does not change the fact that $\{c,e\}$ is admissible. If we considered a modification of SF where the $(\{e\},d)$ attack is not present, then $\{c,e\}$ would not be able to defend c from the $(\{a,d\},c)$ and $(\{a,b,d\},c)$ – again, removing the latter would still not change the fact that c is not acceptable.



(a) Sample SETAF SF (b) Minimal form of SF

Figure 30: Sample SETAF and its minimal normal form framework

4.1.2 EAFC Minimal Form

Another framework for which we can introduce the minimal form is EAFC. Although the attacks as such are binary, the defense attacks can be carried out by groups of arguments. Thus, similarly as in SETAFs, we can consider removing some of the redundant sets.

Definition 4.6. Let $EFC = (A, R, D)$ be an EAFC. The subframework $EFC^{min} = (A, R, D')$ is the **minimal form** of EFC, denoted $EFC^{min} \overset{min}{\sqsubseteq} EFC$, iff $D' \subseteq D$ consists of all and only elements (T, a) in D s.t. $\nexists T' \subset T, (T', a) \in D$.

Theorem 4.7. *Let $EFC = (A, R, D)$ be an EAFC and $EFC^{min} = (A, R, D)$ its minimal form. A set of arguments $E \subseteq A$ is a σ–extension of EFC, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, iff it is a σ–extension of EFC^{min}.*

We can now put our results into the system. Analysis similar to the one given in the SETAF minimal form can be carried out and therefore we will omit further explanations.

Translation 2. *Let Fr^{EAFC} be the collection of all EAFCs and Min^{EAFC} the collection of those EAFCs that are in minimal normal form, both based on argument domain $\mathcal{U}$. The minimal form translation $min\text{-}Tr^{EAFC} : Fr^{EAFC} \rightarrow Min^{EAFC}$ is defined as $min\text{-}Tr^{EAFC}((A, R, D)) = (A, R, D')$, where $D' \subseteq D$ consists of all and only elements (T, a) in D s.t. $\nexists T' \subset T, (T', a) \in D$.*

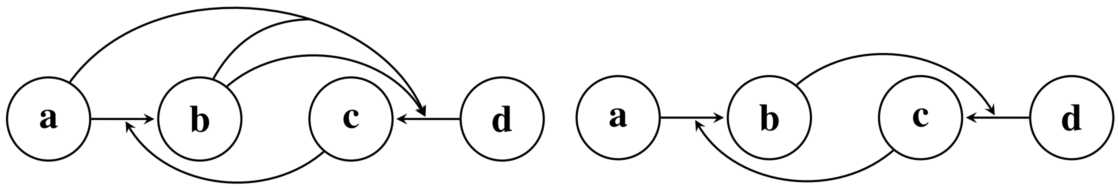
Redefinition of Theorem 4.7: *Let $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the identity casting functions for σ. The translation $min\text{-}Tr^{EAFC}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*

Analysis of Translation 2: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and identity casting functions, the translation $min\text{-}Tr^{EAFC}$ is:

- full, target–subclass and overlapping
- argument domain preserving and defense attack relation removing
- generic, semantics domain preserving and exact
- structural

The translation $min\text{-}Tr^{EAFC}$ is not modular.

Example 43. Let us consider the EAFC $EFC = (\{a, b, c, d\}, \{(a, b), (d, c)\}, \{(\{b\}, (d, c)), (\{a, b\}, (d, c)), (\{c\}, (a, b))\})$. Its minimal form is $EFC^{min} = (\{a, b, c, d\}, \{(a, b), (d, c)\}, \{(\{b\}, (d, c)), (\{c\}, (a, b))\})$. Both are depicted in Figure 31. We can observe that a defeats$_{\{a,d\}}$ b in EFC^{min} and that this defeat has a reinstatement set $\{(a, b), (d, c)\}$ on $\{a, d\}$. In other words, the defense attack on (a, b) carried out by $\{c\}$ is handled by the (d, c) attack, and the defense attack on (d, c) by $\{b\}$ is nullified by the (a, b) attack. In the same way, a defeats$_{\{a,d\}}$ b in EFC and the reinstatement set for this defeat is still $\{(a, b), (d, c)\}$; in this case, (d, c) deals both with $\{b\}$ and $\{a, b\}$. Moreover, $\{a, b, c, d\}$ is conflict–free in both frameworks; removing the $(\{a, b\}, (d, c))$ defense attack does not change that as long as $(\{b\}, (d, c))$ is still present.



(a) Sample EAFC EFC (b) Minimal form of EFC

Figure 31: Sample EAFC and its minimal normal form framework

4.1.3 AFN Minimal Form

The AFN minimal form we have previously introduced in [77, 78][13]. Although AFNs permit only binary attack, we can filter out unnecessary support relations:

Definition 4.8. Let $FN = (A, R, N)$ be an AFN. The subframework $FN^{min} = (A, R, N')$ is the **minimal form** of FN, denoted $FN^{min} \overset{min}{\sqsubseteq} FN$, iff $N' \subseteq N$ consists of all and only elements (T, a) in N s.t. $\nexists T' \subset T, (T', a) \in N$.

Theorem 4.9. *Let $FN = (A, R, N)$ be an AFN and $FN^{min} = (A, R, N')$ its minimal form. A set of arguments $E \subseteq A$ is a σ–extension in FN where $\sigma \in \{$conflict–free, coherent, admissible, preferred, complete, grounded, stable$\}$ iff it is a σ–extension in FN^{min}.*

We can now put the minimal normal form translation into our system. Its properties and their explanations resemble the ones given in the SETAF case and thus we will omit that part of the analysis.

Translation 3. *Let Fr^{AFN} be the collection of all AFNs and Min^{AFN} the collection of those AFNs that are in minimal normal form, both based on argument domain $\mathcal{U}$. The minimal form translation $min\text{-}Tr^{AFN} : Fr^{AFN} \rightarrow Min^{AFN}$ is defined as $min\text{-}Tr^{AFN}((A, R, N)) = (A, R, N')$, where $N' \subseteq N'$ consists of all and only elements (T, a) in N s.t. $\nexists T' \subset T, (T', a) \in N$.*

Redefinition of Theorem 4.9: *Let $\sigma \in \{$conflict–free, coherent, admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the identity casting functions for σ. The translation $min\text{-}Tr^{AFN}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*

Analysis of Translation 3: Under the conflict–free, coherent, admissible, preferred, complete, grounded and stable semantics and identity casting functions, the translation $min\text{-}Tr^{AFN}$ is:

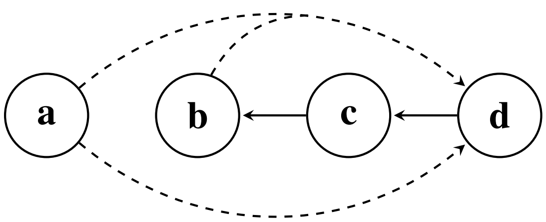
- full, target–subclass and overlapping

[13] Please note that conflict–free and coherent semantics were not explicitly stated in Theorem 4.7 in [78], but were shown in the proofs [77].

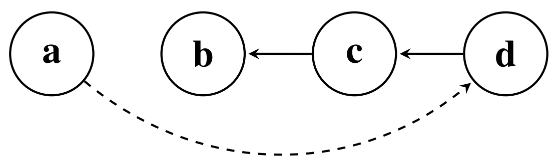
- argument domain preserving, attack relation preserving and support relation removing
- generic, semantics domain preserving and exact
- structural

The translation $min\text{-}Tr^{AFN}$ is not modular.

Example 44. Let $FN = (\{a, b, c, d\}, \{(d, c), (c, b)\}, \{(\{a\}, d), (\{a, b\}, d)\})$ be an AFN; its minimal form is $FN^{min} = (\{a, b, c, d\}, \{(d, c), (c, b)\}, \{(\{a\}, d)\})$. Both are depicted in Figure 32. We can observe that c attacks one of the sets supporting d; however, as the coherent set $\{a, d\}$ still remains unattacked, argument c cannot defend itself from d. Removing the non–minimal support $(\{a, b\}, d)$ does not change this fact. If we were to include an attack (c, a) into the set, then both coherent sets $\{a, d\}$ and $\{a, b, d\}$ would be attacked and thus the redundant support does not provide any alternative evaluation path that would avoid the (c, a) attack.



(a) Sample AFN FN



(b) Minimal form of FN

Figure 32: Sample AFN and its minimal normal form framework

4.1.4 EAS Minimal Form

Just like in the case of AFNs, the EAS minimal normal form has already been introduced in [77, 78][14]. Again, we will reformulate the results so that it is clear how this transformation fits into our system. In this framework we will have to deal with minimality both of the attack and support relation:

Definition 4.10. Let $ES = (A, R, E)$ be an EAS. The subframework $ES^{min} = (A, R', E')$ is the **minimal form** of ES, denoted $ES^{min} \overset{min}{\sqsubseteq} ES$, iff $R' \subseteq R$ (respectively $E' \subseteq E$) consists of all and only elements (T, a) in R (E) s.t. $\nexists T' \subset T, (T', a) \in R$ (E).

[14]Please note that conflict–free and self–supporting semantics were not explicitly stated in Theorem 3.16 in [78], but were shown in the proofs [77]

Theorem 4.11. *Let $ES = (A, R, E)$ be an EAS and $ES^{min} = (A, R', E')$ its minimal form. A set of arguments $S \subseteq A$ is a σ–extension in ES where $\sigma \in \{$conflict–free, sefl–supporting, admissible, preferred, complete, grounded, stable$\}$ iff it is a σ–extension of ES^{min}.*

Translation 4. *Let Fr^{EAS} be the collection of all EASs and Min^{EAS} the collection of those EASs that are in minimal normal form, both based on argument domain $\mathcal{U}$. The minimal form translation $min\text{-}Tr^{EAS} : Fr^{EAS} \rightarrow Min^{EAS}$ is defined as $min\text{-}Tr^{EAS}((A, R, E)) = (A, R', E')$, where $R' \subseteq R$ (respectively $E' \subseteq E$) consists of all and only elements (T, a) in R (E) s.t. $\nexists T' \subset T, (T', a) \in R$ (E).*

Redefinition of Theorem 4.11: *Let $\sigma \in \{$conflict–free, sefl–supporting, admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the identity casting functions for σ The translation $min\text{-}Tr^{EAS}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*
Analysis of Translation 4: Under the conflict–free, self–supporting, admissible, preferred, complete, grounded and stable semantics and identity casting functions, the translation $min\text{-}Tr^{EAS}$ is:

- full, target–subclass and overlapping
- argument domain preserving, attack and support relation removing
- generic, semantics domain preserving and exact
- structural

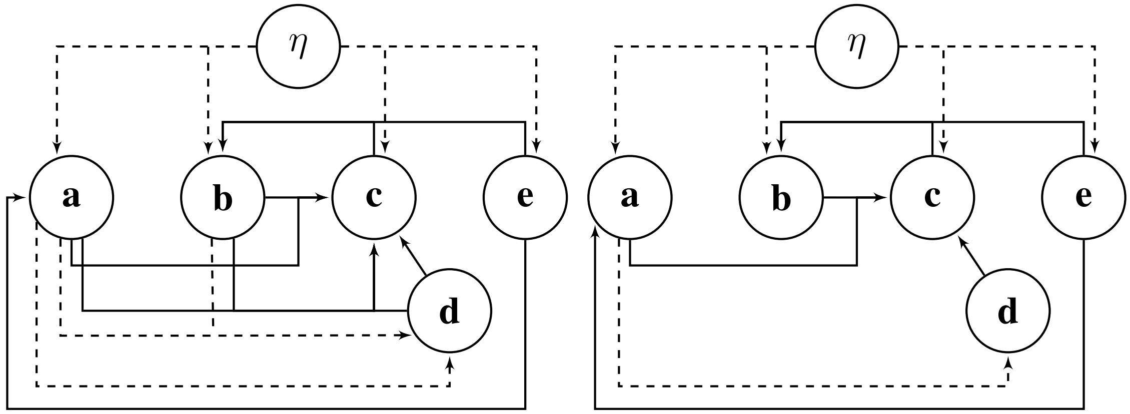
The translation $min\text{-}Tr^{EAS}$ is not modular.

Since the explanations for the properties of the translations are very similar to the ones we have given in the case of SETAFs, we will omit them here.

Example 45. Let us consider an EAS $(\{\eta, a, b, c, d, e\}, \{(\{a, b\}, c), (\{d\}, c), (\{a, b, d\}, c), (\{e\}, a), (\{c, e\}, b)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, c), (\{\eta\}, e), (\{a\}, d), (\{a, b\}, d)\})$. Its minimal form is $(\{\eta, a, b, c, d, e\}, \{(\{a, b\}, c), (\{d\}, c), (\{e\}, a), (\{c, e\}, b)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, c), (\{\eta\}, e), (\{a\}, d)\})$. Both are depicted in Figure 33. We can observe that even though the $\{a, b\}$ support for d is removed, the argument is still e–supported by $\{\eta, a\}$. Moreover, in order to defend against d, one would have to attack all of its evidential sequences, which will always contain a. Thus e has the power to defend c against d no matter whether the redundant support is present or not. Additionally, since it takes care of the $(\{d\}, c)$ attack, it also defends c from the $\{a, b, d\}$ attack, and including it in the framework is no longer necessary.

4.1.5 ADF Redundancy–Free Form

The minimal normal form for ADFs will be referred to as the redundancy–free form, though please note there is a slight difference between how it works in ADFs



(a) Sample EAS ES (b) Minimal form of ES

Figure 33: Sample EAS and its minimal normal form framework

and in other frameworks. In e.g. SETAFs, if an attack set is removed, it does not mean that the attackers and targets become unrelated. For example, $SF = (\{a, b, c\}, \{(\{a\}, c), (\{b\}, c), (\{a, b\}, c)\})$ contains a redundant conflict $(\{a, b\}, c)$. Nevertheless, even after we remove it, both a and b are still attackers of c. This means that from the ADF perspective, these parents are not redundant, and the (functional) ADF corresponding both to SF and SF^{min} would be the same (see Section 6.5). The framework is considered redundancy–free, even though SF is not. This behavior is another example of the differences between ADFs and other structures that were described in Section 2.3.9.

Let us now proceed with defining the redundancy–free form. From each acceptance condition, we will remove the origins of the redundant links, i.e. those links are both supporting and attacking (see Section 2.3). Please note that this removal can be done both with a "negative" reduct, i.e. the one we have recalled and in which removed arguments are assumed to be false (Definitions 2.135 and 4.3), and with a positive one, where they are assumed to be true [48]. This is a result of the fact that neither the presence nor absence of the redundant parents affects the outcome of the condition.

Definition 4.12. Let $D = (A, L, C)$ be an ADF, $a \in A$ be an argument, C_a its acceptance condition and $E \subseteq par(a)$ the set of all parents of a s.t. for every $e \in E$, $(e, a) \in L$ is not redundant. The **redundancy–free form** of C_a, denoted C_a^E, is the reduct of C_a w.r.t. E. Then $D^r = (A, L^r, C^r)$ is the redundancy–free form of D, where $L^r \subseteq L$ is the set of all links in L that are not redundant and $C^r = \{C_a^E \mid C_a \in C$, C_a^E is the reduct of C_a w.r.t. the set E of not redundant parents of $a\}$.

Theorem 4.13. *Let $D = (A, L, C)$ be an ADF and $D^r = (A, L^r, C^r)$ its redundancy–free form. A set $E \subseteq A$ is a σ–extension of D, where $\sigma \in \{$conflict–free, pd–acyclic conflict–free, model, stable, grounded, acyclic grounded, xy–admissible, xy–complete, xy–preferred$\}$ and $x, y \in \{a, c\}$ iff it is a σ–extension of D^r. A three–valued interpretation on*

A is a δ–labeling of D, where $\delta \in \{$three–valued model, admissible, preferred, complete, grounded$\}$ iff it is a δ–labeling of D^r.

We can now put these results into our classification system. Please observe that the translation has the same properties as the previous minimal normal forms; in particular, it is generic, exact, and not modular.

Translation 5. *Let Fr^{ADF} be the collection of all ADFs and RF^{ADF} the collection of those ADFs that do not contain redundant links, both of them on argument domain $\mathcal{U}$. The redundancy–free form translation $rf\text{-}Tr^{ADF}$ is defined as $rf\text{-}Tr^{ADF}((A, L, C)) = (A, L', C')$, where $L' \subseteq L$ is the set of links that are not redundant and $C' = \{C'_a \mid C_a \in C$, C'_a is the reduct of C_a w.r.t. the set of not redundant parents of $a\}$.*

Redefinition of Theorem 4.13: *Let $\sigma \in \{$conflict–free, pd–acyclic conflict–free, model, stable, grounded, acyclic grounded, xy–admissible, xy–complete, xy–preferred$\}$, where $x, y \in \{a, c\}$, be an extension–based semantics and $\delta \in \{$three–valued model, admissible, preferred, complete, grounded$\}$ a labeling–based semantics for ADFs. Let SC^{Tr}_{σ} and SC^{Tr}_{δ} be identity casting functions for σ and δ. The translation $rf\text{-}Tr^{ADF}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$ and $(\delta, SC^{Tr}_{\delta})$.*

Analysis of Translation 5: Under the conflict–free, pd–acyclic conflict–free, xy–admissible, xy–preferred, xy–complete, grounded, acyclic grounded, model, stable, three–valued model, labeling admissible, labeling preferred, labeling complete and labeling grounded semantics and identity casting functions, the translation $rf\text{-}Tr^{ADF}$ is:

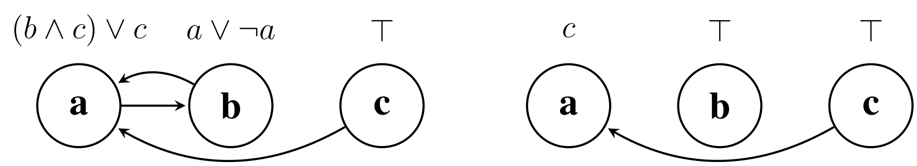
- full, target–subclass and overlapping
- argument domain preserving and relation removing
- generic, semantics domain preserving and exact
- structural

The translation $rf\text{-}Tr^{ADF}$ is neither $\oplus$ nor $\otimes$–modular.

Explanation. The choice of domains in the translation and the fact that ADFs without redundant links do not account for all the ADFs “out there” results in our approach being full and target–subclass. As two different ADFs can be assigned a single redundancy–free form, the translation is also overlapping. For example, let $(\{a, b\}, \{C_a = \{\emptyset : in\}, C_b = \{\emptyset : in\}\})$ and $(\{a, b\}, \{C_a = \{\emptyset : in, \{b\} : in\}, C_b = \{\emptyset : in\}\})$ be two ADFs in the functional representation. Their redundancy free forms are the same and identical to the first framework. The translation obviously preserves both argument and semantics domains. The exactness of our method and the fact it is generic follow from Theorem 4.13. The point of the approach is to remove “worthless” relations, thus classifying it as relation removing should be clear. Furthermore, since it does not require any knowledge on the ADF semantics and modifies the acceptance conditions in a structural manner, the translation is structural.

Just like the minimal normal form translations for other frameworks, $rf\text{-}Tr^{ADF}$ is not modular w.r.t. any of the operators. We can repeat the previously given example in the ADF setting. Consider the ADFs $D_1 = (\{a,b,c\},\{C_a = \top, C_b = \top, C_c = \neg a\})$ and $D_2 = (\{a,b,c\},\{C_a = \top, C_b = \top, C_c = \neg a \vee \neg b\})$. The are both already in the redundancy free–form. However, $D_1 \otimes D_2 = (\{a,b,c\},\{C_a = \top, C_b = \top, C_c = \neg a \wedge (\neg a \vee \neg b)\})$ is not. In this case, the (b,c) link is redundant. Let now $D_3 = (\{a,b,c\},\{C_a = \top, C_b = \top, C_c = \neg a \wedge \neg b\})$. Again, it is in redundancy–free form, but the framework $D_1 \oplus D_3 = (\{a,b,c\},\{C_a = \top, C_b = \top, C_c = \neg a \vee (\neg a \wedge \neg b)\})$ is not. Also in this case the (b,c) link becomes unnecessary. ■

Example 46. Let $D = (\{a,b,c\},\{C_a = (b \wedge c) \vee c, C_b = a \vee \neg a, C_c = \top\})$ be the ADF depicted in Figure 34a. The functional representation of the conditions is $C_a = \{(\emptyset, out), (\{b\}, out), (\{c\}, in), (\{b,c\}, in)\}$, $C_b = \{(\emptyset, in), (\{a\}, in)\}$ and $C_c = \{(\emptyset, in)\}$. We can observe that in this case, both (a,b) and (b,a) are redundant connections. Adding b to $\emptyset$ and $\{c\}$ in no way changes the outcome of C_a; similar for a and $\emptyset$ in C_b. By removing the respective mappings, we obtain conditions $C_a^r = \{(\emptyset, out), (\{c\}, in)$ and $C_b^r = \{(\emptyset, in)\}$, corresponding to formulas c and $\top$. The redundancy–free form of D is thus $(\{a,b,c\},\{C_a = c, C_b = \top, C_c = \top\})$, as seen in Figure 34b.



(a) Sample ADF

(b) Redundancy–free ADF

Figure 34: Sample ADF and its redundancy–free form

4.2 Cleansed Form

Previously, we have mentioned that ADFs have the ability to handle arguments that can be interpreted as "I do not exist" or "I am known to be false" (see Section 2.3.9). Consequently, anything derived from them is interpreted in the same manner. This ability can be seen as unique to ADFs and thus the cleansed form will be considered only for this framework. A falsum argument – or one depending on it – possesses no standard evaluation. Consequently, it will not appear in a conflict–free extension and will always be automatically falsified in any type of range (see discussion in Section 2.3.5, Lemmas 2.125 and 2.132). This means we can consider removing such arguments while still preserving all of the extension–based semantics. However, due to the change in the set of arguments, the labeling–based semantics will not remain completely unaffected.

The point of the cleansed form is to obtain a framework in which every argument has a satisfiable acceptance condition, i.e. for every argument and its acceptance condition, there exists a set of arguments evaluating the condition to in. In order to create this form, we can consider two procedures. In the first approach, resembling the original definition of standard range (Definition 2.124), we can identify arguments possessing no in mappings in their conditions, reduce the framework in order to remove them, and repeat the process till no further modifications are required. The other approach, resembling the evaluation definition of range (Lemma 2.125), would find arguments possessing no standard evaluations and reduce the framework. Due to the fact that falsum arguments will not appear in sound pd–functions, neither them nor arguments based on them can possess standard evaluations and thus the two methods are in fact equivalent. Consequently, we will proceed with the simpler, single step method.

Definition 4.14. Let $D = (A, L, C)$ be an ADF. The **cleansed form** of D is the reduct of D w.r.t. A', where $A' \subseteq A$ is the set of all and only arguments on A that possess a standard evaluation on A.

Theorem 4.15. *Let $D = (A, L, C)$ be an ADF and $D^c = (A', L', C')$ its cleansed form. A set $E \subseteq A$ is a σ–extension of D, where $\sigma \in \{$conflict–free, pd–acyclic conflict–free, model, stable, grounded, acyclic grounded, xy–admissible, xy–complete, xy–preferred$\}$ and $x, y \in \{a, c\}$, iff it is a σ–extension of D^c.*

Theorem 4.16. *Let $D = (A, L, C)$ be an ADF and $D^c = (A', L', C')$ its cleansed form. If v is σ–labeling of D, where $\sigma \in \{$three–valued model, admissible, preferred, complete, grounded$\}$, then $v|_{A'}$ is a σ–labeling of D^c [15]. If v is a σ–labeling of D^c, then the $\mathbf{f}$–completion of v to A is a σ–labeling of D.*

We can now put these results into our system. At this point the added value of our redefinitions becomes more apparent. Although the results above are correct and follow the typical "if then" construction, at first glance it can be difficult to notice that this translation, when combined with using the $\mathbf{f}$–completion for labeling retrieval, is not always strong. Not every three valued model or admissible labeling of the original framework is brought back. This is due to the fact that these semantics do not use any types of maximality and in some cases, the $\mathbf{u}$–completions would be sufficient as well. However, please note that this weakness is caused by the used casting functions, not the fact that any labeling is really "lost", as will become visible in Example 47. Nevertheless, the more common scheme for the semantics theorem can, even though not intentionally, mask such issues.

Translation 6. *Let Fr^{ADF} be the collection of all ADFs and CL^{ADF} the collection of cleansed ADFs, both based on argument domain $\mathcal{U}$. The cleansed normal form translation cl-$Tr^{ADF} : Fr^{ADF} \to CL^{ADF}$ is defined as cl-$Tr^{ADF}((A, L, C)) = (A', L', C')$, where (A', L', C') is the reduct of (A, L, C) w.r.t. the set $A' = \{a \in A \mid a$ has a standard evaluation on $A\}$.*

[15] Recall that $v|_A$ stands for the subinterpretation of v defined over A.

Redefinition of Theorem 4.15: *Let $\sigma \in \{$conflict–free, pd–acyclic conflict–free, model, stable, grounded, acyclic grounded, xy–admissible, xy–complete, xy–preferred$\}$ where $x, y \in \{a, c\}$ be a semantics and let SC^{Tr}_σ be the identity casting functions for σ. The translation $cl\text{-}Tr^{ADF}$ is strong and semantics bijective under (σ, SC^{Tr}_σ).*

Redefinition of Theorem 4.16: *Let $\sigma \in \{$three–valued model, admissible, preferred, complete, grounded$\}$ be a semantics and let SC^{Tr}_σ be the addition casting functions for σ defined as $SC^X_\sigma(v) = v \cup \{(a, \mathbf{f}) \mid a \in A \setminus A'\}$, where $X = (A, C) \in Fr^{ADF}$, A' is the set of arguments of $cl\text{-}Tr^{ADF}(X)$, and $v \in \sigma(cl\text{-}Tr^{ADF}(x))$. The translation $cl\text{-}Tr^{ADF}$ is strong and semantics bijective under the preferred, complete and grounded semantics and the defined casting functions. It is $\supseteq$–weak under the three–valued model and admissible semantics and the defined casting functions.*

Analysis of Translation 6: Under the conflict–free, pd–acyclic conflict–free, model, stable, grounded, acyclic grounded, xy–admissible, xy–complete and xy–preferred semantics for $x, y \in \{a, c\}$ and their identity casting functions, and under the complete, preferred and grounded labeling–based semantics and their addition casting functions, the translation $cl\text{-}Tr^{ADF}$ is:

- full, target–subclass, overlapping
- argument domain preserving, argument removing, induced relation removing
- generic and semantics domain preserving
- semi–structural

Translation $cl\text{-}Tr^{ADF}$ is neither $\otimes$ nor $\oplus$ modular. Under the conflict–free, pd–acyclic conflict–free, model, stable, grounded, acyclic grounded, xy–admissible, xy–complete and xy–preferred semantics for $x, y \in \{a, c\}$ and their identity casting functions, the translation is exact. Under the complete, preferred and grounded labeling–based semantics and their addition casting functions, the translation is faithful.

Explanation. Since any framework can be transformed into the cleansed form, the translation $cl\text{-}Tr^{ADF}$ is easily full. Moreover, only the cleansed ADFs can be produced, and thus it is also target–subclass. Additionally, the translation is overlapping; we can see that the frameworks $(\{a, b\}, \{C_a = \top, C_b = \bot\})$ and $(\{a, b, c\}, \{C_a = \top, C_b = \bot, C_c = \bot\})$ will both be transformed into $(\{a\}, \{C'_a = \top\})$. This approach is clearly both argument and semantics domain preserving. From the amount of the handled semantics it also holds that $cl\text{-}Tr^{ADF}$ is generic. Clearly, it is also argument removing – this is after all the point of the translation. Relations between arguments are removed iff the deleted arguments take part in them, thus we can speak about the induced removal. However, please note that certain removals can cause the remaining relations to change their nature and become redundant. Let us consider the framework $(\{a, b, c\}, \{C_a = \top, C_b = \bot, C_c = \neg a \vee \neg b\})$. The links from a and b to c are attacking. The cleansed form of this structure is $(\{a, c\}, \{C'_a = \top, C'_c = \neg a \vee \top\})$ (the functional version of C'_c is $\{\emptyset : in, \{a\} : in\}$).

Therefore, the link from a to c is now both attacking and supporting and is considered redundant.

Possessing a standard evaluation is a rather basic requirement, in some ways even more basic than conflict–freeness, and we can see that it is extension–based semantics independent. It can be seen as verifying if an argument "exists". Although it suffices to construct the decisive interpretations by mapping to true those arguments for which a condition is satisfied and assuming everything else is false, constructing an evaluation from them requires some computation and thus we choose to classify this translation as semi–structural.

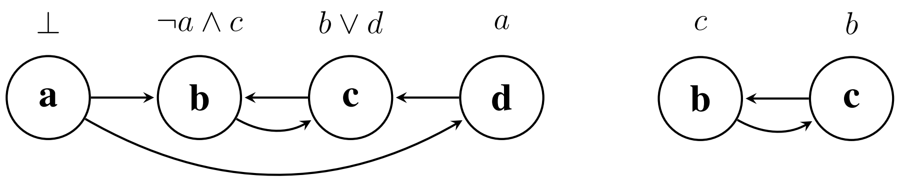
Let us now focus on $\otimes$–modularity and let $D_1 = (\{a\}, \{C_a = a\})$ and $D_2 = (\{a\}, \{C_a = \neg a\})$ be two ADFs. We can observe that $cl\text{-}Tr^{ADF}(D_1) = D_1$ and $cl\text{-}Tr^{ADF}(D_2) = D_2$. However, the framework $D_1 \otimes D_2 = (\{a\}, \{C_a = a \wedge \neg a\})$ is not in cleansed form; the condition of a is equivalent to $\bot$. Consequently, the cleansed form of this structure is the empty framework and thus $cl\text{-}Tr^{ADF}(D_1) \otimes cl\text{-}Tr^{ADF}(D_2) \neq cl\text{-}Tr^{ADF}(D_1 \otimes D_2)$. Therefore, our translation is not $\otimes$–modular.

For $\oplus$–modularity, let us look at the frameworks $D_1 = (\{a, b\}, \{C_a = \bot, C_b = a\})$ and $D_2 = (\{a, b\}, \{C_a = b, C_b = \bot\})$. The cleansed form of both of them is the empty framework. However, in the framework $D_1 \oplus D_2 = (\{a, b\}, \{C_a = b \vee \bot, C_b = a \vee \bot\})$ both arguments a and b possess standard evaluations and thus the structure will not be affected by the translation. Clearly, it is not empty, and thus $cl\text{-}Tr^{ADF}$ is not $\oplus$–modular.

The fact that $cl\text{-}Tr^{ADF}$ is exact (faithful) under listed semantics comes from Theorems 4.15 and 4.16, their proofs and redefinitions. ■

Example 47. Let $D = (\{a, b, c, d\}, \{C_a = \bot, C_b = \neg a \wedge c, C_c = b \vee d, C_d = a\})$ be an ADF and $D^{Cln} = (\{b, c\}, \{C_b = c, C_c = b\})$ its cleansed form. We can observe that "cleansing" the framework removed both the falsum argument a and argument d based on it. Let us focus on $\emptyset$. Its standard, partially acyclic and acyclic discarded sets are respectively $\{a, d\}$, $\{a, b, c, d\}$ and $\{a, b, c, d\}$ in D. When we look at D^{Cln}, we obtain $\emptyset$, $\{b, c\}$ and $\{b, c\}$. Therefore, the cleansed discarded sets are the result of deleting from the original sets the arguments that were removed by the translation. When we focus on $\{b, c\}$, the discarded sets are all $\{a, d\}$ in D and $\emptyset$ in D^{Cln}. In both of these frameworks, b and c possess only standard evaluations that cannot be made acyclic. It is now easy to show that the extensions of D and D^{Cln} coincide.

The admissible labelings of D are $v_1 = \{a : \mathbf{u}, b : \mathbf{u}, c : \mathbf{u}, d : \mathbf{u}\}$, $v_2 = \{a : \mathbf{u}, b : \mathbf{u}, c : \mathbf{u}, d : \mathbf{f}\}$, $v_3 = \{a : \mathbf{f}, b : \mathbf{u}, c : \mathbf{u}, d : \mathbf{u}\}$, $v_4 = \{a : \mathbf{f}, b : \mathbf{u}, c : \mathbf{u}, d : \mathbf{f}\}$, $v_5 = \{a : \mathbf{u}, b : \mathbf{f}, c : \mathbf{f}, d : \mathbf{f}\}$, $v_6 = \{a : \mathbf{f}, b : \mathbf{f}, c : \mathbf{f}, d : \mathbf{f}\}$, all corresponding to the $\emptyset$ extension, and $v_7 = \{a : \mathbf{f}, b : \mathbf{t}, c : \mathbf{t}, d : \mathbf{u}\}$ and $v_8 = \{a : \mathbf{f}, b : \mathbf{t}, c : \mathbf{t}, d : \mathbf{f}\}$, which produce $\{b, c\}$. The answers produced by D^{Cln} are $w_1 = \{b : \mathbf{u}, c : \mathbf{u}\}$, $w_2 = \{b : \mathbf{f}, c : \mathbf{f}\}$ and $w_3 = \{b : \mathbf{t}, c : \mathbf{t}\}$, which again are associated with extensions $\emptyset$ and $\{b, c\}$. We can observe that although the sets of arguments accepted in both frameworks are the same, there is a difference on those mapped to $\mathbf{f}$ and $\mathbf{u}$. Nevertheless, one can easily observe that D^{Cln} labelings can be easily obtained as limitations of the interpretations of D to the set $\{b, c\}$. Moreover, some of the D answers are retrieved by extending D^{Cln} results with $\mathbf{f}$ mappings to arguments a and d. If we move to complete semantics (labelings v_4, v_6, v_8, and all w's), the relation between the interpretations of both frameworks becomes bijective. The $\mathbf{f}$–completion also retrieves precisely the desired answers.



(a) Sample ADF D (b) Cleansed form of D

Figure 35: Sample ADF and its cleansed form

4.3 Validity Forms

In bipolar frameworks we often deal with the concept of validity of an argument, which based on current semantics is related to support cycles. A valid argument is one that possesses at least one powerful/evidential sequence or an acyclic pd–evaluation in the framework. The arguments that are not valid do not need to be defended from (or are automatically defended from) and do not show up in any extensions. Therefore, removing them from a given framework will not affect the semantics. Filtering out the invalid arguments will give us the weak validity from.

Independently of whether a framework is weakly valid or not, it can happen that not all of the support that an argument receives can be considered valid or relevant. In other words, not all of the supporters might be used in constructing a proper sequence or evaluation for a given argument. Consequently, we may consider removing such relations, since it will not affect how e.g. coherent or self–supporting sets look like. This brings us to the relation valid form.

Unfortunately, neither weak nor relation validity forms ensure that the framework is completely free from support cycles. In other words, despite the fact that every argument has a sequence (evaluation) and every relation can be used to construct a desirable sequence (evaluation), there can exist a set of arguments E s.t. all arguments in E are sufficiently supported by E, but the set is not coherent (self–supporting, pd–acyclic). Making sure that every combination of valid supports is valid itself brings us to the strong validity form. Please note that for now, we will not provide translations to strongly valid forms, even though we believe they can be created. Consequently, we let the reader see them as subclasses if he/she wishes to do so.

What needs to be said about all of the validity normal form translations is that, unlike in the previously described approaches, they are indisputably semantical. This means we can expect them to be computationally more expensive than other forms. However, depending on how a given framework was obtained, it might already be in a given form.

Moreover, due to the properties of the validity forms, certain transformations and semantics can be simplified. The last – strongly valid – form will be most important to us. Even though we do not have any strong form transformations yet, various translations from the attack–based to bipolar frameworks will produce structures in this form (see for example Translations 21, 22, 23, 29, 30 and 31). This form will allow us to simplify certain translations, in particular the defender ones (see Sections 10.2.2 and 11.2.3). Let us now introduce our three validity forms.

4.3.1 Weak Validity

In AFNs, EASs and aa–semantics of ADFs, arguments that are not acyclic do not appear in any extensions and do not need to be defended from. Consequently, their removal does not change the extensions we can obtain. Moreover, they are frequently "lost" when the target framework of our translation does not have the validity concept. For example, this is the case when we consider translations from bipolar frameworks to the attacked–based structures (see e.g. Translations 61 and 71). Thus, in the weak validity form we simply remove just undesired arguments. In what follows we will show how to transform AFNs, EASs and ADFs into this form. The produced structures will be again subframeworks of the original ones.

4.3.1.1 AFN Weak Validity

In the AFN case, we simply require arguments to possess at least one powerful sequence in the framework. Equivalently, we can simply state that they are contained in a coherent extension:

Definition 4.17. Let $FN = (A, R, N)$ be an AFN and $A' = \{a \in A \mid$ there exists a powerful sequence for a on $A\}$. The trimmed full subframework $FN^{A'} = (A', R', N')$ of FN induced by A' is the **weak validity form** of FN. We denote it with $FN^{A'} \overset{wv}{\sqsubseteq} FN$.

We can observe that the only semantics affected by this translation is the conflict–free semantics. This is due to the fact that in this case, the validity of an argument does not play any role. However, starting from the coherent semantics, we have a one–to–one relation between the extensions of the original framework and its weakly valid normal form.

Theorem 4.18. *Let $FN = (A, R, N)$ be an AFN and $FN^{wv} = (A', R', N')$ be its weak validity form. A set $E \subseteq A$ is a σ–extension of FN, where $\sigma \in \{$coherent, strongly coherent,admissible, preferred, complete, grounded, stable$\}$, iff it is a σ–extension of FN^{wv}. If $E \subseteq A$ is a conflict–free extension of FN, then $E \cap A'$ is conflict–free in FN^{wv}. If $E \subseteq A'$ is conflict–free in FN^{wv}, then it is conflict–free in FN.*

We can now change our definition into a translation, put it into the system and analyze its properties.

Translation 7. *Let Fr^{AFN} be the collection of all AFNs and WV^{AFN} the collection of weakly valid AFNs, both based on argument domain $\mathcal{U}$. The weakly valid normal form translation $wv\text{-}Tr^{AFN} : Fr^{AFN} \rightarrow WV^{AFN}$ is defined as $wv\text{-}Tr^{AFN}((A, R, N)) = (A', R', N')$, where (A', R', N') is the trimmed full subframework of (A, R, N) induced by the set $A' = \{a \in A \mid a$ has a powerful sequence on $A\}$.*

Redefinition of Theorem 4.18: *Let $\sigma \in \{$coherent, strongly coherent, admissible, complete, preferred, grounded, stable$\}$ be a semantics and let SC^{Tr}_{σ} be the identity casting functions for σ. The translation $wv\text{-}Tr^{AFN}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$. It is $\supseteq$–weak under conflict–free semantics and identity casting functions.*

Analysis of Translation 7: Under conflict–free, strongly coherent, coherent, admissible, preferred, complete, grounded and stable semantics and their removal casting functions, the translation $wv\text{-}Tr^{AFN}$ is:

- full, target–subclass, overlapping
- argument domain preserving, argument removing, induced attack and support relation removing
- generic and semantics domain preserving
- semantical

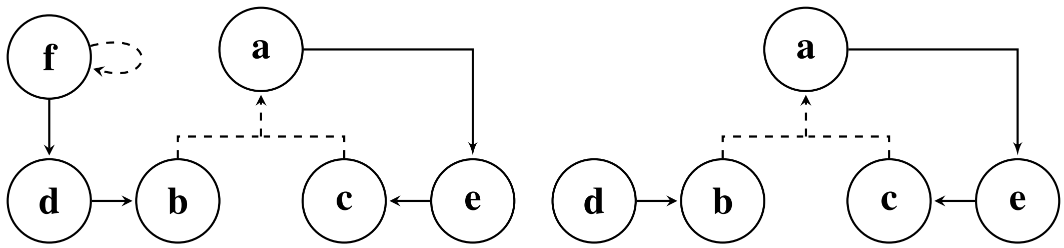
Translation $wv\text{-}Tr^{AFN}$ is not modular. Under strongly coherent, coherent, admissible, preferred, complete, grounded and stable semantics $wv\text{-}Tr^{AFN}$ is generic and exact.

Explanation. Since any framework can be transformed into weakly valid form and all the weakly valid frameworks do not account for all the possible AFNs, the translation is easily full and target–subclass. More than one framework can have the same weakly valid form. For example, consider structures $(\{a, b\}, \emptyset, \{(\{b\}, b)\})$ and $(\{a, b, c\}, \emptyset, \{(\{b\}, b), (\{c\}, c)\})$ – they are both transformed into $(\{a\}, \emptyset, \emptyset)$. Consequently, $wv\text{-}Tr^{AFN}$ is overlapping.

The translation clearly preserves the argument and semantics domain. It also is argument removing, which was the point of the whole approach. Since deleting an argument causes us to delete any supporting and attacking edges related to it, we can classify it as induced removal. Due to the amount of the handled semantics, $wv\text{-}Tr^{AFN}$ is generic. Moreover, as validity of an argument is a semantical notion in AFNs, the translation is clearly semantical.

In order to show that our translation is not modular, let us consider two frameworks $FN_1 = (\{a, b\}, \emptyset, \{(\{a\}, b)\})$ and $FN_2 = (\{a, b\}, \emptyset, \{(\{b\}, b)\})$. In the first case, b is powerful in $\{a, b\}$, and in the other it is not. The weak validity forms of the two frameworks are $FN_1^{wv} = (\{a, b\}, \emptyset, \{(\{a\}, b)\})$ and $FN_2^{wv} = (\{a\}, \emptyset, \emptyset)$ respectively. We can observe that $FN_1^{wv} = FN_1$. The union of FN_1^{wv} and FN_2^{wv} is simply FN_1^{wv} again. Let us now look at the framework $FN_1 \cup FN_2 = (\{a, b\}, \emptyset, \{(\{a\}, b), (\{b\}, b)\})$. We can observe that b is no longer powerful in $\{a, b\}$. The weak validity form of $FN_1 \cup FN_2$ is $(\{a, b\}, \emptyset, \emptyset)$, which is different from FN_1^{wv}. Thus, translation $wv\text{-}Tr^{AFN}$ is not modular.

The fact that it is exact under strongly coherent, coherent, admissible, preferred, complete, grounded and stable semantics follows from Theorem 4.18 and its redefinition. We cannot say this about conflict–free semantics, as under it $wv\text{-}Tr^{AFN}$ is not even strong. ■



(a) Sample AFN FN

(b) Weakly valid form of FN

Figure 36: Sample AFN and its weakly valid form

Example 48. We can consider the AFN $FN = (\{a, b, c, d, e, f\}, \{(a, e), (d, b), (e, c), (f, d)\}, \{(\{b, c\}, a), (\{f\}, f)\})$ previously analyzed in Example 16 and for convenience again depicted in Figure 36. The admissible extensions of FN are $\emptyset$, $\{d\}$, $\{a, c\}$, $\{d, e\}$ and $\{a, c, d\}$. The sets $\{d\}$, $\{d, e\}$ and $\{a, c, d\}$ are also complete, with the first one being grounded and the latter two being preferred. In this case, both $\{d, e\}$ and $\{a, c, d\}$ are stable.

The weakly valid form of FN is $FN^{wv} = (\{a, b, c, d, e\}, \{(a, e), (d, b), (e, c)\}, \{(\{b, c\}, a)\})$. Previously, d was defended by any set due to the fact that f possessed no coherent set. In FN^{wv}, it is trivially defended as it is not attacked at all. It is also easy to observe that if a set of arguments could not defend against d in FN, then it cannot do it in FN^{wv} either and vice versa. Finally, we can observe that as f was not present in any coherent set, it was not present in any extension either. With this at hand it can now be easily shown that the extensions of FN^{wv} are the same as in FN.

4.3.1.2 EAS Weak Validity

The EAS weak validity form is very similar to the AFN form. In this case, we simply remove the arguments not possessing an evidential sequence in the framework. Equivalently, we can delete the arguments not appearing in any self–supporting set:

Definition 4.19. Let $ES = (A, R, E)$ be an EAS and $A' = \{a \in A \mid$ there exists at least one evidential sequence for a on $A\}$. The full subframework $ES^{A'} = (A', R', E')$ of ES induced by A' is the **weak validity form** of ES. We denote it with $ES^{A'} \overset{wv}{\sqsubseteq} ES$.

Theorem 4.20. *Let $ES = (A, R, E)$ be an EAS and $ES^{wv} = (A', R', E')$ be its weak validity form. A set $S \subseteq A$ is a σ–extension of ES, where $\sigma \in \{$self-supporting, strongly* *self–supporting, admissible, preferred, complete, grounded, stable$\}$, iff it is a σ–extension of ES^{wv}. If $S \subseteq A$ is a conflict–free extension of ES, then $S \cap A'$ is conflict–free in ES^{wv}. If $S \subseteq A'$ is conflict–free in ES^{wv}, then it is conflict–free in ES.*

We can now put these results into our translation system in the same manner as in the AFN case.

Translation 8. *Let Fr^{EAS} be the collection of all EASs and WV^{EAS} the collection of weakly valid EASs, both based on argument domain $\mathcal{U}$. The weakly valid normal form translation $wv\text{-}Tr^{EAS} : Fr^{EAS} \rightarrow WV^{EAS}$ is defined as $wv\text{-}Tr^{EAS}((A, R, E)) = (A', R', E')$, where (A', R', E') is the full subframework of (A, R, E) induced by the set $A' = \{a \in A \mid a$ has an evidential sequence on $A\}$.*

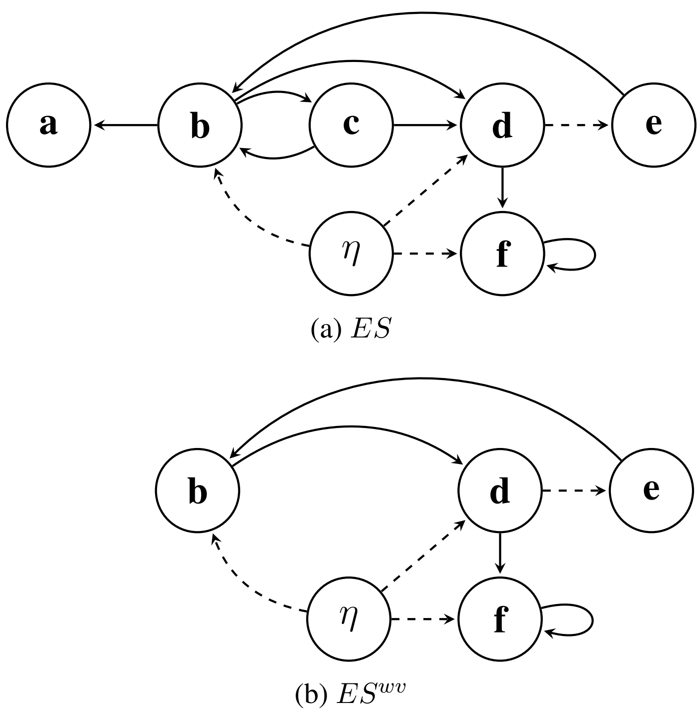
Redefinition of Theorem 4.20: *Let $\sigma \in \{$self–supporting, self–supporting conflict–free, admissible, complete, preferred, grounded, stable$\}$ be a semantics and let SC_σ^{Tr} be the identity casting functions for σ. The translation $wv\text{-}Tr^{EAS}$ is strong and semantics bijective under (σ, SC_σ^{Tr}). It is $\supseteq$–weak under the conflict–free semantics and identity casting functions.*
Analysis of Translation 8: Under conflict–free, self–supporting, self–supporting conflict–free, admissible, complete, preferred, grounded and stable semantics and identity casting functions, the translation $wv\text{-}Tr^{EAS}$ is:

- full, target–subclass, overlapping
- argument domain preserving, argument removing, induced attack and support relation removing
- semantics domain preserving
- semantical

Translation $wv\text{-}Tr^{EAS}$ is not modular. Under self–supporting, self–supporting conflict–free, admissible, complete, preferred, grounded and stable semantics, it is exact and generic.

Explanation. The explanations we have given in the AFN case (analysis of Translation 7) also hold in the EAS case. We will only provide a different example to show lack of modularity. Consider a simple framework $ES = (\{\eta, a, b\}, \emptyset, \{(\{\eta\}, a), (\{a\}, b)\})$ and its two subframeworks $ES_1 = (\{\eta, a, b\}, \emptyset, \{(\{\eta\}, a)\})$ and $ES_2 = (\{\eta, a, b\}, \emptyset, \{(\{a\}, b)\})$. We can observe that $ES_1 \cup ES_2 = ES$. The framework ES is already in weak validity form and thus will not be affected by the translation. The weak validity forms of its two subframeworks are $ES_1^{wv} = (\{\eta, a\}, \emptyset, \{(\{\eta\}, a)\})$ and $ES_2^{wv} = (\{\eta\}, \emptyset, \emptyset)$ respectively. Their union is equal to ES_1^{wv}, which is clearly not the same as ES. Therefore, our translation is not modular. ■

Example 49. Let $ES = (\{\eta, a, b, c, d, e, f\}, \{(\{b\}, a), (\{b\}, c), (\{b\}, d), (\{c\}, b), (\{c\}, d), (\{d\}, f), (\{e\}, b), (\{f\}, f)\}, \{(\{\eta\}, b), (\{\eta\}, d), (\{\eta\}, f), (\{d\}, e)\})$ be the EAS depicted in Figure 37a. We can observe that neither a nor c can be present in an extension of ES that is at least self–supporting; the arguments simply lack evidential support. Based on Definition 2.101, we can also see that any set will always defend its arguments against the attacks from a and c. Therefore, the admissible extensions of ES are $\emptyset$, $\{\eta\}$, $\{b\}$ and $\{e, d\}$. With the exception of $\emptyset$, they are all complete. η is the grounded extension of ES and $\{b\}$ and $\{e, d\}$ are its preferred ones. Finally, $\{e, d\}$ is the only stable extension. The weakly valid form of $ES^{wv} = (\{\eta, b, d, e, f\}, \{(\{b\}, d), (\{d\}, f), (\{e\}, b), (\{f\}, f)\}, \{(\{\eta\}, b), (\{\eta\}, d), (\{d\}, e), (\{\eta\}, f)\})$, visible in Figure 37b. It is easy to verify that the admissible, complete, preferred, grounded and stable extensions it produces are exactly the ones we could obtain from η.



(a) ES

(b) ES^{wv}

Figure 37: Sample EAS and its weakly valid form

4.3.1.3 ADF Weak Validity

In the cleansed forms of ADFs, we were removing from the framework the arguments that did not possess standard evaluations. In the weak validity form, we restrict this further to acyclic pd–evaluations. The arguments for which there are no such evaluations cannot appear in any aa–extensions and are automatically in the acyclic discarded set. Therefore, their removal does not affect this type of semantics. Deleting the arguments and the relevant relations will be done with the help of the reduct (see Definitions 2.135 and 4.3). However, unlike in the EAS and AFN case, where the powerful and evidential sequences were not affected by the translation, the evaluations need to undergo some modifications. The deleted arguments will not appear in the pd–sequences, but can appear in the blocking sets. Consequently, the blocking sets and the decisively in interpretations for the remaining arguments need to be adapted and reanalyzed.

Definition 4.21. Let $D = (A, L, C)$ be an ADF and $A' = \{a \in A \mid$ a has an acyclic pd–evaluation on $A\}$. The reduct $D^{A'} = (A', L^{A'}, C^{A'})$ of D w.r.t. A' is the **weak validity form** of D. We denote it with $D^{A'} \overset{wv}{\sqsubseteq} D$.

Theorem 4.22. *Let $D = (A, L, C)$ be an ADF and $D^{wv} = (A', L', C')$ be its weak validity form. A set $E \subseteq A$ is a σ–extension of D, where $\sigma \in \{$pd–acyclic conflict–free, aa–admissible, aa–preferred, aa–complete, acyclic grounded, stable$\}$, iff it is a σ–extension of D^{wv}.*

We can observe that the weakly valid form is quite similar to cleansed – it is just a different type of evaluation that is taken into account. Thus, not surprisingly, these two are in fact related – every weakly valid ADF will also be cleansed:

Theorem 4.23. *Let $D = (A, L, C)$ be an ADF. If D is weakly valid, then it is also cleansed, but not vice versa.*

We can now proceed with redefining the notions in accordance with the translation system we have introduced and analyzing their properties. Please note while conflict–free semantics did not behave that well in the case of weakly valid forms of AFNs and EASs, they are retrieved exactly in the ADF case. This is due to the difference between the definitions of this semantics in these frameworks (see Section 2.3.5) and the fact that (pd–acyclic) conflict–freeness in ADFs is in fact similar to strong coherence and self–supporting conflict–free sets in AFNs and EASs rather than pure conflict–freeness (see Definition 3.2).

Translation 9. *Let Fr^{ADF} be the collection of all ADFs and WV^{ADF} the collection of weakly valid ADFs, both based on argument domain $\mathcal{U}$. The weakly valid normal form translation $wv\text{-}Tr^{ADF} : Fr^{ADF} \rightarrow WV^{ADF}$ is defined as $wv\text{-}Tr^{ADF}((A, L, C)) = (A', L', C')$, where (A', L', C') is the reduct of (A, L, C) w.r.t. the set $A' = \{a \in A \mid a$ has an acyclic pd–evaluation on $A\}$.*

Redefinition of Theorem 4.22: *Let $\sigma \in \{$pd–acyclic conflict–free, aa–admissible, aa–preferred, aa–complete, acyclic grounded, stable$\}$ be a semantics and let SC_{σ}^{Tr} be the identity casting functions for σ. The translation $wv\text{-}Tr^{EAS}$ is strong and semantics bijective under $(\sigma, SC_{\sigma}^{Tr})$.*

Analysis of Translation 9: Under the pd–acyclic conflict–free, aa–admissible, aa–preferred, aa–complete, acyclic grounded and stable semantics and identity casting functions, the translation $wv\text{-}Tr^{EAS}$ is:

- full, target–subclass, overlapping
- argument domain preserving, argument removing, induced relation removing
- generic, semantics domain preserving and exact
- semantical

Translation $wv\text{-}Tr^{ADF}$ is neither $\otimes$ nor $\oplus$–modular.

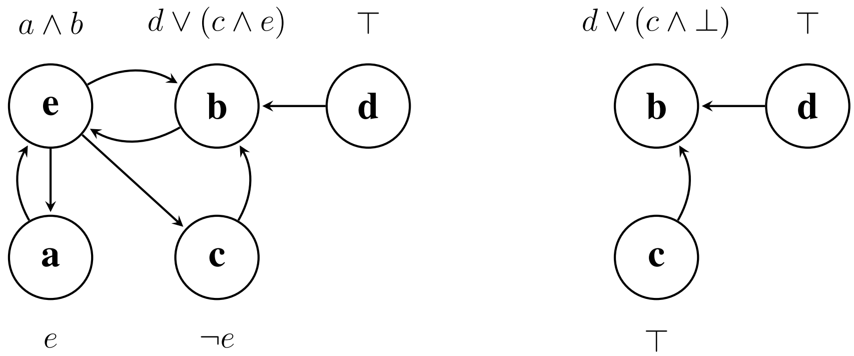
Explanation. The explanations we have given in the AFN and EAS cases (analysis of Translations 7 and 8) also hold in the ADF case. However, due to the different notions of modularity in ADFs, this property requires further analysis.

In order to show that the translation is not $\otimes$–modular, we can reuse the example given in the AFN case. Let us consider the ADF $D = (\{a, b\}, \{C_a = \top, C_b = a \wedge b\})$ and two of its subframeworks $D_1 = (\{a, b\}, \{C_a = \top, C_b = a\})$ and $D_2 = (\{a, b\}, \{C_a = \top, C_b = b\})$. Please observe that $D_1 \otimes D_2 = D$. The weak validity forms of our frameworks are $D^{wv} = (\{a\}, \{C_a = \top\})$, $D_1^{wv} = (\{a, b\}, \{C_a = \top, C_b = a\})$ and $D_2^{wv} = (\{a\}, \{C_a = \top\})$. We can observe that $D_1^{wv} \otimes D_2^{wv} = (\{a, b\}, \{C_a = \top, C_b = a\})$, which is the same as D_1^{wv} and clearly different from D^{wv}. Thus, we can observe that the translation $wv\text{-}Tr^{ADF}$ is not $\otimes$–modular.

In order to show that the translation is not $\oplus$–modular, we can adapt the example given in the EAS case. Consider a simple framework $D = (\{\eta, a, b\}, \{C_\eta = \top, C_a = \eta \vee a, C_b = a \vee b\})$ and its two subframeworks $D_1 = (\{\eta, a, b\}, \{C_\eta = \top, C_a = \eta, C_b = b\})$ and $D_2 = (\{\eta, a, b\}, \emptyset, \{C_\eta = \top, C_a = a, C_b = a\})$. We can observe that every argument in D has an acyclic pd–evaluation – $((\eta), \emptyset)$ for η, $((\eta, a), \emptyset)$ for a and $((\eta, a, b), \emptyset)$ for b. Consequently, D is already in weak validity form and thus will not be affected by the translation. Concerning D_1, the argument b does not posses an acyclic pd–evaluation. In the case of D_2, both a and b do not have acyclic pd–evaluations. Therefore, the weak validity forms of its two subframeworks are $D_1^{wv} = (\{\eta, a\}, \{C_\eta = \top, C_a = \eta\})$ and $D_2^{wv} = (\{\eta\}, \{C_\eta = \top\})$ respectively. The framework $D_1^{wv} \oplus D_2^{wv} = (\{\eta, a\}, \{C_\eta = \top, C_a = \eta\})$ is clearly different from D. Thus, translation $wv\text{-}Tr^{ADF}$ is not $\oplus$–modular. ■

Example 50. Let us come back to the ADF $D = (\{a, b, c, d, e\}, \{C_a = e, C_b = d \vee (c \wedge e), C_c = \neg e, C_d = \top, C_e = a \wedge b\})$ previously analyzed in Example 28 and for convenience again depicted in Figure 38. The acyclic conflict–free extensions of D are $\emptyset, \{c\}, \{d\}, \{b, d\}, \{c, d\}$ and $\{b, c, d\}$. Coincidentally, they are also our aa–admissible sets, with $\{b, c, d\}$ being our single complete, preferred, stable and acyclic grounded extension.

The minimal decisively in arguments for our arguments are $v_a = \{e : \mathbf{t}\}$, $v_b^1 = \{d : \mathbf{t}\}$, $v_b^2 = \{c : \mathbf{t}, e : \mathbf{t}\}$, $v_c = \{e : \mathbf{f}\}$, $v_d = \emptyset$ and $v_e = \{a : \mathbf{t}, b : \mathbf{t}\}$. We can observe that there is a positive dependency cycle between a and e. Thus, we will need to remove these arguments. Both c and d have trivial acyclic evaluations due to the fact that they both



(a) Sample ADF D

(b) Weakly valid form of D

Figure 38: Sample ADF and its weakly valid form

satisfy a_0 requirements of a pd–sequence. Finally, even though v_b^2 will not be used in an acyclic evaluation, v_b^1 is still good. Hence, b remains in the framework. The weakly valid normal form of D is thus $D^{wv} = (\{b, c, d\}, \{C_b = d \vee (c \wedge \bot), C_c = \top, C_d = \top\})$. We can choose to view the condition for b as simply $C_b = d$. The resulting framework is quite straightforward and it is easy to see that its acyclic extensions coincide with the ones from D.

4.3.2 Relation Validity

In weak validity, we have focused on removing those arguments that could not be derived in an acyclic manner. However, just because an argument possesses a powerful (evidential) sequence or acyclic pd–evaluation, it does not mean that all of its supporters will be used to create such sequences (evaluations), as can be seen in the next example. Thus, the weak validity form is insufficient for ensuring the validity of support. In this section we will focus on the relation validity form, which removes undesirable support edges from the framework. As such, it is again a subframework of the original structure. Please note that in every framework, the relation valid form will be obtained slightly differently. Moreover, for now we will limit ourselves only to a possible translation analysis in the ADF case.

Example 51. Let us consider the EAS $(\{\eta, a\}, \emptyset, \{(\{\eta\}, a), (\{\eta, a\}, a)\})$ as depicted in Figure 39. We can see that the framework is weakly valid, however, it is easy to see that we could not obtain the e–support for a through the $(\{\eta, a\}, a)$ relation.

4.3.2.1 AFN Relation Validity

In AFNs, removing invalid support relations cannot be done from an arbitrary framework. This is due to the fact that if the argument is not valid in the first place due to e.g. a support cycle, removing this relation would change it into a valid argument, which is not a desirable behaviour:

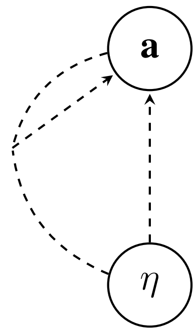


Figure 39: Strongly, but not relation valid EAS

Example 52. Assume a simple AFN $(\{a\}, \emptyset, \{(\{a\}, a)\})$ where a is a self–supporter. The $(\{a\}, a)$ support will not be used in any powerful sequence for a; in fact, no such sequence exists. However, when we remove this relation and create the framework $(\{a\}, \emptyset, \emptyset)$, a becomes a valid argument with a trivial powerful sequence (a) and will even appear in the grounded extension.

This brings us to the conclusion that we need to use weakly valid form as an intermediary step for the relation valid form:

Definition 4.24. Let $FN = (A, R, N)$ be an AFN and $FN^{wv} = (A', R', N')$ its weakly valid form. Let $coh(X)$ denote the set of all coherent subsets of $X \subseteq A'$. The trimmed subframework $FN^{rv} = (A', R', N'')$ of FN^{wv} is the **relation valid form** of FN, denoted $FN^{rv} \overset{rv}{\sqsubseteq} FN$, iff $N'' = \{(X \cap \bigcup coh(A' \setminus \{a\}) \mid (X, a) \in N'\}$.

From Theorem 4.18 we know that the extensions that are at least coherent coincide between FN and FN^{wv}. Consequently, for these semantics it suffices to perform the analysis for FN^{wv} and FN^{rv} in order to obtain the relation between FN and FN^{rv}.

Theorem 4.25. *Let $FN^{wv} = (A', R', N')$ be a weakly valid AFN and $FN^{rv} = (A', R', N'')$ its relation valid form. A set $E \subseteq A'$ is a σ–extension of FN^{wv}, where $\sigma \in \{$conflict–free, coherent, strongly coherent, admissible, preferred, complete, grounded, stable$\}$ iff it is a σ–extension of FN^{rv}.*

However, even though the conflict–free extensions of FN^{wv} and FN^{rv} coincide, it is not the case in FN and FN^{wv}. The removal of arguments means that two different conflict–free extensions of FN can be mapped to a single set in FN^{wv} and only one of them will be "retrieved". Thus, by combining Theorems 4.18 and 4.25, we obtain the following result:

Theorem 4.26. *Let $FN = (A, R, N)$ be an AFN and $FN^{rv} = (A', R', N'')$ its relation valid form. A set $E \subseteq A$ is a σ–extension of FN, where $\sigma \in \{$coherent, strongly coherent, admissible, preferred, complete, grounded, stable$\}$ iff it is a σ–extension of FN^{rv}. If $E \subseteq A$ is a conflict–free extension of FN, then $E \cap A'$ is conflict–free in FN^{rv}. If $E \subseteq A'$ is conflict–free in FN^{rv}, then it is conflict–free in FN.*

We can now put our results into the system. What we will do is break it into two steps; first, we will do the analysis assuming that the source frameworks are weakly valid, and then extend it to arbitrary frameworks.

Translation 10. *Let WV^{AFN} be the collection of all weakly valid AFNs and RV^{AFN} the collection of relation valid AFNs, both based on argument domain $\mathcal{U}$. The relation valid normal form from weakly valid form translation $rwv\text{-}Tr^{AFN} : WV^{AFN} \rightarrow RV^{AFN}$ is defined as $rwv\text{-}Tr^{AFN}((A, R, N)) = (A, R, N')$, where $N' = \{(X \cap \bigcup coh(A \setminus \{a\}) \mid (X, a) \in N\}$ and $coh(X)$ denotes the set of all coherent subsets of $X \subseteq A$.*

The semantics theorem relevant for this translation is simply Theorem 4.25:
Redefinition of Theorem 4.25: *Let $\sigma \in$ {conflict–free, coherent, strongly coherent, admissible, complete, preferred, grounded, stable} be a semantics and let SC^{Tr}_{σ} be the identity casting functions for σ. The translation $rwv\text{-}Tr^{AFN}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*
Analysis of Translation 10: Under the conflict–free, strongly coherent, coherent, admissible, preferred, complete, grounded and stable semantics and their removal casting functions, the translation $rwv\text{-}Tr^{AFN}$ is:

- source–subclass, target–subclass, overlapping
- argument domain preserving, argument set preserving, attack relation preserving and support relation removing
- generic and semantics domain preserving
- semantical and exact

Translation $rwv\text{-}Tr^{AFN}$ is not modular.

Explanation. We can observe that both WV^{AFN} and RV^{AFN} do not account for all possible AFNs. Consequently, the translation $rwv\text{-}Tr^{AFN}$ is source and target–subclass. Moreover, two weakly valid AFNs can be transformed into the same relation valid form. Consider the frameworks $FN_1 = (\{a, b, c\}, \emptyset, \{(\{a, c\}, c), (\{a, b, c\}, c)\})$ and $FN_2 = (\{a, b, c\}, \emptyset, \{(\{a, c\}, c), (\{a, b\}, c)\})$. We can observe that the supports sets in FN_1 and FN_2 are not the same. The relation valid form of both structures is $(\{a, b, c\}, \emptyset, \{(\{a\}, c), (\{a, b\}, c)\})$. Thus, we can conclude that $rwv\text{-}Tr^{AFN}$ is overlapping.

All the properties concerning how the structure of the source framework is modified can be observed clearly from Translation 10 itself. The approach is also clearly semantics domain preserving and generic. Due to the fact that coherence and validity of a support relation are semantical notions, we classify $rwv\text{-}Tr^{AFN}$ as semantical. The fact that it is exact under the listed semantics and identity casting functions follows from the (redefinition of) Theorem 4.25.

In order to show that the translation is not modular, let us consider the framework $FN = (\{a, b, c\}, \emptyset, \{(\{a\}, b), (\{a, c\}, b), (\{b\}, c)\})$ and two of its subframeworks $FN_1 = (\{a, b, c\}, \emptyset, \{(\{a\}, b), (\{b\}, c)\})$ and $FN_2 = (\{a, b, c\}, \emptyset, \{(\{a, c\}, b)\})$. We can observe that $FN_1 \cup FN_2 = FN$ and that FN is weakly, but not relation valid. Nevertheless, both of the subframeworks FN_1 and FN_2 are weakly and relation valid to start with. Consequently, they will not be affected by the translation at all. Thus, $rwv\text{-}Tr^{AFN}(FN_1) \cup rwv\text{-}Tr^{AFN}(FN_2) = FN$. Since FN is not in relation valid form (the $\{a, c\}$ support needs to be reduced to $\{a\}$), then clearly $FN \neq rwv\text{-}Tr^{AFN}(FN)$. Thus, our translation is not modular. ■

The version taking all types of AFNs into account is now:

Translation 11. *Let Fr^{AFN} be the collection of all AFNs and RV^{AFN} the collection of weakly valid AFNs, both based on argument domain $\mathcal{U}$. The weakly valid normal form translation $rv\text{-}Tr^{AFN} : Fr^{AFN} \rightarrow RV^{AFN}$ is defined as $rv\text{-}Tr^{AFN}((A, R, N)) = rwv\text{-}Tr^{AFN}(wv\text{-}Tr^{AFN}((A, R, N)))$.*

Redefinition of Theorem 4.26: *Let $\sigma \in \{$coherent, strongly coherent, admissible, complete, preferred, grounded, stable$\}$ be a semantics and let SC^{Tr}_{σ} be the identity casting functions for σ. The translation $rv\text{-}Tr^{AFN}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$. It is $\supseteq$–weak under conflict–free semantics and identity casting functions.*

Analysis of Translation 11: Under the conflict–free, strongly coherent, coherent, admissible, preferred, complete, grounded and stable semantics and their removal casting functions, the translation $rv\text{-}Tr^{AFN}$ is:

- full, target–subclass, overlapping
- argument domain preserving, argument removing, induced attack removing, and support relation removing
- semantics domain preserving
- semantical

Translation $rv\text{-}Tr^{AFN}$ is not modular. Under coherent, strongly coherent, admissible, preferred, complete, grounded and stable semantics $rv\text{-}Tr^{AFN}$ is exact and generic.

Explanation. The properties of our translation will be simply an outcome of the properties of $rwv\text{-}Tr^{AFN}$ and $wv\text{-}Tr^{AFN}$. The fact that the translation is full and target–subclass follows easily from the fact that $wv\text{-}Tr^{AFN}$ takes any framework as input and that $rwv\text{-}Tr^{AFN}$ returns relation valid structures. Since both of them are overlapping, then so is $rv\text{-}Tr^{AFN}$. Similar follows for the argument and semantics domain preserving properties. Now, as $rwv\text{-}Tr^{AFN}$ is argument preserving (attack relation preserving) and $wv\text{-}Tr^{AFN}$ is argument removing (induced attack removing), then $rv\text{-}Tr^{AFN}$ is argument removing (induced attack removing). In both of the translations we remove support relations, but only in one case this removal is induced; therefore, $rv\text{-}Tr^{AFN}$ is just support removing. The set of semantics on which both of the translations are generic is the same as in the $wv\text{-}Tr^{AFN}$ case. Consequently, $rv\text{-}Tr^{AFN}$ is also classified as generic. As $wv\text{-}Tr^{AFN}$ is not modular, then neither is $rv\text{-}Tr^{AFN}$. Finally, both $wv\text{-}Tr^{AFN}$ and $rwv\text{-}Tr^{AFN}$ are semantical and exact w.r.t. strongly coherent, coherent, admissible, preferred, complete, grounded and stable semantics. Therefore, these properties follow for $rv\text{-}Tr^{AFN}$ as well. ■

Example 53. Let us consider an AFN $(\{a, b, c\}, \emptyset, \{(\{a, b\}, a), (\{b, c\}, a)\})$, in which sets $\{a, b\}$ and $\{b, c\}$ support a. We can observe that the framework is weakly valid – for b and c we can consider trivial sequences (b) and (c), while for a, the sequence (b, a) is powerful. It is also worth noting that the framework is in minimal normal form. Despite weak validity, we can observe that e.g. the set $\{a, b\}$ is entirely useful; namely, we can never use a to derive a due to acyclicity requirements in the AFN semantics. Thus, the framework is not in relation validity form and by brining it to it, we obtain the structure $(\{a, b, c\}, \emptyset, \{(\{b\}, a), (\{b, c\}, a)\})$. It is easy to see that the powerful sequences created for the arguments would be exactly the same in the relation valid and in the original framework, thus leading to a correspondence between the source and target extensions. Finally, we can observe that the new structure is not in minimal form.

4.3.2.2 EAS Relation Validity

The weakly valid intermediary step is not required when we consider EASs. This is due to the fact that if an argument is not valid in the first place, removing a support relation is not going help it anyway – only by adding a relation from a valid argument we can change the status of this argument:

Example 54. We can consider a simple EAS $(\{a, \eta\}, \emptyset, \{(\{a\}, a)\})$, corresponding to the AFN from Example 52. It can be observe that a possesses no evidential sequence and that its self–supporting link will never be used. We can remove this relation and obtain the framework $(\{a, \eta\}, \emptyset, \emptyset)$. The argument a still does not possess an evidential sequence and the extensions remain unchanged.

Consequently, the relation valid translation for EASs is a single–step one, and preserves the conflict–free semantics as well as those that are self–supporting:

Definition 4.27. Let $ES = (A, R, E)$ be an EAS. The subframework $ES^{rv} = (A, R, E')$ is the **relation valid form** of ES, denoted $ES^{rv} \overset{rv}{\sqsubseteq} ES$, iff $E' = \{(X, a) \in E \mid$ there exists a self–supporting set $S \subseteq A \setminus \{a\}$ s.t. $X \subseteq S\}$.

Theorem 4.28. *Let $ES = (A, R, E)$ be an EAS and $ES^{rv} = (A, R, E')$ its relation valid form. A set $S \subseteq A$ is a σ–extension of ES, where $\sigma \in \{$conflict–free, self-supporting, strongly self-supporting, admissible, preferred, complete, grounded, stable$\}$ iff it is a σ–extension of ES^{rv}.*

The relation valid form put into our system is now as follows:

Translation 12. *Let Fr^{EAS} be the collection of all EASs and RV^{EAS} the collection of relation valid EASs, both based on argument domain $\mathcal{U}$. The relation valid normal form translation $rv\text{-}Tr^{EAS} : Fr^{EAS} \rightarrow RV^{EAS}$ is defined as $rv\text{-}Tr^{EAS}((A, R, E)) = (A, R, E')$, where $E' = \{(X, a) \in E \mid$ there exists a self–supporting set $S \subseteq A \setminus \{a\}$ s.t. $X \subseteq S\}$.*

Redefinition of Theorem 4.28: *Let $\sigma \in \{$conflict–free, self–supporting, strongly self–supporting, admissible, complete, preferred, grounded, stable$\}$ be a semantics and let SC^{Tr}_{σ} be the identity casting functions for σ. The translation $rv\text{-}Tr^{EAS}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*

Analysis of Translation 12: Under the conflict–free, self–supporting, strongly self–supporting, admissible, complete, preferred, grounded and stable semantics and identity casting functions, the translation $rv\text{-}Tr^{EAS}$ is:

- full, target–subclass, overlapping
- argument domain preserving, argument and attack preserving, support relation removing
- generic and semantics domain preserving
- semantical and exact

Translation $rv\text{-}Tr^{EAS}$ is not modular.

Explanation. We can repeat the analysis we have done in the case of AFNs (Translation 10) to explain the properties of $rv\text{-}Tr^{EAS}$. Thus, we will only discuss the modularity a little bit further. In order to show that the translation is not modular, we can reuse the example from the analysis of Translation 8. Consider a simple framework $ES = (\{\eta, a, b\}, \emptyset, \{(\{\eta\}, a), (\{a\}, b)\})$ and its two subframeworks $ES_1 = (\{\eta, a, b\}, \emptyset, \{(\{\eta\}, a)\})$ and $ES_2 = (\{\eta, a, b\}, \emptyset, \{(\{a\}, b)\})$. We can observe that $ES_1 \cup ES_2 = ES$. The framework ES is already in relation validity form and thus will not be affected by the translation. The relation validity forms of its two subframeworks are $ES_1^{rv} = (\{\eta, a, b\}, \emptyset, \{(\{\eta\}, a)\})$ and $ES_2^{rv} = (\{\eta, a, b\}, \emptyset, \emptyset)$ respectively. Their union is equal to ES_1^{rv}, which has different evidential support from ES. Therefore, our translation is not modular. Even if we considered only the weakly valid frameworks as input, we could still adapt the example given in the analysis of Translation 10 and again conclude that our approach is not modular. ■

Example 55. Let $(\{a, b, c, \eta\}, \emptyset, \{(\{\eta\}, a), (\{\eta, a\}, a), (\{b\}, a), (\{b, c\}, a), (\{\eta\}, b)\})$ be an EAS. We can observe that c does not possess an evidential sequence. Consequently, the $(\{b, c\}, a)$ support is not interesting. However, neither is $(\{\eta, a\}, a)$. By removing them we obtain the framework $(\{a, b, c, \eta\}, \emptyset, \{(\{\eta\}, a), (\{b\}, a), (\{\eta\}, b)\})$. Since the links that we have removed are those that were not used in the construction of an evidential sequence, it is easy to see that the sequences themselves remain unchanged.

4.3.2.3 ADF Relation Validity

Finally, we come to ADFs. In this case, we cannot yet give a fully working translation, but will provide an analysis showing where the challenges are. In ADFs, we deal with links that do not necessarily fall into the "supporting" and "attacking" categories; when they actually do, we work with the bipolar ADFs, which are only a subclass of the general ones. Moreover, when we are dealing with AFNs and EASs, relations of different types are clearly separate in the structure and removing one does not affect the other. For example, if we wanted to remove an attack from a to b, the support from a to b would remain unaffected. As already discussed in Section 2.3.9, ADFs take a more general view on the acceptance of an argument and thus avoid this separation. For these two reasons, the relation validity form for ADFs is not as straightforward as in other frameworks. We need to identify which occurrences and of what arguments need to be removed and how to proceed with the removal.

Let us start with the description of a relation valid form. First of all, we can adapt the definition known from AFNs and EASs. Thus, we would say that $D = (A, L, C)$ is in relation valid form iff for every argument $a \in A$ and every $S \subseteq par(a)$ s.t. $C_a(S) = in$, there exists a pd–acyclic set $E \subseteq A \setminus \{a\}$ s.t. $S \subseteq E$. However, as we can see in the example below, this approach is not entirely adequate:

Example 56. Let us consider a simple ADF $D = (\{a, b, c\}, \{C_a = \top, C_b = a \vee \neg c, C_c = b\})$. The functional form of C_b is $\{\emptyset : in, \{a\} : in, \{c\} : out, \{a, c\} : in\}$. All of the arguments possess acyclic pd–evaluations: $((a), \emptyset)$ for a, $((a, b), \emptyset)$ and $((b), \{c\})$ for b and $((a, b, c), \emptyset)$ and $((b, c), \{c\})$ for c. Thus, we can observe that every minimal decisively in interpretation for an argument is used in some evaluation. However, the framework is not in a relation valid form if we take the AFN (EAS) approach. The set $\{a, c\}$, which satisfies the condition of b, is not included in any pd–acyclic set that does not contain b. This is not an answer we want. In our opinion, b is more a self–attacker (i.e. supports c that in turn can attack b) rather than self–supporter, and the AFN (EAS) formulation does not behave in the intended way when we deal with overpowering support, which as such is not present in those frameworks.

Let us now consider the other option, already mentioned in the example, that is focused on minimal decisively in interpretations. However, again we can observe that it has its drawbacks:

Example 57. Consider a simple framework $(\{a, b\}, \{C_a = \top, C_b = a \vee (a \wedge b)\})$. The functional representations of the conditions would be $C_a = \{\emptyset : in\}$ and $C_b = \{\emptyset : out, \{a\} : in, \{b\} : out, \{a, b\} : in\}$. We can also read the structure as an EAS, where a is playing the role of η. The (b, b) link is not a desirable one – although this self–support is actually redundant, we would prefer it to be removed just like in any other framework. However, b does not affect the outcome of the acceptance condition of b (see Section 4.1.5) and thus will never show up in a minimal decisively in interpretation for b. Consequently, the need for its removal will not be detected.

The cause of the issue described in the example is the minimality of the interpretations. However, without it we basically come back to the AFN (EAS) approach, since for every decisively in interpretation v, $v^{\mathbf{t}}$ satisfies the acceptance condition of a given argument. Therefore, we decide to limit ourselves to redundancy–free ADFs, as it makes sure that every parent of a given argument will show up in an interpretation. Although the redundancy–free form would also remove redundant attacking arguments (see e.g. SETAF in Section 4.1.1), we will accept this solution for now. We thus obtain the following definition of relation validity:

Definition 4.29. Let $D = (A, L, C)$ be a redundancy–free ADF. We say that D is in the **relation valid form** iff every minimal decisively in interpretation of an argument $a \in A$ can be used in constructing an acyclic pd–evaluation for a.

The main issue why we we do not provide means to translate an ADF into a relation valid forms concerns detecting which occurrences of given arguments to be removed from an acceptance condition and how to proceed with the removal. Let us look at the following example:

Example 58. We will consider two simple ADFs: $D_1 = (\{a, b\}, \{C_a = \top, C_b = a \vee b\})$ and $D_2 = (\{a, b, c\}, \{C_a = \top, C_b = a \veebar c, C_c = b\})$[16].

Let us start with D_1. The functional representation of C_b is $\{\emptyset : out, \{a\} : in, \{b\} : in, \{a, b\} : in\}$. We can observe that b has a decisively in interpretation $v_1 = \{b : \mathbf{t}\}$ that does not meet the relation validity requirements. In order to make v_1 not decisively in, we can consider two options; remove all of the occurrences of b from C_b or replace the assignment of any set containing b with out. In the first case, we receive $C_b' = \{\emptyset : out, \{a\} : in\}$ (equivalent to $C_b' = a$), which is the desired outcome. In other words, the reduct of the condition w.r.t. arguments that appear in decisively in interpretations satisfying relation validity conditions produced a correct new condition. Let us now consider the value replacement; we obtain the condition $C_b'' = \{\emptyset : out, \{a\} : in, \{b\} : out, \{a, b\} : out\}$ (equivalent to $C_b'' = a \wedge \neg b$). Although the new decisively in interpretations satisfy validity requirements, clearly the meaning of the condition has changed. The argument b, that previously could be accepted in an extension, is now a self–attacker. Therefore, the value replacement approach is not suitable for this framework.

Let us now consider D_2. The functional representation of C_b is $\{\emptyset : out, \{a\} : in, \{c\} : in, \{a, c\} : out\}$. Argument b has two minimal decisively in interpretations, $v_1 = \{a : \mathbf{t}, c : \mathbf{f}\}$ and $v_2 = \{a : \mathbf{f} : c : \mathbf{t}\}$. We can observe that v_2 does not satisfy our validity requirements and needs to be taken care of. Let us first consider the reduct method, which worked in the previous framework. We obtain the condition $C_b' = \{\emptyset : out, \{a\} : in\}$ (or equivalently, $C_b' = a$). This is not a correct modification; we can observe that while $\{a, b\}$ was not aa–admissible in D_2, it is aa–admissible in the modified form $D_2 = (\{a, b, c\}, \{C_a = \top, C_b = a, C_c = b\})$. This is due to the fact that

[16] $a \veebar c$ stands for the XOR of a and c and is equivalent to $(a \vee c) \wedge (\neg a \vee \neg c)$ and $(a \wedge \neg c) \vee (\neg a \wedge c)$.

c, even though assigned $\mathbf{f}$, still occurs in v_1. Therefore, not all of its occurrences are undesirable. However, we cannot just remove the $\{c\}$ from the condition, as then it would no longer be a total function. Therefore, we can consider changing its assignment to out. The new condition is thus $C''_b = \{\emptyset : out, \{a\} : in, \{c\} : out, \{a, c\} : out\}$ (equivalent to $C''_b = a \land \neg c$). We can observe that v_2 no longer appears as a decisively in interpretation and that v_1 is preserved. The modified framework will produce desirable extensions.

This example brings us to the conclusion that there is more than one way to adapt a condition to the relation valid form and that not all methods are suitable in every possible scenario. We thus leave obtaining a relation valid form translation and fine–tuning its definition as an open issue for the future work. However, please note that just like in EASs, we believe the translation will not have the weakly valid form prerequisite that is present in AFNs. A cyclic argument that loses all of its support can receive a falsum acceptance condition, which as we will explain in Section 4.2, does not affect the aa–semantics family.

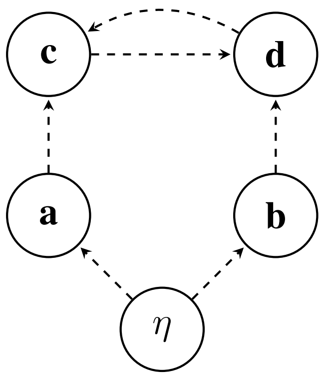
4.3.3 Strong Validity

We now reach the most restrictive of all validity forms, the strong validity. Unfortunately, we do not have translations to this form yet, though we believe that they exist (see Example 61). Consequently, we leave it to the reader to reclassify it as a subclass when required.

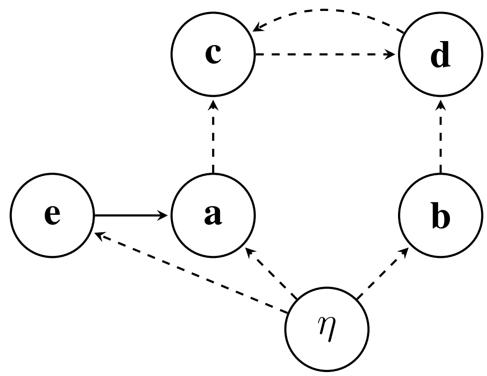
The point of strong validity is to reach a representation of the framework which, from the support perspective, is free from any "impurities", and in which the powerful (evidential, pd–acyclic) verification step is no longer required. Therefore, not only all arguments and relations need to be valid, but also all the possible "paths" that the support relation might create. In other words, if we have a set of arguments S s.t. every $a \in S$ is sufficiently supported through the N relation in an AFN (or the E relation in an EAS, or satisfies the condition of a in an ADF), then the set is coherent (self–supporting or pd–acyclic). Let us show that weak and relation validity forms do not remove all the support cycles from the framework:

Example 59. Let us consider the EAS $ES_1 = (\{\eta, a, b, c, d\}, \emptyset, \{(\{\eta\}, a), (\{\eta\}, b), (\{a\}, c), (\{b\}, d), (\{d\}, c), (\{c\}, d)\})$ as depicted in Figure 40a. All arguments possess an evidential sequence and the framework is weakly valid. Moreover, every support can be used to construct an evidential sequence for an argument. However, even though set $\{c, d\}$ supports a and b through the support relation, it is clearly not self–supporting. Thus, the framework is not strongly valid.

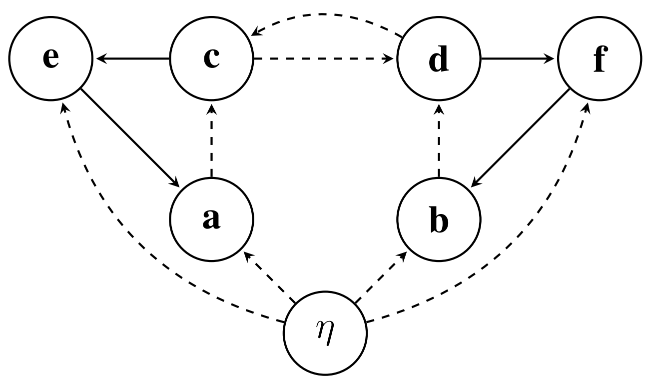
To such support cycles that still occur despite the weakly and relation valid forms we will refer to as optional or hidden cycles. In other words, a given argument can be derived both in acyclic and not acyclic manner. This situation cannot be simply fixed by removing some support edges. The point is that all of those supports lead to powerful sequences (or similar constructions) with which an argument can be derived. Consequently, removing such links is not without effect on the semantics:



(a) ES_1



(b) ES_2



(c) ES_3

Figure 40: Weakly and relation valid, but not strongly valid EASs

Example 60. In the framework $ES_1 = (\{\eta, a, b, c, d\},\ \emptyset,\ \{(\{\eta\}, a), (\{\eta\}, b), (\{a\}, c), (\{b\}, d), (\{d\}, c), (\{c\}, d)\})$ from Figure 40a, removing the $(\{d\}, c)$ and $(\{c\}, d)$ supports would not change the extensions of ES_1 under any standard semantics.

However, let us consider a slight modification of ES_1, the framework $ES_2 = (\{\eta, a, b, c, d, e\},\ \{(\{e\}, a)\},\ \{(\{\eta\}, a),\ (\{\eta\}, b),\ (\{\eta\}, e),\ (\{a\}, c),\ (\{b\}, d),\ (\{d\}, c), (\{c\}, d)\})$ depicted in Figure 40b. Removing the $(\{d\}, c)$ link would mean that all coherent sets of c are attacked by $\{\eta, e\}$ and that the argument will not appear in any admissible extension, which is not the case in ES_2.

Let us now go to the framework $ES_3 = (\{\eta, a, b, c, d, e, f\},\ \{(\{e\}, a),\ (\{f\}, b), (\{c\}, e),\ (\{d\}, f)\},\ \{(\{\eta\}, a),\ (\{\eta\}, b),\ (\{\eta\}, e),\ (\{\eta\}, f),\ (\{a\}, c),\ (\{b\}, d),\ (\{d\}, c), (\{c\}, d)\})$ depicted in Figure 40c. We can observe that the sets $\{\eta, e\}$ and $\{\eta, f\}$ are not admissible, even though $\{\eta, a, f\}$ is. Let us now remove $(\{d\}, c)$ and $(\{c\}, d)$. This means that the sets $\{\eta, b, c, d\}$ and $\{\eta, a, c, d\}$ are no longer self–supporting. As a result, the sets $\{\eta, e\}$ and $\{\eta, f\}$ now become admissible, which is not the desired behavior.

Therefore, as we could have observed in the example, simply removing the support links taking part in an optional cycle can "break" the behavior of the framework. Furthermore, such optional cycles can even exist between arguments that appear in the grounded extension, as was the case in e.g. ES_1. The approach we want to pursue in the future will focus on argument duplication. This means that a given argument can receive a number of additional representations s.t. the obtained structure is in fact free from support cycles. The construction resembles trying to build a support relation tree rooted at η, where we introduce a duplicate argument whenever we come back to an element that has already appeared in the structure. The duplicate also copies all the attacks carried out or received by the original argument:

Example 61. We will now come back to the frameworks from Example 60 and show their possible strongly valid forms. Let us start with $ES_1 = (\{\eta, a, b, c, d\}, \emptyset, \{(\{\eta\}, a), (\{\eta\}, b), (\{a\}, c), (\{b\}, d), (\{d\}, c), (\{c\}, d)\})$ from Figure 40a. The arguments c and d took part in an optional cycle through the $(\{d\}, c)$ and $(\{c\}, d)$ support links. We thus duplicate the arguments c and d and let them take over the problematic edges. This brings us to the framework $ES_1^{st} = (\{\eta, a, b, c, d, c', d'\}, \emptyset, \{(\{\eta\}, a), (\{\eta\}, b), (\{a\}, c), (\{b\}, d), (\{d\}, c'), (\{c'\}, d)\})$ from Figure 41a. We can observe that the resulting structure is strongly valid. Moreover, replacing the occurrences of duplicate arguments with their original ones in the extensions retrieves the answers of ES_1.

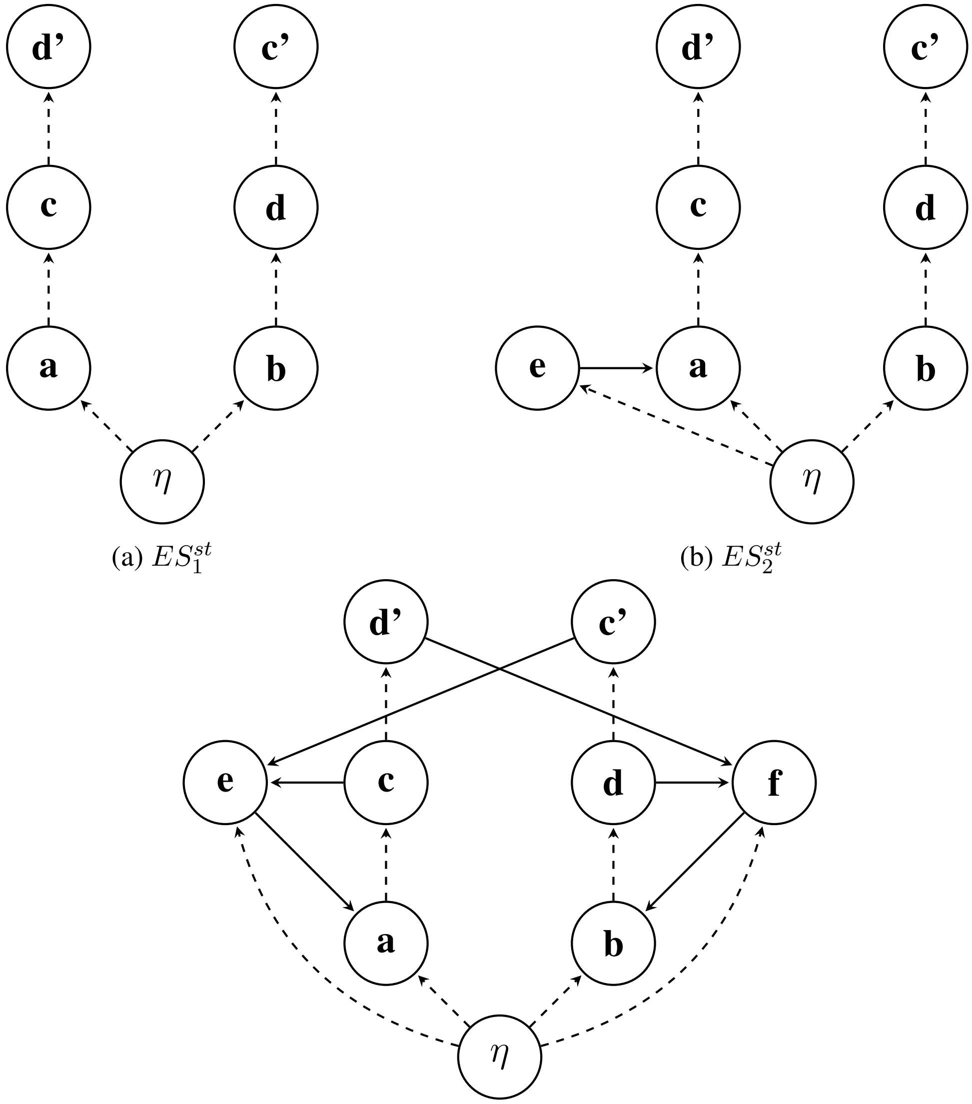
Let $ES_2 = (\{\eta, a, b, c, d, e, c', d'\}, \{(\{e\}, a)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, e), (\{a\}, c), (\{b\}, d), (\{d\}, c), (\{c\}, d)\})$ be the framework depicted in Figure 40b. Its strongly valid form is $ES_2^{st} = (\{\eta, a, b, c, d, e\}, \{(\{e\}, a)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, e), (\{a\}, c), (\{b\}, d), (\{d\}, c'), (\{c'\}, d)\})$, visible in Figure 41b. In ES_2, $\{\eta, b, d, c\}$ was an admissible extension. It is no longer the case in ES_2^{st}, since all coherent sets of c are attacked by $\{\eta, c\}$. However, $\{\eta, b, d, c'\}$ is admissible in ES_2^{st} and by replacing c' with the original c, we retrieve the ES_2 answer.

We end this example with the analysis of the EAS $ES_3 = (\{\eta, a, b, c, d, e, f\}, \{(\{e\}, a), (\{f\}, b), (\{c\}, e), (\{d\}, f)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, e), (\{\eta\}, f), (\{a\}, c), (\{b\}, d), (\{d\}, c), (\{c\}, d)\})$ from Figure 40c. Its strong validity from is the framework $ES_3^{st} = (\{\eta, a, b, c, d, e, f, c', d'\}, \{(\{e\}, a), (\{f\}, b), (\{c\}, e), (\{d\}, f), (\{c'\}, e), (\{d'\}, f)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, e), (\{\eta\}, f), (\{a\}, c), (\{b\}, d), (\{d'\}, c), (\{c'\}, d)\})$ that can be seen in Figure 41c. The sets $\{\eta, e\}$ and $\{\eta, f\}$ are not admissible in ES_3. They are also not admissible in ES_3^{st}, as the first set cannot do anything about the attack from c', the other from d'. The set $\{\eta, a, f\}$ is admissible in both frameworks – f deals with d and c', while e with a and d'. Therefore, again the transformation exhibits desired behavior.

Although we have focused on EASs in this analysis, it can be easily adapted to AFNs and ADFs as well. We are now ready to introduce the strongly valid normal forms for our frameworks and analyze their properties.

4.3.3.1 AFN Strong Validity

In the previous section, we could have observed that relation validity did not necessarily mean that there are no support cycles in the framework. This means that that checking whether every set of arguments sufficiently supporting a given argument (i.e. one that has



(a) ES_1^{st} (b) ES_2^{st}

(c) ES_3^{st}

Figure 41: Strongly valid EASs

an element in common with every supporting set in N) can be used in the construction of a powerful sequence is not enough. Therefore, we strengthen this approach by requiring that this restriction holds not just for one argument at a time, but for all of them simultaneously:

Definition 4.30. Let $FN = (A, R, N)$ be an AFN. With $sup(a) = \bigcup_{C \subseteq A, CNa} C$ we will denote all arguments supporting a and by $suf(a) = \{S \mid S \subseteq sup(a)$ and $\forall C \subseteq$ s.t. CNa, $C \cap S \neq \emptyset\}$ all subsets of $sup(a)$ that have an element in common with every support set of a. The framework FN is **strongly valid** iff for every function $f : A \rightarrow \{S \mid a \in A, S \in suf(a)\}$ we can create a powerful sequence $(a_0, ..., a_n)$ consisting of all elements of A s.t. $f(a_0) = \emptyset$ and $f(a_i) \subseteq \{a_0, ..., a_{i-1}\}$ for $i > 0$.

As expected, a framework meeting the strong validity requirements is also weakly and relation valid. Moreover, for frameworks in this form, it suffices to focus simply on sets in which all arguments are supported, without being forced to do any analysis on the validity of the support in order to ensure coherence:

Theorem 4.31. *Let $FN = (A, R, N)$ be an AFN. If FN is strongly valid, then it is weakly and relation valid.*

Theorem 4.32. *Let $FN = (A, R, N)$ be a strongly valid AFN. A set of arguments $E \subseteq A$ is coherent iff for every argument $a \in E$ and set $C \subseteq A$ s.t. CNa, $C \cap E \neq \emptyset$.*

We can observe that the support function f is in fact not much different from the notion of a pd–function in ADFs and the construction of the powerful sequence with it is similar to the creation of an acyclic evaluation. Hence, we can observe that this definition was driven by our understanding of a strongly valid framework from the perspective of ADFs. It will become more useful when comparing AFNs and ADFs. Nevertheless, this approach is somewhat semantical and says little about the structural aspects of the framework in question. We will now address this issue and propose an alternative way of looking at the strongly valid frameworks. We can observe that for strongly valid frameworks, we can create an ordering of the arguments s.t. independently of the used support function, the ordering will form a powerful sequence:

Theorem 4.33. *Let $FN = (A, R, N)$ be an AFN. Let $sup(a) = \bigcup_{C \subseteq A, CNa} C$ denote all arguments supporting a and $suf(a) = \{S \mid S \subseteq sup(a)$ and $\forall C \subseteq$ s.t. CNa, $C \cap S \neq \emptyset\}$ stand for all subsets of $sup(a)$ that have an element in common with every support set of a. FN is strongly valid iff there exists a sequence $(a_0, ..., a_n)$ of all arguments in A s.t. given any function $f : A \to \{S \mid a \in A, S \in suf(a)\}$, $(a_0, ..., a_n)$ is a powerful sequence s.t. $f(a_0) = \emptyset$ and $f(a_i) \subseteq \{a_0, ..., a_{i-1}\}$ for $i > 0$.*

This "ultimate" sequence of arguments strongly resembles the idea of a topological ordering for directed graphs. We can induce such graphs using the support relation and using the previous theorems and their proofs, we can show that our sequence is indeed a topological ordering for it. This means that the following property is true:

Theorem 4.34. *Let $FN = (A, R, N)$ be an AFN and $SG^{FN} = (A, N')$, where $N' = \{(a, b) \mid \exists E \subseteq A, a \in E$ s.t. $ENb\}$, the support graph induced by FN. FN is strongly valid iff SG^{FN} is a directed acyclic graph.*

Hence, we have shown that what we understand by strong validity indeed coincides with the directed acyclic graphs in the case of AFNs and thus gave both a semantical and structural interpretation of this form.

Example 62. Let us consider the framework $(\{a, b, c, d, e, f\}$, $\{(a, e)$, (d, b), (e, c), $(f, d)\}$, $\{(\{b, c\}, a)$, $(\{f\}, f)\})$, previously analyzed in Examples 16 and 48. We can observe that the framework is not strongly valid; in particular, the only possible support assignment for f is $\{f\}$ and clearly we cannot create a powerful sequence with it. Similarly, the support graph $(\{a,b,c,d,e,f\}, \{(\{b\},a), (\{c\},a), (\{f\},f)\})$ is not acyclic.

Let us now consider the framework $(\{a,b,d,e,f,g\}, \{(b,a), (b,c), (c,b), (c,d), (d,f), (f,f)\}, \{(\{a,g\},c), (\{d\},e), (\{e\},f), (\{g\},d), (\{g\},f)\})$ depicted in Figure 42. We have two support functions, one assigning $\{g\}$ to c and one assigning $\{a\}$ to c; all other assignments stay the same and are as follows. For a, b and g we have simply $\emptyset$, for d we have $\{g\}$, for e we have $\{d\}$, and for f we have $\{e,g\}$. We can observe that independently of the support function, the sequence (a,b,g,c,d,e,f) will be a powerful sequence. Consequently, our framework is strongly valid. The support graph $(\{a,b,c,d,e,f,g\}, \{(a,c),(d,e),(e,f),(g,c),(g,d),(g,f)\})$ is also directed acyclic – however, we can see that if we ignore the direction of the edges, acyclicity is lost.

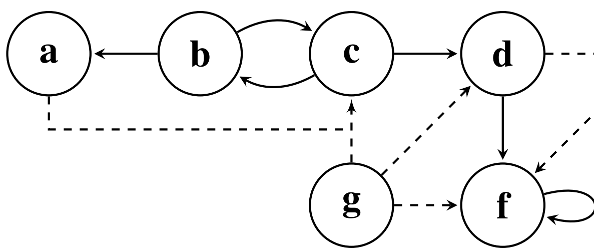


Figure 42: Sample AFN

4.3.3.2 EAS Strong Validity

Let us now look at the strongly valid normal form in the context of EASs. The first approach is in the same pd–function style as Definition 4.30. The construction is somewhat simpler due to the differences between how group relations are interpreted in the evidential support relation E and necessary support N.

Definition 4.35. Let $ES = (A, R, E)$ be an EAS. The framework ES is **strongly valid** iff for every function $f : A \to \{S \mid a \in A, S \subseteq A, SEa\}$ we can create an evidential sequence $(a_0, ..., a_n)$ consisting of all elements of A s.t. $f(a_0) = \emptyset$ and for $i > 0$ and $f(a_i) \subseteq \{a_0, ..., a_{i-1}\}$.

Just like in the case of AFNs, strongly valid frameworks are also weakly and relation valid. Furthermore, we can replace the self–supporting extensions by sets that simply support their members, i.e. e–support checks become unnecessary.

Theorem 4.36. *Let $ES = (A, R, E)$ be an EAS. If ES is strongly valid, then it is weakly and relation valid.*

Theorem 4.37. *Let $ES = (A, R, E)$ be a strongly valid EAS. A set of arguments $S \subseteq A$ is self–supporting iff for every argument $a \in S$ there is a set $S' \subseteq S$ s.t. $S'Ea$.*

Similarly as strongly valid AFNs, strongly valid EASs possess (at least one) special sequence of arguments s.t. independently of the chosen support function, it meets the requirements of an evidential sequence. Consequently, a strongly valid EAS induced a directed acyclic graph as well. However, we can recall that there was one significant difference between powerful and evidential sequences, namely that the latter enforced the existence of a supporting set for every non–η argument. This means that the induced support graph is in fact rooted, with η being the obvious root. Therefore, the "structural" aspects of a strongly valid EAS are not exactly the same as in AFNs:

Theorem 4.38. *Let $ES = (A, R, N)$ be an EAS. ES is strongly valid iff there exists a sequence $(a_0, ..., a_n)$ of all arguments in A s.t. given any function $f : A \rightarrow \{S \mid a \in A, S \subseteq A, SEa\}$, $(a_0, ..., a_n)$ is an evidential sequence s.t. $f(a_0) = \emptyset$ and for $i > 0$ and $f(a_i) \subseteq \{a_0, ..., a_{i-1}\}$.*

Theorem 4.39. *Let $ES = (A, R, E)$ be an EAS s.t. $A \neq \emptyset$ and $SG^{ES} = (A, E')$, where $E' = \{(a, b) \mid \exists X \subseteq A, a \in X$ s.t. $XEb\}$, the support graph induced by ES. ES is strongly valid iff SG^{ES} is a rooted directed acyclic graph s.t. η is the root.*

Example 63. In Example 59 we have considered the EAS $(\{\eta, a, b, c, d\}, \emptyset, \{(\{\eta\}, a), (\{\eta\}, b), (\{a\}, c), (\{b\}, d), (\{d\}, c), (\{c\}, d)\})$ and concluded it does not meet our strong validity intuitions. Let us now describe this in terms of support functions. Independently of the function, for η we have $\emptyset$ and for a and b we have $\{\eta\}$. It is only the assignments for c and d that differ. We can first assign $\{a\}$ to c and $\{b\}$ to d; under these assumptions, the sequence (η, a, b, c, d) is an evidential sequence. The same holds if we assign $\{a\}$ to c and $\{c\}$ to d. If we choose to assign $\{d\}$ to c and $\{b\}$ to d, then (η, a, b, d, c) is an evidential sequence. However, we finally come to the function that associates c with $\{d\}$ and d with $\{c\}$. We can observe that independently of how we order the arguments, we will not obtain an evidential sequence. Hence, the framework is not strongly valid. If we look at the support graph $(\{\eta, a, b, c, d\}, \{(\eta, a), (\eta, b), (a, c), (b, d), (d, c), (c, d)\})$, we can also see that is not directed acyclic.

Let us now consider the EAS $(\{\eta, a, b, c, d, e, f\}, \{(\{b\}, a), (\{b\}, c), (\{c\}, b), (\{c\}, d), (\{d\}, f), (\{f\}, f)\}, \{(\{\eta\}, b), (\{\eta\}, c), (\{\eta\}, d), (\{\eta\}, f), (\{d\}, e)\})$ previously described in Example 17 and for convenience again depicted in Figure 43. We can observe it is not strongly valid; although we do have support cycles as such, argument a does not posses an evidential sequence at all. Hence, independently of the chosen support function, we will never be able to order the arguments into an evidential sequence. If we look at the support graph $(\{\eta, a, b, c, d, e, f\}, \{(\eta, b), (\eta, c), (\eta, d), (\eta, f), (d, e)\})$, we can observe it is directed acyclic; however, it is not rooted at η.

We can now modify this framework by removing argument a and the associated attack. We have a single support function assigning $\{\eta\}$ to b, c, d and f and $\{d\}$ to e. The

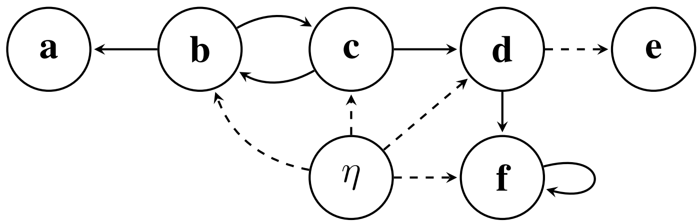


Figure 43: Sample EAS

framework is trivially strongly valid; we can observe that the associated support graph $(\{\eta, b, c, d, e, f\}, \{(\eta, b), (\eta, c), (\eta, d), (\eta, f), (d, e)\})$ is both rooted and directed acyclic.

4.3.3.3 ADF Strong Validity

Finally, we come to the strongly valid ADFs. To put it simply, a framework is strongly valid if every possible pd–function leads to an acyclic pd–evaluation covering all of the arguments:

Definition 4.40. Let $D = (A, L, C)$ be a redundancy–free ADF. The framework D is **strongly valid** iff every pd–function pd_A^D on A is sound and the set A can be ordered into a pd–sequence of an acyclic pd–evaluation on A w.r.t. pd_A^D.

Similarly as in AFNs and EASs, strong validity implies weak and relation validity. Verifying whether a set is pd–acyclic is reduced to checking whether there exists a (minimal) decisively in interpretation s.t. its positive part is contained in the set.

Theorem 4.41. *Let $D = (A, L, C)$ be an ADF. If D is strongly valid, then it is weakly and relation valid.*

Theorem 4.42. *Let $D = (A, L, C)$ be a strongly valid ADF. A set of arguments $E \subseteq A$ is pd–acyclic iff for every argument $a \in E$ there exists a minimal decisively in interpretation v_a s.t. $v_a^{\mathbf{t}} \subseteq E$.*

We can observe that the concept of the strongly valid form in ADFs is actually quite close to the AADF$^+$ subclass, which has been introduced in [75] (see Section 2.3.8). However, strong validity is in fact more restrictive, as it enforces the redundancy–freeness and the fact that every argument has to have an evaluation, which is not the case in AADF$^+$.

Theorem 4.43. *Let $D = (A, L, C)$ be an ADF. If D is strongly valid, then it is an AADF$^+$. If D is a redundancy–free cleansed AADF$^+$, then it is strongly valid.*

From this relation we can also make an observation that may be useful in the future. Although the strong validity translation we hope to create would as such be focused on preserving the aa–family of semantics, the relation of this normal form to the AADF$^+$

subclass means that strong validity is also valuable for other types. Of course, the translation for e.g. the cc–semantics would be different from the one for the aa–semantics. Nevertheless, the strength of this form makes it useful also for semantics that do not deal with support cycles.

Let us now focus on the alternative approach towards strong validity. The fact that AFNs and EASs that were in strongly valid forms induced acyclic support graphs was not particularly surprising. This method may also be simpler when compared to Definitions 4.30 and 4.35. However, creating and interpreting the support graph in ADFs is a little bit more problematic than in these frameworks, which makes the structural interpretation of this form somewhat less obvious.

A positive dependency graph can be induced from the positive parts of the minimal decisively in interpretations appearing in the framework. However, just because an argument appears in the $\mathbf{t}$–assignments of a decisively in interpretation, it does not mean that the associated link is supporting in the context of Definition 2.114. In fact, it might be neither supporting nor attacking, as can happen if we e.g. analyze a XOR or a XNOR condition. Moreover, there is also the issue of redundant links. The sources of these links do not have to appear in the minimal decisively in interpretations. However, by definition, redundant links are those that are both supporting and attacking. Therefore, there might be a link that is considered supporting based on the Definition 2.114, but does not appear in the positive dependency graph. Finally, in order to be able to induce the graph, we need to deal with the arguments that do not possess a decisively in interpretation in the first place. This means that we either turn them into "self–supporters" in the positive dependency graph in order to make a topological ordering impossible, or we limit ourselves to ADFs in cleansed forms where this issue does not occur.

We can observe that some of these difficulties are the ones that caused us to draw a line between AADF$^+$s and strongly valid ADFs. Bearing in mind the discussion we have carried out in Section 4.3.2.3 and the nature of the falsum arguments, we choose to define the positive dependency graph only for the redundancy–free and cleansed ADF for the time being. We choose to interpret the edges in this graph as representing the fact that a given argument is, in some situation, necessary for the acceptance of another argument. Therefore, the following version of the strong validity form can be introduced for ADFs:

Theorem 4.44. *Let $D = (A, L, C)$ be a redundancy–free ADF. D is strongly valid iff there exists a sequence $(a_0, ..., a_n)$ of all arguments in A s.t. given any pd–function pd on A, $((a_0, ..., a_n), \bigcup_{i=0}^{n} pd(a_i))$ is an acyclic pd–evaluation.*

Theorem 4.45. *Let $D = (A, L, C)$ be a redundancy–free and cleansed ADF and $PDG^D = (A, L')$, where $L' = \{(a, b) \mid \exists v \in min\_dec(in, b) \text{ s.t. } a \in v^{\mathbf{t}}\}$, its associated positive dependency graph. D is strongly valid iff PDG^D is a directed acyclic graph.*

With this example we close our section on strong validity in frameworks with support.

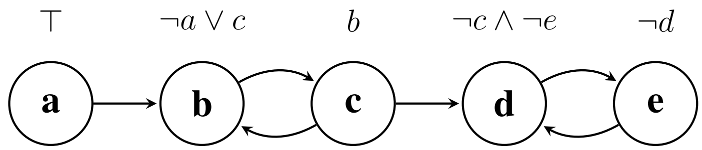


Figure 44: Sample ADF

Example 64. Let $(\{a, b, c, d, e\}, \{C_a = \top, C_b = \neg a \vee c, C_c = b, C_d = \neg c \wedge \neg e, C_e = \neg d\})$ be the ADF briefly described in Example 19 and for convenience again depicted in Figure 44. The minimal decisively in interpretations for our arguments are as follows: $v_a = \emptyset$ for a, $v_b^1 = \{a : \mathbf{f}\}$ and $v_b^2 = \{c : \mathbf{t}\}$ for b, $v_c = \{b : \mathbf{t}\}$ for c, $v_d = \{c : \mathbf{f}, e : \mathbf{f}\}$ for d and $v_e = \{d : \mathbf{f}\}$ for e. We can therefore create two pd–functions for this framework differing only by the interpretation assigned to b. If we chose v_b^1, we can observe that the sequence (a, b, c, d, e) is a part of an acyclic pd–evaluation $((a, b, c, d, e), \{a, c, d, e\})$. However, if we pick v_b^2, then due to the fact that the interpretation for b requires c and the one for c requires b, it is not possible to create an acyclic pd–evaluation that would have all of our arguments in its pd–sequence. Hence, our framework is not strongly valid. The positive dependency graph for this framework is $(\{a, b, c, d, e\}, \{(c, b), (b, c)\})$ and is clearly not directed acyclic.

Let us now analyze the ADF version of the AFN described in Example 62, i.e. the framework $(\{a, b, c, d, e, f, g\}, \{C_a = \neg b, C_b = \neg c, C_c = \neg b \wedge (a \vee g), C_d = \neg c \wedge g, C_e = d, C_f = \neg d \wedge \neg f \wedge e \wedge g, C_g = \top\})$. The decisively in interpretation for our arguments are the following: $v_a = \{b : \mathbf{f}\}$ for a, $v_b = \{c : \mathbf{f}\}$ for b, $v_c^1 = \{b : \mathbf{f}, a : \mathbf{t}\}$ and $v_c^2 = \{b : \mathbf{f}, g : \mathbf{t}\}$ for c, $v_d = \{c : \mathbf{f}, g : \mathbf{t}\}$ for d, $v_e = \{d : \mathbf{t}\}$ for e, $v_f = \{d : \mathbf{f}, f : \mathbf{f}, e : \mathbf{t}, g : \mathbf{t}\}$ for f and $v_g = \emptyset$ for g. We can therefore construct two sound pd–function on our set of arguments, one using v_c^1 and the other using v_c^2. However, independently of that, the sequence (g, a, b, c, d, e, f) is a part of an acyclic pd–evaluation (which in both cases is $((g, a, b, c, d, e, f), \{b, c, d, f\})$). Hence, our framework is strongly valid. We can observe that the positive dependency graph is $(\{a, b, c, d, e, f, g\}, \{(a, c), (g, c), (g, d), (d, e), (e, f), (g, f)\})$ and it is directed acyclic.

Let $(\{a, b, c\}, \{C_a = (b \wedge c) \vee c, C_b = a \vee \neg a, C_c = \top\})$ be the ADF previously described in Example 46 depicted in Figure 34a. We can observe that this framework is an AADF$^+$; the minimal decisively in interpretations for b and c are empty and the one for a is simply $v_a = \{c : \mathbf{t}\}$. Hence, any standard pd–evaluation can be made acyclic. However, both (a, b) and (b, a) are redundant connections, which means that our framework is not strongly valid. Only by bringing it to the redundancy–free form $(\{a, b, c\}, \{C_a = c, C_b = \top, C_c = \top\})$, we obtain validity.

4.4 Consistency Form

The consistency normal form will be especially relevant in the case of translations from AFNs and EASs into ADFs. Basically speaking, this form needs to be considered whenever we decide to transform a bipolar framework into an ADF. As already explained in Section 2.3.9, not ensuring that a framework is in the consistency form can cause the resulting ADF to produce undesirable extensions. Due to the relation of defense attack to overpowering support in ADFs (see Section 8.6.1), EAFs also need to be taken into account, even though technically they do not qualify as support frameworks. Please note that in this case, we will talk about the consistent subclass, not normal form, since we are not convinced that a suitable translation exists. Moreover, for now we will not define the consistent form for BAFs, as they will not be translated into ADFs (see Section 9.4).

Unlike the minimal form, the consistency form introduces auxiliary arguments and thus is not a subframework. The point of the translation is to ensure that no argument is directly attacked or supported by the same argument.This is done by changing the undesirable direct support into an indirect one through an additional bypass argument. In what follows we will consider the pure bypass consistency form and the self–attacker consistency form. In the first approach, the bypass is just an intermediate argument in the support relation and as such can appear in an extension. The translation is not very intrusive and has the chance to be applicable to non–standard semantics as well. However, it can be faithful at best. The other approach turns the bypass argument into a self–attacker, and although now we can have exact translations, the method cannot be used with some of the semantics.

We close this section with the definitions of consistent AFNs and EASs; EAFs will be described in Section 4.4.3. From now on when we will talk about consistency, we will refer to the strong version, unless stated otherwise.

Definition 4.46. Let $FN = (A, R, N)$ be a (set–form) AFN, a an argument in A and $X \subseteq A$ the set of attackers of a. We say that a is **consistent** iff $\nexists X' \subseteq X$, $X'Na$. a is **strongly consistent** iff $\forall B \subseteq A$ s.t. BNa, $X \cap B = \emptyset$. FN is (strongly) consistent iff all of its arguments are (strongly) consistent.

Definition 4.47. Let $ES = (A, R, E)$ be an EAS, a an argument in A, X the collection of all and only arguments s.t. $\exists X' \subseteq X$, $X'Ra$ and B the collection of all and only arguments s.t. $\exists B' \subseteq B$, $B'Ea$. a is **strongly consistent** iff $X \cap B = \emptyset$. ES is (strongly) consistent iff all of its arguments are (strongly) consistent.

4.4.1 Bypass Consistency Form

In Section 2.3.9 we have already discussed why the consistency form is needed and gave a possible method for obtaining it. We recall two of the analyzed examples:

Example 34. Let us analyze an AFN $FN_2 = (\{a, b, c, d\}, \{(a, d), (b, a), (d, c)\}, \{(\{b, c\}, a)\})$ depicted in Figure 27b. We have that a is again attacked by b, and either b or c needs to be present in order to accept a. If we were to defend from a, we would need to either accept its attacker or cut off its support. In the AFN setting, we would need to either accept b in the set, or attack both b and c, since the two coherent sets that we need to attack are $\{a, b\}$ and $\{a, c\}$. In ADF, only the set $\{c\}$ (or, equivalently, interpretation $\{b : \mathbf{f}, c : \mathbf{t}\}$) can satisfy the acceptance condition of a, thus assuming b or discarding c is sufficient. This can be seen as considering b primarily as an attacker. As a result, $\{d\}$ is ADF, but not AFN admissible.

Example 37. Now let us consider $FN_2 = (\{a, b, c, d\}, \{(a, d), (b, a), (d, c)\}, \{(\{b, c\}, a)\})$ from Example 34. We again introduce a bypass for b and obtain a framework $D_2 = (\{a, b, c, d, b^b\}, \{(a, d), (b, a), (d, c)\}, \{(\{b\}, b^b), (\{b^b, c\}, a)\})$ depicted in Figure 29b. It still holds that $\{d\}$ is not AFN admissible. The corresponding ADF is now $(\{a, b, c, d, b^b\}, \{C_a = \neg b \wedge (b^b \vee c), C_b = \top, C_c = \neg d, C_d = \neg a, C_{b^b} = b\})$ and this time $\{d\}$ is no longer admissible; its range interpretation falsifies c only and cannot prevent acceptance of b and b^b, and thus not of a.

In this section we will formalize the proposed method and show that it is indeed correct.

4.4.1.1 AFN Bypass Consistency Form

Let us now introduce the bypass consistency form of AFNs. The focus is on replacing not the arguments that are not consistent, but on those that caused them to be so. To an argument that is supporting and attacking another argument at the same time we will refer to as the inconsistency origin:

Definition 4.48. Let $FN = (A, R, N)$ be an AFN and $a \in A$ an argument. The **inconsistency origin** of a is defined as $O^a = \{b \in A \mid \exists B \subseteq A \text{ s.t. } b \in B, BNa \text{ and } bRa\}$.

We can observe that if a is strongly consistent, then $O^a = \emptyset$. By the abuse of notation, we will write O^E to denote the collection of all inconsistency origins of the arguments in $E \subseteq A$.

For those arguments that caused the inconsistency, we now need introduce bypass arguments. The bypasses are supported by their origins and, take their place in supporting the target argument that was previously inconsistent. In order to make the translation more readable, we will introduce the notion of a replacement function. The occurrences of a given argument in the supporting sets of another argument are replaced with auxiliary elements:

Definition 4.49. Let $FN = (A, R, N)$ be an AFN and $a \in A$ an argument. Given a set of arguments $S \subseteq A$, the replacing arguments $P^b = \{e^b \mid e \in S\}$ and a support $(B, a) \in N$, the **replacement function** is defined as:

$$rep(a, S, P^b, (B, a)) = \begin{cases} (B, a) & \text{if } B \cap S = \emptyset \\ (B', a) & \text{if } B \cap S \neq \emptyset, \text{ where } B' = (B \setminus S) \cup \{e^b \mid e \in B \cap S\} \end{cases}$$

Translation 13. *Let $FN = (A, R, N)$ be an AFN and $E \subseteq A$ the set of arguments that are not strongly consistent. The strongly consistent AFN $FN^{sc} = (A', R, N')$ corresponding to FN is created as follows:*

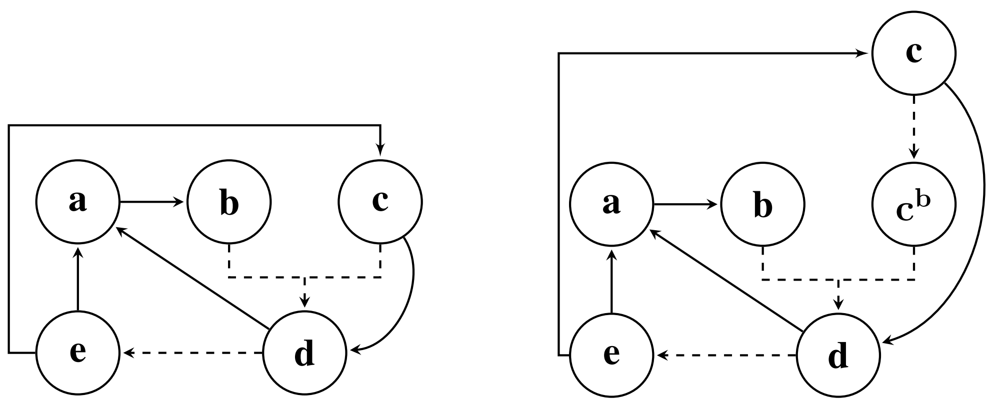
- *we introduce the bypass arguments to A: $A' = A \cup A^b$, where $A^b = \{a^b \mid a \in O^E\}$,*

- *for every pair $(B, a) \in N$, we replace the arguments in B causing inconsistency of a (if there is any) by their bypasses: we add $rep(a, O^a, A^b, (B, a))$ to N', and*

- *for every $a \in O^E$, add the support to its bypass, i.e. put $(\{a\}, a^b)$ in N'*

Example 65. Let us consider the AFN $FN_1 = (\{a, b, c, d, e\}, \{(a, b), (c, d), (d, a), (e, a), (e, c)\}, \{(\{b, c\}, d), (\{d\}, e)\})$ depicted in Figure 45a. Its admissible extensions are $\emptyset$, $\{c\}$, $\{a, c\}$ and $\{b, d, e\}$. The set $\{c\}$ attacks d, and since d is contained in every coherent set for e, $\{c\}$ defends itself. Moreover, it also defends a and we obtain the first three extensions. Extension $\{b, d, e\}$ attacks both a and c and defends its members. Additionally, the set is coherent – b requires no support, d is (acyclically) supported by b and e by d. Based on the presented analysis, the sets $\emptyset$, $\{a, c\}$ and $\{b, d, e\}$ are our complete extensions, with the first being grounded and the latter two preferred and stable.

We can observe that the argument d is consistent, but not strongly consistent and c is its inconsistency origin. Thus, we will introduce a bypass argument c^b for c. It will now be necessarily supported by c and take over the support from c to d. We thus obtain the strongly consistent normal form framework $FN^{sc} = (\{a, b, c, c^b, d, e\}, \{(a, b), (c, d), (d, a), (e, a), (e, c)\}, \{(\{c\}, c^b), (\{b, c^b\}, d), (\{d\}, e)\})$ from Figure 45b. Our admissible extensions are now $\emptyset$, $\{c\}$, $\{c, c^b\}$, $\{a, c\}$, $\{a, c, c^b\}$ and $\{b, d, e\}$, with $\emptyset$, $\{a, c, c^b\}$ and $\{b, d, e\}$ being the complete ones. Again, the first is also grounded and the remaining two are preferred and stable. Therefore, we can observe that the original extensions of FN_1 can be retrieved by simply removing the bypass argument c^b from the extensions.



(a) FN_1 (b) Strongly consistent form of FN_1

Figure 45: Sample AFN and its strongly consistent form

Theorem 4.50. *Let $FN = (A, R, N)$ be an AFN and $FN^{sc} = (A', R, N')$ its corresponding strongly consistent framework obtained through Translation 13. Let $E \subseteq A$, $E' \subseteq A'$ be sets of arguments and E^b the (possibly empty) set of bypass arguments generated by E in A'. If E is a σ–extension of FN, where $\sigma \in$ {conflict–free, coherent, admissible,* *preferred, complete, grounded, stable*$\}$*, then* $E \cup E^b$ *is a* σ*–extension of* FN^{sc}*. If* E' *is a* σ*–extension of* FN^{sc}*, then* $E' \setminus E^b$ *is a* σ*–extension of* FN*.*

From this theorem we can also observe that e.g. validity and minimal forms are preserved by our translation.

Theorem 4.51. *Let* $FN = (A, R, N)$ *be an AFN and* $FN^{sc} = (A', R, N')$ *its bypass consistency form obtained through Translation 13.* FN *is weakly, relation and strongly valid iff* FN^{sc} *is.* FN *is in minimal form iff* FN^{sc} *is.*

We can now put our translation into the system and analyze its properties.
Redefinition of Translation 13: *Let* Fr^{AFN} *be the collection of all AFNs on the domain* $\mathcal{U}$ *and* $SCons^{AFN}$ *the collection of all strongly consistent AFNs on the domain* $\mathcal{U} \cup \mathcal{U}^b$*. For a framework* $FN = (A, R, N)$*, let* $E \subseteq A$ *be the set of not strongly consistent arguments. The bypass consistency form translation bc-*$Tr^{AFN} : Fr^{AFN} \rightarrow SCons^{AFN}$ *is defined as bc-*$Tr^{AFN}((A, R, N)) = (A', R, N')$ *for* $(A, R, N) \in Fr^{AFN}$*, where:*

- $A' = A \cup A^b$*, where* $A^b = \{a^b \mid a \in O^E\}$*, and*
- $N' = \{(\{a\}, a^b) \mid a \in O^E\} \cup \{rep(a, O^a, A^b(B, a)) \mid (B, a) \in N\}$.

Redefinition of Theorem 4.50: *Let* $\sigma \in \{$*conflict–free, coherent, admissible, complete, preferred, grounded, stable*$\}$ *be a semantics and* SC_σ^{Tr} *the removal casting functions for* σ *defined as* $SC_\sigma^X(E) = E \cap A$*, where* $X \in Fr^{AFN}$ *is an AFN with the set of arguments* A *and* $E \in \sigma($*bc-*$Tr^{AFN}(X))$*. The translation bc-*Tr^{AFN} *is strong under* (σ, SC_σ^{Tr})*. It is semantics bijective under the complete, preferred, grounded and stable semantics and the defined removal casting functions.*
Analysis of Translation 13: Under the conflict–free, coherent, admissible, preferred, complete, grounded and stable semantics and their removal casting functions, the translation *bc-*Tr^{AFN} is:

- full, target–subclass, injective
- weakly argument domain altering, argument introducing, induced support relation introducing, support relation removing, attack relation preserving
- generic and weakly semantics domain altering
- structural

Translation *bc-*Tr^{AFN} is not modular. Under the complete, preferred, grounded, stable and the removal casting functions, it is faithful.

Explanation. Any AFN can be taken as an input, even if it is in the consistency form already. The produced AFNs will always be strongly consistent and it is easy to see this does not cover all the possible AFNs. Thus, our approach is a target–subclass one. Our method is also easily seen to be injective; it suffices to remove the bypass argument and reconnect the support to uniquely retrieve the source framework.

We can observe that both the semantics and argument domain are weakly altered – we need access to bypass arguments that would not appear in any source framework. The translation also introduces additional arguments – the bypasses – but does not delete any existing ones. Technically speaking, we are removing and adding some support relations – after all, N and N' are not comparable. However, we are rather dealing with a replacement; a given argument is still supporting its target, just indirectly. Moreover, the change is only in the "vicinity" of the bypass arguments. Since R does not undergo any changes, our translation is clearly attack preserving. The fact that $bc\text{-}Tr^{AFN}$ is generic can be easily seen from the amount of semantics it handles. Finally, as the modifications done to the framework do not depend in any way on the semantics, $bc\text{-}Tr^{AFN}$ is structural.

Let us now show that our approach is, unfortunately, not modular. Consider a simple framework $FN_1 = (\{a,b\}, \{(a,b)\}, \{(\{a\}, b)\})$, in which a both supports and attacks b. Its consistency normal form is $FN_1'(\{a, a^b, b\}, \{(a,b)\}, \{(\{a^b\}, b), (\{a\}, a^b)\})$. Let us look at two subframeworks $FN_2 = (\{a,b\}, \emptyset, \{(\{a\}, b)\})$ and $FN_3 = (\{a,b\}, \{(a,b)\}, \emptyset\})$ of FN_2. Both of them are strongly consistent and will not be affected by the translation. However, their union is again the original framework FN_1, not its strongly consistent form FN_1', and as such is not even in $SCons^{AFN}$. ■

4.4.1.2 EAS Bypass Consistency Form

The consistency form and the semantics correspondence proofs for EASs follow the same line of reasoning as for AFNs. We introduce a bypass argument whenever there is an argument playing both attacking and supporting role and let the bypass take over the problematic support:

Definition 4.52. Let $ES = (A, R, E)$ be an EAS and $a \in A$ an argument. The **inconsistency origin** of a is defined as $O^a = \{b \in A \mid \exists B, B' \subseteq A s.t. b \in B \cap B', BEa$ and $B'Ra\}$.

We can observe that if a is strongly consistent, then $O^a = \emptyset$. By the abuse of notation, we will write O^E to denote the collection of all inconsistency origins of the arguments in $E \subseteq A$.

For those arguments that caused the inconsistency, we now need introduce bypass arguments. The bypasses are supported by their origins and, take their place in supporting the target argument that was previously inconsistent. In order to make the translation more readable, we will introduce the notion of a replacement function. The occurrences of a given argument in the supporting sets of another argument are replaced with auxiliary elements:

Definition 4.53. Let $ES = (A, R, E)$ be an EAS and $a \in A$ an argument. Given a set of arguments $S \subseteq A$, the replacing arguments $P^b = \{e^b \mid e \in S\}$ and a support $(B, a) \in E$, the **replacement function** is defined as:

$$rep(a, S, P^b, (B, a)) = \begin{cases} (B, a) & \text{if } B \cap S = \emptyset \\ (B', a) & \text{if } B \cap S \neq \emptyset, \text{ where } B' = (B \setminus S) \cup \{e^b \mid e \in B \cap S\} \end{cases}$$

Translation 14. *Let $ES = (A, R, E)$ be an EAS and $S \subseteq A$ the set of arguments that are not strongly consistent. By the inconsistency origin of an argument $e \in S$ we will understand the set $O^e = \{a \in A \mid \exists B, B' \subseteq A s.t. a \in B, B', BEe$ and $B'Re\}$. The union of the inconsistency origins of all arguments in S will be denoted O^S. The strongly consistent EAS $ES^{sc} = (A', R, E')$ corresponding to ES is created as follows:*

- *we introduce the bypass arguments to A: $A' = A \cup A^b$, where $A^b = \{a^b \mid a \in O^S\}$,*
- *for every pair $(B, a) \in E$, replace the arguments causing inconsistency by their bypasses, i.e. we add $rep(a, O^a, A^b, (B, a))$ to E', and*
- *for every $a \in O^S$, add the support to its bypass, i.e. put $(\{a\}, a^b)$ in E'.*

Theorem 4.54. *Let $ES = (A, R, E)$ be an EAS and $ES^{sc} = (A', R, E')$ its corresponding strongly consistent framework obtained through Translation 14. Let $S \subseteq A$, $S' \subseteq A'$ be sets of arguments and S^b the (possibly empty) set of bypass arguments generated by S in A'. If S is a σ–extension of ES, where $\sigma \in \{$conflict–free, self–supporting, admissible, preferred, complete, grounded, stable$\}$, then $S \cup S^b$ is a σ–extension of ES^{sc}. If S' is a σ–extension of ES^{sc} then $S' \setminus S^b$ is a σ–extension of ES.*

Similarly as in the AFN case, the translation preserves the validity and minimal normal forms.

Theorem 4.55. *Let $ES = (A, R, E)$ be an EAS and $ES^{sc} = (A', R, E')$ its bypass consistency form obtained through Translation 14. ES is weakly, relation and strongly valid iff ES^{sc} is. ES is in minimal form iff ES^{sc} is.*

We can now put our translation into the system and analyze its properties.

Redefinition of Translation 14: *Let Fr^{EAS} be the collection of all EASs on the domain $\mathcal{U}$ and $SCons^{EAS}$ the collection of all strongly consistent EASs on the domain $\mathcal{U} \cup \mathcal{U}^b$. For a framework $ES = (A, R, E)$, let $S \subseteq A$ be the set of not strongly consistent arguments. The bypass consistency form translation bc-$Tr^{EAS} : Fr^{EAS} \to SCons^{EAS}$ is defined as bc-$Tr^{EAS}((A, R, E)) = (A', R, E')$ for $(A, R, E) \in Fr^{EAS}$, where:*

- *$A' = A \cup A^b$, where $A^b = \{a^b \mid a \in \bigcup_{e \in S} O^e\}$, and*
- *$E' = \{(\{a\}, a^b) \mid a \in O^S\} \cup \{rep(a, O^a, A^b(B, a)) \mid (B, a) \in E\}$.*

Redefinition of Theorem 4.54: *Let $\sigma \in \{$conflict–free, self–supporting, admissible, complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_σ the removal casting functions for σ defined as $SC^X_\sigma(S) = S \cap A$, where $X \in Fr^{EAS}$ is an EAS with the set of arguments A and $S \in \sigma($bc-$Tr^{EAS}(X))$. The translation bc-Tr^{EAS} is strong under (σ, SC^{Tr}_σ). It is* *semantics bijective under the complete, preferred, grounded and stable semantics and the removal casting functions.*

Analysis of Translation 13: Under the conflict–free, self–supporting, admissible, preferred, complete, grounded and stable semantics and their removal casting functions, the translation $bc\text{-}Tr^{EAS}$ is:

- full, target–subclass, injective
- weakly argument domain altering, argument introducing, induced support relation introducing, support relation removing, attack relation preserving
- generic and weakly semantics domain altering
- structural

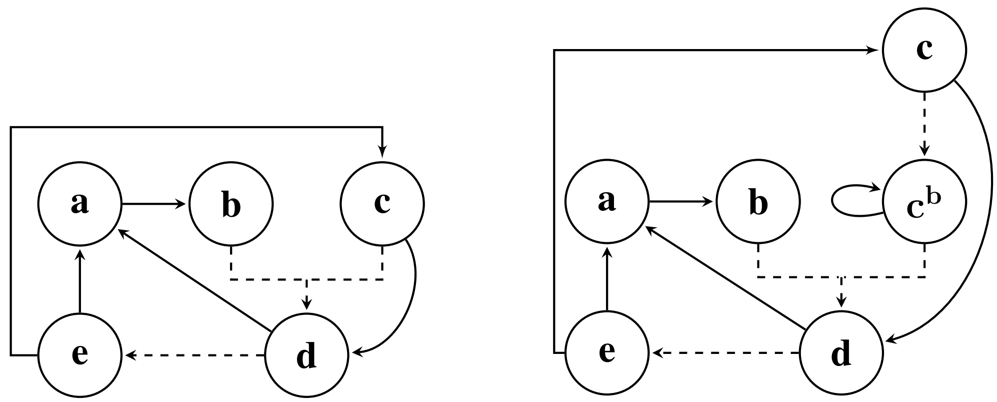
Translation $bc\text{-}Tr^{EAS}$ is not modular. Under the complete, preferred, grounded and stable semantics and the removal casting functions, it is faithful.

The same explanations as given in the AFN case hold for the EAS translation. Thus, we will omit them.

4.4.2 The Self–Attacker Consistency Form

The bypass consistency form has the advantage of not affecting the attack relation and only changing the length of existing supports. Consequently, this normal form has also the chance of satisfying more demanding types of semantics, such as prudent or careful [32, 33], if and when they are moved to AFNs and EASs. Or, in other words, if we looked at these frameworks from the perspective of BAFs, then e.g. the safety constraints would not be greatly affected and the created auxiliary attacks for bypass arguments would be the same as for their origins. The price for this is that we need auxiliary arguments which show up in the extensions. Thus, with this approach we can have faithful translations at best. Consequently, we propose another way to obtain the consistent normal form, in which the bypass arguments are also self–attacking. Since they will not appear in the extensions, we can obtain an exact normal form translation, with the side effect that it might be less useful for other types of semantics in the future. Unfortunately, this approach is not applicable for stable semantics:

Example 66. Let us come back to the AFN $FN_1 = (\{a, b, c, d, e\}, \{(a, b), (c, d), (d, a), (e, a), (e, c)\}, \{(\{b, c\}, d), (\{d\}, e)\})$ from Example 65, again depicted in Figure 46a. Its admissible extensions were $\emptyset$, $\{c\}$, $\{a, c\}$ and $\{b, d, e\}$. By introducing the c^b bypass and changing it into a self–attacker we obtain the strongly consistent normal form framework $FN^{sc} = (\{a, b, c, c^b, d, e\}, \{(a, b), (c, d), (d, a), (e, a), (e, c), (c^b, c^b)\}, \{(\{c\}, c^b), (\{b, c^b\}, d), (\{d\}, e)\})$ from Figure 46b. Our admissible extensions are now $\emptyset$, $\{c\}$, $\{a, c\}$ and $\{b, d, e\}$, with $\emptyset$, $\{a, c\}$ and $\{b, d, e\}$ being the complete ones. Again, the first is also grounded and the remaining two are preferred. However, we can observe that while $\{b, d, e\}$ is a stable extension, $\{a, c\}$ not anymore. It fails to attack c^b in any way and thus does not meet the stability requirements. Fortunately, aside from stable semantics, we can see that the original extensions of FN_1 are exactly retrieved from FN_2.



(a) FN_1

(b) Self–attacker strongly consistent form of FN_1

Figure 46: Sample AFN and its self–attacker strongly consistent form

We will now introduce a new translation type for obtaining a consistent framework from the original one.

4.4.2.1 AFN Self–Attacker Consistency Form

The self–attacker consistency form transformation is just a slight modification of Translation 13 and we will still use the inconsistency origin and replacement functions from Definitions 4.48 and 4.49. The only difference is in the modification of the attack relation:

Translation 15. *Let $FN = (A, R, N)$ be an AFN and $E \subseteq A$ the set of arguments that are not strongly consistent. The strongly consistent AFN $FN^{sc} = (A', R', N')$ corresponding to FN is created as follows:*

- *we introduce the bypass arguments to A: $A' = A \cup A^b$, where $A^b = \{a^b \mid a \in O^E\}$,*
- *the attack relation extends the existing one by adding bypass self–attacks: $R' = R \cup \{(a^b, a^b) \mid a^b \in A^b\}$,*
- *for every pair $(B, a) \in N$, we replace the arguments in B causing inconsistency of a (if there is any) by their bypasses: we add $rep(a, O^a, A^b, (B, a))$ to N', and*
- *for every $a \in O^E$, add the support to its bypass, i.e. put $(\{a\}, a^b)$ in N'.*

Theorem 4.56. *Let $FN = (A, R, N)$ be an AFN and $FN^{sc} = (A', R', N')$ its corresponding strongly consistent framework obtained through Translation 15. Let E^b the (possibly* *empty) set of bypass arguments generated by a set $E \subseteq A$ in A'. If a set of arguments E is coherent in FN, then $E \cup E^b$ is pd–acyclic in FN^{sc}. If $E' \subseteq A'$ is pd–acyclic in FN^{sc}, then $E' \cap A$ is coherent in E. $E \subseteq A$ is a σ–extension of FN, where $\sigma \in \{$conflict–free, strongly coherent, admissible, preferred, complete, grounded$\}$, iff it is a σ–extension of FN^{sc}. Every stable extension E of FN^{sc} is stable in FN but not vice versa.*

In order to address the issue of stability, one would have to provide means for attacking the bypass arguments. If we add an attack from the origin to a bypass, we breach the consistency restrictions and the whole approach is useless as a normal form. If we replace the origin–bypass support by attack, we make it impossible for the arguments that attack all coherent sets of the origin to attack all coherent sets of the bypass and breach the stability requirements again, though from a different side. Propagating the attacks on the origin to the bypass would require the use of group attack, which is not present in AFNs (see attack propagation translations, e.g. Translation 63). Moreover, it would be quite an expensive approach just to handle a rather minor local problem of inconsistency. Another way of dealing with this problem would be to copy the support and attack relation the origin receives to its bypass, thus avoiding the need for propagation. We would obtain basically a duplicate argument attacked by the original one and although the inconsistency originating and the initial argument would be resolved, we might be introducing a new one. The original argument might be inconsistent itself and appear as a member of its own support set, thus clashing on the bypass level with the attack it carries out. Although by using some validity forms that will be presented in the next section we can prevent inconsistent bypasses, such an approach seems rather excessive compared to the current ones.

The validity and minimal forms are preserved by our translation in the same way they were by Translation 13.

Theorem 4.57. *Let $FN = (A, R, N)$ be an AFN and $FN^{sc} = (A', R, N')$ its self–attacker consistency form obtained through Translation 15. FN is weakly, relation and strongly valid iff FN^{sc} is. FN is in minimal form iff FN^{sc} is.*

Redefinition of Translation 15: *Let Fr^{AFN} be the collection of all AFNs on the domain $\mathcal{U}$ and $SCons^{AFN}$ the collection of all strongly consistent AFNs on the domain $\mathcal{U} \cup \mathcal{U}^b$. For a framework $FN = (A, R, N)$, let $E \subseteq A$ be the set of not strongly consistent arguments. The self–attacker consistency form translation sa-Tr^{AFN} : $Fr^{AFN} \rightarrow SCons^{AFN}$ is defined as sa-$Tr^{AFN}((A, R, N)) = (A', R', N')$ for $(A, R, N) \in Fr^{AFN}$, where:*

- $A' = A \cup A^b$, *where* $A^b = \{a^b \mid a \in \bigcup_{e \in E} O^e\}$,
- $R' = R \cup \{(a^b, a^b) \mid a \in A^b\}$, *and*
- $N' = \{(\{a\}, a^b) \mid a \in O^E\} \cup \{rep(a, O^a, A^b(B, a)) \mid (B, a) \in N\}$.

Redefinition of Theorem 4.56: *Let $\sigma \in \{$conflict–free, coherent, strongly coherent, admissible, complete, preferred, grounded$\}$ be a semantics and SC^{Tr}_{σ} their identity casting* *functions. The translation $sa\text{-}Tr^{AFN}$ is strong under (σ, SC_σ^{Tr}). With the exception of coherent semantics, it is also semantics bijective. It is $\supseteq$–weak under the stable semantics and identity casting functions.*

Analysis of Translation 15: Under the conflict–free, coherent, strongly coherent, admissible, preferred, complete, grounded and stable semantics and their identity casting functions, the translation $sa\text{-}Tr^{AFN}$ is:

- full, target–subclass, injective
- weakly argument domain altering, argument introducing, induced support relation introducing, support relation removing, induced attack relation introducing
- semantics domain preserving
- structural

Translation $sa\text{-}Tr^{AFN}$ is not modular. Under the conflict–free, strongly coherent, admissible, preferred, complete and grounded semantics and the identity casting functions, it is generic and exact.

Explanation. Most of the properties can be explained the same way as it was done during the analysis of Translation 13. Since $R \subseteq R'$, it should be clear that the translation introduces new attacks. The same example can be used to show that this approach is also not modular. The fact that the translation is exact for the standard semantics (with the exception of stable) follows easily from the redefinition of Theorem 4.56. Moreover, this time the semantics domain is preserved – the auxiliary arguments do not show up in the extensions. ■

4.4.2.2 EAS Self–Attacker Consistency Form

Similarly as in the AFN case, the EAS self–attacker consistency form is only a minor modification of the bypass one. We only add additional conflicts for the bypass arguments:

Translation 16. *Let $ES = (A, R, E)$ be an EAS and $S \subseteq A$ the set of arguments that are not strongly consistent. The strongly consistent EAS $ES^{sc} = (A', R, E')$ corresponding to ES is created as follows:*

- *we introduce the bypass arguments to A: $A' = A \cup A^b$, where $A^b = \{a^b \mid a \in O^S\}$,*
- *the attack relation extends the existing one by adding bypass self attacks: $R' = R \cup \{(\{a^b\}, a^b) \mid a^b \in A^b\}$,*
- *for every pair $(B, a) \in E$, replace the arguments causing inconsistency by their bypasses, i.e. we add $rep(a, O^a, A^b, (B, a))$ to E', and*
- *for every $a \in O^S$, add the support to its bypass, i.e. put $(\{a\}, a^b)$ in E'.*

Theorem 4.58. *Let $ES = (A, R, E)$ be an EAS and $ES^{sc} = (A', R, E')$ its corresponding strongly consistent framework obtained through Translation 16. Let $S \subseteq A$, $S' \subseteq A'$ be sets of arguments and S^b the (possibly empty) set of bypass arguments generated by S in A'. S is a σ–extension of ES, where $\sigma \in \{$conflict–free, self–supporting conflict–free, admissible, preferred, complete, grounded, stable$\}$, iff it is a σ–extension of ES.*

Due to the fact that the actual support relation in both self–attacker and bypass forms is the same, the same normal forms are preserved.

Theorem 4.59. *Let $ES = (A, R, E)$ be an EAS and $ES^{sc} = (A', R, E')$ its self–attacker consistency form obtained through Translation 16. ES is weakly, relation and strongly valid iff ES^{sc} is. ES is in minimal form iff ES^{sc} is.*

We can now put our translation into the system and analyze its properties.

Redefinition of Translation 16: *Let Fr^{EAS} be the collection of all EASs on domain $\mathcal{U}$ and $SCons^{EAS}$ the collection of all strongly consistent EASs on domain $\mathcal{U} \cup \mathcal{U}^b$. For a framework $ES = (A, R, E)$, let $S \subseteq A$ be the set of not strongly consistent arguments. The self–attacker consistency form translation $sa\text{-}Tr^{EAS} : Fr^{EAS} \rightarrow SCons^{EAS}$ is defined as $sa\text{-}Tr^{EAS}((A, R, E)) = (A', R', E')$ for $(A, R, E) \in Fr^{EAS}$, where:*

- *$A' = A \cup A^b$, where $A^b = \{a^b \mid a \in \bigcup_{e \in S} O^e\}$,*
- *$R' = R \cup \{(\{a^b\}, a^b) \mid a^b \in A^b\}$, and*
- *$E' = \{(\{a\}, a^b) \mid a \in O^S\} \cup \{rep(a, O^a, A^b(B, a)) \mid (B, a) \in E\}$.*

Redefinition of Theorem 4.58: *Let $\sigma \in \{$conflict–free, self–supporting conflict–free, admissible, complete, preferred, grounded$\}$ be a semantics and SC^{Tr}_{σ} the identity casting functions for σ. The translation $sa\text{-}Tr^{EAS}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$. The translation is $\supseteq$–weak under stable semantics and identity casting functions.*

Analysis of Translation 15: Under the conflict–free, self–supporting conflict–free, admissible, preferred, complete, grounded and stable semantics and their identity casting functions, the translation $sa\text{-}Tr^{EAS}$ is:

- full, target–subclass, injective
- weakly argument domain altering, argument introducing, induced support relation introducing, support relation removing, induced attack relation introducing
- semantics domain preserving
- structural

Translation $sa\text{-}Tr^{EAS}$ is not modular. With the exception of the stable semantics, the translation is generic and exact under the listed semantics and identity casting functions.

The same explanations as given in the AFN self–attacker consistency translation hold for the EAS version. Thus, we will omit them.

4.4.3 Consistent EAF and EAFC Subclasses

Finally, we would also like to introduce a subclass of EAFs (EAFCs) referred to as (strongly) consistent. The reason for introducing it is basically the same as in AFNs and EASs; since defense attack can be translated into a particular form of support in ADFs (see Section 8.6.1), the issue of being attacked and supported by the same argument arises also in this framework. Already in Section 2.3.9 we could have observed that ignoring inconsistencies produces framework with undesirable extensions. Therefore, in the consistent EAF(C)s we intend to prevent situations where an argument is at the same time an attacker and defense attacker of the same argument.

However, we are not aware of any method that would transform a given EAF or EAFC into a consistent one. The sketch we have proposed in Section 2.3.9 created an ADF that used support not related to defense attacks. Consequently, we will consider consistent EAFs (EAFCs) a subclass, not a normal form for now. Please note that due to the fact that EAFs allow only binary defense attacks, the consistent and strongly consistent subclass will be one and the same thing.

Definition 4.60. Let $EF = (A, R, D)$ be an EAF. EF is **(strongly) consistent** iff there is no $x, y, z \in A$ s.t. $(x, y) \in R$ and $(x, (z, y)) \in D$.

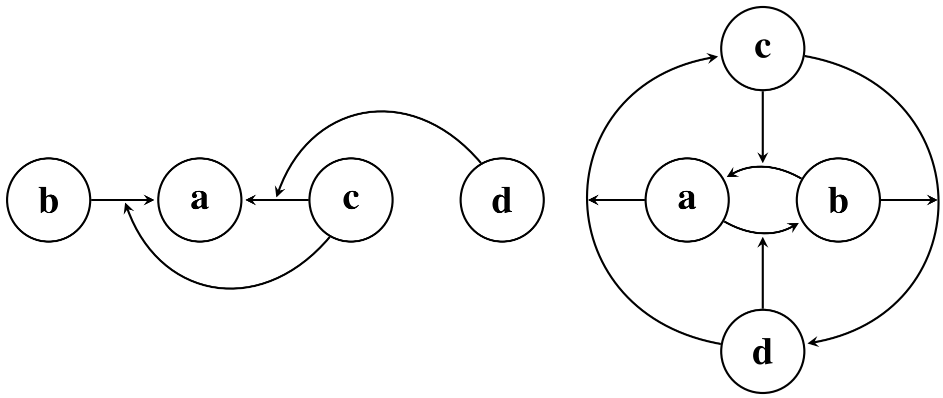
Definition 4.61. Let $EFC = (A, R, D)$ be an EAFC. EFC is **consistent** iff there is no $x, y, z \in A$ s.t. $(x, y) \in R$ and $(\{x\}, (z, y)) \in D$. It is **strongly consistent** iff there is no $x, y, z \in A$ and $X \subseteq A$ s.t. $(x, y) \in R$, $x \in X$ and $(X, (z, y)) \in D$.

Example 67. Let us consider the EAF $(\{a, b, c, d\}, \{(c, a), (b, a)\}, \{(d, (c, a)), (c, (b, a))\})$ depicted in Figure 47a. We can observe that at the same time, c attacks a and defense attacks the conflict (b, a). Thus, this is not a consistent framework.

We can also recall the framework $(\{a, b, c, d\}, \{(a, b), (b, a), (c, d), (d, c)\}, \{(a, (d, c)), (b, (c, d)), (c, (b, a)), (d, (a, b))\})$ from Example 5, for convenience again depicted in Figure 47b. Although the structure is quite cyclic, it is still consistent; at no occasion is a given argument a (direct) attacker and defense attacker of another argument. Similar holds for EAFs from Examples 6 and 8.

4.5 Additional Framework Subclasses

We close this section with a list of various subclasses of the frameworks we are working with that will become useful in the next parts of this work. Please note that the majority of the introduced definitions will be quite straightforward. Many of our translations will be target–subclass, and in many cases the produced frameworks will have some special structural properties. Consequently, we have decided to gather the possible subclasses into a single section. Additionally, we will show what types of normal forms they satisfy.



(a) Not strongly consistent EAF (b) Strongly consistent EAF

Figure 47: Examples of strongly and not strongly consistent EAFs

4.5.1 SETAF Subclasses

Concerning SETAF, there is one subclass worth mentioning explicitly, which is the Dung–style SETAF. By this we understand that all sets of attacking arguments contain only a single argument:

Definition 4.62. Let $SF = (A, R)$ be a SETAF. SF is **binary** iff for every $E \subseteq A$ s.t. $\exists a, (E, a) \in R$ it holds that $|E| = 1$. With $Bin^{SETAF} \subseteq Fr^{SETAF}$ we denote the binary subclass of SETAFs.

Clearly, any binary SETAF is in minimal form. Moreover, we can easily create SETAFs that are in minimal form but are not binary, and thus the relation between the two is strict:

Lemma 4.63. *Let $SF = (A, R)$ be a SETAF. It holds that $Bin^{SETAF} \subsetneq Min^{SETAF}$.*

4.5.2 AFRA Subclasses

For AFRAs, we would like to introduce the notion of the depth of a conflict and use it to limit the level of recursion. It is important to note that not in every AFRA we can determine how deep a given conflict is. For example, the AFRA depicted in Figure 48 is basically one conflict cycle.

Definition 4.64. Let $FR = (A, R)$ be an AFRA. The **zero–depth conflicts** of FR are defined as $R_0 = R \cap (A \times A)$, i.e. they are directed at arguments only. The **i–depth conflicts** of FR, where $i > 0$, are defined as $R_i = R \cap (A \times R_{i-1})$. FR is of recursion depth i iff $R = \bigcup_{j=0}^{i} R_i$.

With Rec_i^{AFRA} we will denote the subclass of AFRAs with recursion depth i. If $i = 0$, then we call FR non–recursive.

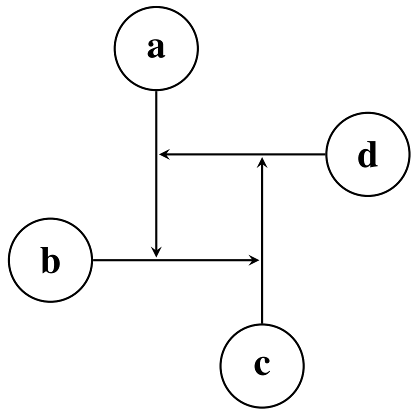


Figure 48: Cyclic AFRA

4.5.3 EAF and EAFC Subclasses

Apart from the previously considered strongly consistent and bounded hierarchical subclasses, three more types of EAF(C)s will be useful to use. The first encompasses the frameworks in which no defense attacks occur. Another, excludes symmetric attacks from the framework, which are responsible for the semantical differences between EAFs and EAFCs (see Section 2.1.4). Finally, we simply distinguish the binary EAFCs, where all defense attack are carried out by sets of size 1.

Definition 4.65. Let $EF = (A, R, D)$ be an EAF(C). EF is **without defense attacks** iff $D = \emptyset$. It is **without symmetric attacks** iff there are no $a, b \in A$ s.t. $(a, b), (b, a) \in R$.

Definition 4.66. Let $EFC = (A, R, D)$ be an EAFC. EFC is **binary** iff for every $(a, (b, c)) \in D$, $|a| = 1$. With $Bin^{EAFC} \subsetneq Fr^{EAFC}$ we denote the subclass of binary EAFCs.

With $NDef^{EAF(C)} \subsetneq Fr^{EAF(C)}$ and $NSym^{EAF(C)} \subsetneq Fr^{EAF(C)}$ we will denote the subclasses of EAF(C)s without defense attacks and symmetric attacks respectively. With $Bin^{EAFC} \subsetneq Fr^{EAFC}$ we mark the binary EAFCs.

We can observe that an EAF(C) without defense attacks is basically an AF. Consequently, it is trivially bounded hierarchical and strongly consistent in both of the frameworks. Lack of defense attacks in EAFCs also qualifies it as binary. The only less trivial, though extremely useful property, is that bounded hierarchical frameworks are also strongly consistent; this comes from the restriction that an attacker of an argument has to be in the same layer as the argument it attacks, but the defense attacker has to be in the next group.

Lemma 4.67. *The following holds between the subclasses and normal forms of EAF(C)s:*

- $NDef^{EAF(C)} \subsetneq BH^{EAF(C)}$
- $NDef^{EAF(C)} \subsetneq SCons^{EAF(C)}$

- $BH^{EAF(C)} \subsetneq SCons^{EAF(C)}$
- $NDef^{EAFC} \subsetneq Bin^{EAFC} \subsetneq Min^{EAFC}$

4.5.4 BAF Subclasses

A number of subclasses for BAFs can be distinguished. For future research, it would be interesting to distinguish classes of BAFs on which the semantics classification collapses. However, in our case we will focus only on three aspects. First, we consider BAFs without support and those that are support acyclic. Additionally, we will introduce the notion of support depth.

Definition 4.68. Let $BF = (A, R, S)$ be a BAF. BF is **without support** iff $S = \emptyset$. It is **support acyclic** iff the graph (A, S) is directed acyclic. BF is of **support depth** n if all (simple) paths in the directed graph (A, S) are of length at most n,

With $NSup^{BAF} \subsetneq Fr^{BAF}$, $SAcy^{BAF} \subsetneq Fr^{BAF}$ and $Sup_i^{BAF} \subsetneq Fr^{BAF}$ we denote the subclasses of BAFs without support, with acyclic support and with support depth i respectively.

It is easy to see that $NSup^{BAF} \subsetneq SAcy^{BAF}$. Since we did not really consider any further normal forms for BAFs, we close this analysis only with a small remark on some properties of BAFs without support:

Lemma 4.69. *Let $BF = (A, R, S)$ be a BAF with $S = \emptyset$. Then $R^{ind} = \emptyset$. Every set of arguments $E \subseteq A$ is closed and inverse closed under S and if E is +conflict–free, it is also safe.*

4.5.5 AFN Subclasses

Just like in BAFs, we will distinguish subclasses without support and with a given support depth. Due to the fact that AFNs permit support from groups of arguments, we can also consider framework in which the sets carrying out support are only of size 1 or where at most one supporting set exists per argument.

Definition 4.70. Let $FN = (A, R, N)$ be an AFN and $SG^{FN} = (A, N')$, where $N' = \{(a, b) \mid \exists E \subseteq A, a \in E \text{ s.t. } ENb\}$, the support graph induced by FN. Then FN is:

- **without support** iff $N = \emptyset$.
- **support binary** iff for every $S \subseteq A$ s.t. $\exists a, (S, a) \in N$ it holds that $|S| = 1$.
- **support singular** iff for every argument $a \in A$, there exists at most one set $S \subseteq A$ s.t. $(S, a) \in N$.
- of **support depth** n if all (simple) paths in SG^{FN} are of length at most n.

- **elementary** with depth n if it is support binary, strongly valid and of support depth n.

- **well–structured** if it is minimal, strongly consistent and strongly valid.

With $NSup^{AFN} \subsetneq Fr^{AFN}$, $SBin^{AFN} \subsetneq Fr^{AFN}$ and $SSig^{AFN} \subsetneq Fr^{AFN}$ we denote the subclasses of AFNs without support, support binary and support singular respectively. We will use $Sup_n^{AFN} \subset Fr^{AFN}$ to refer to the subclass of AFNs with support depth n. Finally, $SEle_n^{AFN} = SBin^{AFN} \cap SV^{AFN} \cap Sup_n^{AFN}$ stands for the elementary AFNs of depth n and $WSt^{AFN} = Min^{AFN} \cap SCons^{AFN} \cap SV^{AFN}$ denotes the well–structured frameworks.

We can now show some of the relations between our subclasses and normal forms which will become useful in further parts of this work.

Lemma 4.71. *The following holds between the subclasses and normal forms of AFNs:*

- $NSup^{AFN} \subsetneq SCons^{AFN}$
- $NSup^{AFN} \subsetneq SBin^{AFN}$
- $NSup^{AFN} \subsetneq SSig^{AFN}$
- $NSup^{AFN} = Sup_0^{AFN} \cap SV^{AFN}$
- $SBin^{AFN} \subsetneq Min^{AFN}$
- $SSig^{AFN} \subsetneq Min^{AFN}$
- $SV^{AFN} \subsetneq (WV^{AFN} \cap RV^{AFN})$

4.5.6 EAS Subclasses

In addition to the previously presented normal forms, we will also distinguish five simple EAS subclasses. The support and attack singular classes assumes that there exists at most one supporting (attacking) set for every argument. The support and attack binary frameworks allow their respective relations to be carried out by sets of size 1 only. The all–supported subclass assumes that every argument is supported this way or the other, independently of whether it is valid or not. This will become useful when translating from EASs to other frameworks with support (see e.g. Section 11.4). Finally, we distinguish the pure evidence supported frameworks, where every supporting set consists only of η. The intersection of these two classes produces a type of frameworks which is often associated with translations from attack–based structures to EASs.

Definition 4.72. Let $ES = (A, R, E)$ be an EAS. It is:

- **support singular** iff for every argument $a \in A$ there exists at most one set $S \subseteq A$ s.t. $(S, a) \in E$.

- **attack singular** iff for every argument $a \in A$ there exists at most one set $S \subseteq A$ s.t. $(S, a) \in R$.
- **support binary** iff for every $S \subseteq A$ s.t. $\exists a, (S, a) \in E$ it holds that $|S| = 1$.
- **attack binary** iff for every $S \subseteq A$ s.t. $\exists a, (S, a) \in R$ it holds that $|S| = 1$.
- **all–supported** iff for every $a \in A \setminus \{\eta\}$ there is a set $S \subseteq A$ s.t. $(S, a) \in E$.
- **pure evidence supported** iff every $S \subseteq A$ s.t. $\exists a, (S, a) \in E$ is of the form $S = \{\eta\}$.

With $ABin^{EAS} \subsetneq Fr^{EAS}$ and $ASig^{EAS} \subsetneq Fr^{EAS}$ we denote the attack binary and attack singular subclasses of EASs. Their support counterparts are $ABin^{EAS}$ and $SBin^{EAS}$. With $AllSup^{EAS} \subsetneq Fr^{EAS}$ we distinguish the all–supported EASs and use $EvSup^{EAS} \subsetneq Fr^{EAS}$ to denote the pure evidence supported ones.

The following properties can be shown for the listed subclasses and the introduced normal forms:

Lemma 4.73. *The following holds between the subclasses and normal forms of EAFs:*

- $(ABin^{EAS} \cup ASig^{EAS}) \cap (SBin^{EAS} \cup SSig^{EAS})) \subsetneq Min^{EAS}$.
- $EvSup^{EAS} \subsetneq (SBin^{EAS} \cap SSig^{EAS})$
- $EvSup^{EAS} \subsetneq SCons^{EAS}$.
- $(EvSup^{EAS} \cap AllSup^{EAS}) \subsetneq (SCons^{EAS} \cap SV^{EAS})$
- $SV^{EAS} \subsetneq (WV^{EAS} \cap RV^{EAS})$
- $WV^{EAS} \subsetneq AllSup^{EAS}$

4.5.7 ADF Subclasses

Finally, we come to the subclasses for ADFs. We will focus on distinguishing certain types of conditions that correspond to various types of attacks – the binary, group and defense conflicts. We will refer to them by the names of the frameworks that developed a given relation.

Definition 4.74. Let $D = (A, L, C)$ be an ADF. D is **Dung–style** (or AF–style) if for every argument $a \in A$, $C_a(\emptyset) = in$ and for every nonempty $E \subseteq par(a)$, $C_a(E) = out$. Equivalently, it is Dung–style if every condition is a conjunction of negated arguments or $\top$.

Definition 4.75. Let $D = (A, L, C)$ be an ADF. D is **SETAF–style** if for every argument $a \in A$, *i*) $C_a(\emptyset) = in$, *ii*) if $par(a) \neq \emptyset$, then there exists $E \subseteq par(a)$ s.t. $C_a(E) = out$, *iii*) and for every nonempty $E \subseteq par(a)$ s.t. $C_a(E) = out$, $C_a(E') = out$ for every $E \subseteq E' \subseteq par(a)$. Equivalently, it is SETAF–style if every condition is a conjunction of clauses containing only negated arguments or $\top$.

Although we do not provide a functional description for EAF(C)–style conditions, we choose to distinguish this subclass due to the relation between defense attacks and overpowering support (see Section 8.6.1 for further discussion).

Definition 4.76. Let $D = (A, L, C)$ be an ADF. D is **EAFC–style** if for every argument $a \in A$, the condition C_a is a conjunction of clauses containing exactly one negated argument and an arbitrary amount of positive ones or $C_a = \top$. D is **EAF–style** if it is EAFC–style and no negation of an argument can appear in two different clauses.

To classes of frameworks containing the conditions of a given style we will refer to as $ADF^{AF} \subsetneq Fr^{ADF}$, $ADF^{SETAF} \subsetneq Fr^{ADF}$, $ADF^{EAFC} \subsetneq Fr^{ADF}$ and $ADF^{EAF} \subsetneq ADF^{EAFC}$.

We will now give some of the properties of our subclasses and repeat some of the relations between the normal forms for easy access.

Lemma 4.77. *The following holds between the subclasses and normal forms of ADFs:*

- $ADF^{AF} \subsetneq (ADF^{SETAF} \cap RFree^{ADF})$
- $ADF^{SETAF} \subsetneq (BADF \cap AADF^{+})$
- $ADF^{SETAF} \subsetneq WV^{ADF}$
- $ADF^{SETAF} \not\subseteq RFree^{ADF}$
- $BADF \not\subseteq AADF^{+}$ *and* $AADF^{+} \not\subseteq BADF$
- $WV^{ADF} \subsetneq Cln^{ADF}$
- $SV^{ADF} \subsetneq AADF^{+}$
- $(AADF^{+} \cap RFree^{ADF} \cap Cln^{ADF}) \subseteq SV^{ADF}$
- $SV^{ADF} \subsetneq (WV^{ADF} \cap RV^{ADF})$
- $(RV^{ADF} \cap Cln^{ADF}) \subsetneq WV^{ADF}$

5 Translating AFs

The Dung's framework is at the heart of abstract argumentation. Not surprisingly, all other frameworks try to be "backward compatible" with it. By this we understand that given a more advanced framework representing a given AF, the extensions under standard semantics usually coincide for both structures. In this section we will recall how Dung's framework is retrieved by the structures from Section 2, including ADFs. In the majority of the cases, the required modifications will be straightforward. We will also analyze every translation from the point of view of the classification system we have introduced in this work. Consequently, the readers will be given an opportunity to get more accustomed to it before moving on to more complicated approaches. Our running example in this section will be Example 1 from Section 2.1.1. We restate it here for the readers' convenience:

Example 1. Consider the Dung's framework $F = (A, R)$ with $A = \{a, b, c, d, e\}$ and the attack relation $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$, as depicted in Figure 1. It has eight conflict–free extensions in total: $\{a, c\}$,$\{a, d\}$, $\{b, d\}$, $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$ and $\emptyset$. As b is attacked by an unattacked argument, it cannot be defended against it and will not be in any admissible extension. From this $\{a, c\}$, $\{a, d\}$ and $\{a\}$ are complete. We end up with two preferred extensions, $\{a, c\}$ and $\{a, d\}$. However, only $\{a, d\}$ is stable, and $\{a\}$ is the grounded extension.

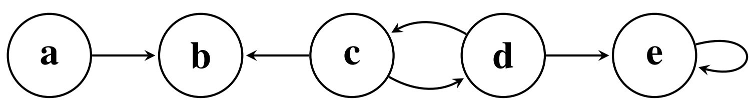


Figure 49: Sample Dung's framework

5.1 AF as SETAF

SETAFs can naturally represent Dung's framework by using attacking sets consisting of just single elements [68, 73]. The correspondence of the semantics follows straightforwardly from the definitions.

Translation 17. *Let $F = (A, R)$ be a Dung's framework. The corresponding SETAF is $SF^F = (A, R')$, where $R' = \{(\{a\}, b) \mid (a, b) \in R\}$.*

Please note that the SETAFs produced by our translation are clearly belong to the Bin^{SETAF} subclass. Moreover, for any Bin^{SETAF} framework we can also easily imagine a Dung's framework from which it can be obtained and thus it is precisely the subclass of AF obtained SETAFs. Since $Bin^{SETAF} \subset Min^{SETAF}$ (Lemma 4.63), the produced frameworks also satisfy the minimal form:

Theorem 5.1. *Let $F = (A, R)$ be an AF and SF^F its corresponding SETAF obtained through Translation 17. SF^F is in minimal form.*

Theorem 5.2. *Let $F = (A, R)$ be a Dung's framework and $SF^F = (A, R')$ its corresponding SETAF obtained through Translation 17. A set of arguments $E \subseteq A$ is a σ–extensions of F, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$ iff it is a σ–extension of SF^F.*

The presented translation and theorem can be moved into our system the following way; we will also include the translation's properties.
Redefinition of Translation 17: *Let Fr^{AF} be the collection of all Dung's frameworks and Bin^{SETAF} the collection of all binary SETAFs, both based on argument domain $\mathcal{U}$. The translation $Tr^{AF}_{SETAF} : Fr^{AF} \to Bin^{SETAF}$ is defined as $Tr^{AF}_{SETAF}((A, R)) = (A, R')$, where $R' = \{(\{a\}, b) \mid (a, b) \in R\}$ for a framework $(A, R) \in Fr^{AF}$.*
Redefinition of Theorem 5.2: *Let $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the identity casting functions for σ. The translation Tr^{AF}_{SETAF} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*
Analysis of Translation 17: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and identity casting functions, the translation Tr^{AF}_{SETAF} is:

- full, target–subclass and injective
- argument domain and structure preserving
- generic, semantics domain preserving and exact
- modular and structural

Translation 17 is classified as basic under the listed semantics and casting functions.

Explanation. The fullness and subclass properties follow from the redefinition of Translation 17. It is also easy to observe that the translation is nothing more than adding brackets to an attack and no two different AFs can be translated into a single SETAF. Consequently, our translation is an injection. The argument domain stays the same and although the binary attacks are now represented by single–element set attacks, no arguments or conflicts are removed and thus the structure is preserved. The translation is also clearly generic and semantics domain preserving. Exactness follows straightforwardly from the redefinition of Theorem 5.2. Since the structure is preserved and every attack is translated independently of another, we can observe that the translation is modular. Finally, as it does not require any semantical notions from neither the source nor the target framework and does not process the attack relation in any complicated manner, we can classify it as structural. All of those properties qualify the Translation 17 as basic. ■

Example 68. Let us come back to the framework $F = (A, R)$, where $A = \{a, b, c, d, e\}$ and $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$, previously described in Example 1. The associated SETAF is $SF^F = (A, R')$, where $R' = \{(\{a\}, b), (\{c\}, b), (\{c\}, d), (\{d\}, c), (\{d\}, e), (\{e\}, e)\}$. We can easily verify that $\emptyset$, $\{a\}$, $\{c\}$, $\{d\}$, $\{a, c\}$ and $\{a, d\}$ are admissible in SF^F. The complete extensions are $\{a\}$, $\{a, c\}$ and $\{a, d\}$, with the first set being the grounded extension and the other two preferred. $\{a, d\}$ is the stable extension of SF^F. We can observe that the produced sets are exactly the same as the ones we would obtain from F.

5.2 AF as AFRA

While in AFRA attacks can be directed both at other attacks and arguments, Dung's framework allows only the latter. Consequently, an AFRA framework corresponding to a given AF is obtained quite trivially [9]. However, since according to AFRA semantics attacks also need to be explicitly included in an extension, precisely retrieving the AF extensions requires more work and the authors resolve this issue by introducing a the AFRA operator $\rightarrow AFRA$. Thus, our semantics casting function will not be an identity anymore and the semantics domain will not be the same.

Translation 18. *Let $F = (A, R)$ be a Dung's framework. The corresponding AFRA is $FR^F = (A, R)$.*

We can observe that the produced AFRA will be a non–recursive one, i.e. it will contain only attacks on arguments:

Theorem 5.3. *Let $F = (A, R)$ be an AF and $FR^F = (A, R)$ its corresponding AFRA obtained through Translation 18. $FR^F \in Rec_0^{AFRA}$ is non–recursive.*

Definition 5.4. Let $FR = (A, R)$ be an AFRA. Given a set of arguments $E \subseteq A$, $E^{\rightarrow AFRA} \triangleq E \cup \{V \in R \mid src(v) \in E\}$.

The work in [9] gives us the following results:

Theorem 5.5. *Let $F = (A, R)$ be a Dung's framework, $FR^F = (A, R)$ its corresponding AFRA obtained through Translation 18 and $\sigma \in \{$complete, preferred, grounded, stable$\}$ a semantics. The set $E \subseteq A \cup R$ is a σ–extension of FR^F iff $E = U^{\rightarrow AFRA}$, where U is a σ–extension of F.*

We will extend the existing results by explaining what happens to the conflict–free and admissible semantics. An AF–obtained AFRA does not have attacks on attacks. Thus, if an argument $a \in A$ is defended by a given set, so is any attack that has a as the source. Since it also holds that if an attack is defended then so is its source by Lemma 2.32, we can conclude that every complete extension E of an AF–obtained AFRA will be of the form $E = (E \cap A)^{\rightarrow AFRA}$ (please refer to [9] for further explanations). Thus, the correspondence between AF and AFRA complete extensions is bijective. However, as not every argument (or attack in the AFRA case) that is defended needs to be included in an admissible extension, we no longer (or rather, not yet) deal with a bijection. For example, in a simple Dung's framework in which a attacks b, the AF admissible extension $\{a\}$ would have two corresponding AFRA sets, namely $\{a\}$ and $\{a, (a, b)\}$. Moreover, due to the fact that it is the attacks that carry out the defeats and their sources do not necessarily have to be present in admissible extensions, we do not even deal with a strong translation anymore when we consider removal casting functions. The extensions would have to be completed with missing arguments in order to preserve admissibility. The loss of strength also occurs when we go down to conflict–freeness. Since an attack has to appear in an AFRA extension in order for the source and target elements to be considered conflicting, two arguments that would not form a conflict–free extension in an AF can be conflict–free in the corresponding AFRA.

Theorem 5.6. *Let $F = (A, R)$ be a Dung's framework and $FR^F = (A, R)$ its corresponding AFRA obtained by Translation 18. If $E \subseteq A$ is an admissible (conflict–free) extension of F, then $E^{\rightarrow AFRA}$ is an admissible (conflict–free) extension of FR^F. If $E' \subseteq A \cup R$ is an admissible (conflict–free) extension of FR^F, then $E' \cap A$ might not be admissible (conflict–free) in F.*

Example 69. Let us consider a simple AF $F_1 = (\{a, b, c\}, \{(a, b), (b, c)\})$. We can observe that in its associated AFRA, the argument c is defeated by the (b, c) attack, which in turns is (indirectly) defeated by the attack (a, b). Moreover, there are no attacks defeating (a, b) in the framework. Consequently, we can show that $\{c, (a, b)\}$ is admissible and conflict–free in the associated AFRA. However, the corresponding set $\{c\}$ is not admissible in F_1, as c is not defended from b.

We can also look at the AF $F_2 = (\{a\}, \{(a, a)\})$. The set $\{a\}$ is conflict–free in its associated AFRA, even though it is not conflict–free in F_2.

With this, we can finally put the AF–AFRA translation into our system. We already know that the produced AFRAs are non–recursive. However, for every AFRA that has only attacks between arguments we can easily construct an AF producing it. Thus, Rec_0^{AFRA} is the most accurate codomain for the translation.

Redefinition of Translation 18: *Let Fr^{AF} be the collection of all Dung's frameworks and Rec_0^{AFRA} the collection of all AFRAs without recursive attacks, both on domain $\mathcal{U}$. The translation $Tr^{AF}_{AFRA} : Fr^{AF} \rightarrow Rec_0^{AFRA}$ is defined as $Tr^{AF}_{AFRA}((A, R)) = (A, R)$ for $(A, R) \in Fr^{AF}$.*

Redefinition of Theorems 5.5 and 5.6: *Let $\sigma \in \{$complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_σ the removal casting functions for σ defined as $SC^X_\sigma(E) = E \cap A$, where $X = (A, R) \in Fr^{AF}$ is a framework and $E \in \sigma(Tr^{AF}_{AFRA}(X))$. The translation Tr^{AF}_{AFRA} is strong and semantics bijective under (σ, SC^{Tr}_σ).*

Let $\delta \in \{$conflict–free, admissible$\}$ be a semantics and SC^{Tr}_δ the removal casting functions defined as for σ. The translation Tr^{AF}_{AFRA} is $\subseteq$–weak under (δ, SC^{Tr}_δ).

Analysis of Translation 18: Under the conflict–free, admissible, complete, preferred, grounded and stable semantics and the removal casting functions the translation Tr^{AF}_{AFRA} is:

- full, target–subclass and injective
- argument domain and structure preserving

- generic and semantics domain altering
- modular and structural

Under the complete, preferred, grounded and stable semantics and the removal casting functions the translation Tr^{AF}_{AFRA} is faithful.

The translation Tr^{AF}_{AFRA} is classified as basic under the listed semantics and casting functions.

Explanation. Just like in the SETAF case, most of the mentioned properties can be easily observed from the redefinitions of the translation and respective theorems. The translation in no way affects the framework and is trivially modular and structural. The semantics domain altering property is a result of the fact that AF extensions contain arguments only, while AFRA sets also include attacks. The faithfulness of the translation under the listed semantics can be easily observed from the redefinitions of Theorems 5.5 and 5.6. ■

Example 70. Let us come back to the framework $F = (A, R)$ previously described in Example 1, where $A = \{a, b, c, d, e\}$ and $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$. Its associated AFRA FR^F looks exactly the same. For the sake of simplicity, we will label the attacks: $\alpha = (a, b)$, $\beta = (c, b)$, $\gamma = (c, d)$, $\delta = (d, c)$, $\epsilon = (d, e)$ and $\zeta = (e, e)$. Due to the amount of possible admissible and conflict–free extensions, we will not focus on these semantics. FR^F has three complete extensions; $\{a, \alpha\}$, $\{a, c, \alpha, \beta, \gamma\}$ and $\{a, d, \alpha, \delta, \epsilon\}$. We can observe that the sets $\{a\}$, $\{a, c\}$ and $\{a, d\}$ associated with them are the complete extensions of F. The set $\{a, \alpha\}$ is the grounded extension of FR^F, while $\{a, c, \alpha, \beta, \gamma\}$ and $\{a, d, \alpha, \delta, \epsilon\}$ are preferred. They again correspond to the desired results in F. Finally, we can observe that $\{a, c, \alpha, \beta, \gamma\}$ is not stable; it defeats neither e nor ζ. Fortunately, $\{a, d, \alpha, \delta, \epsilon\}$ meets the stability criterion, and again corresponds to the extension $\{a, d\}$ produced by F.

5.2.1 Improvements

In this section we will discuss certain possible changes to the translation that would improve its strength w.r.t. admissible semantics. Although the current approach is already satisfactory based on how other semantics behave, the discussion presented here is valuable for understanding how changing the casting function can improve the range of our translation without affecting the way the target framework is obtained.

Theorem 5.7. *Let $F = (A, R)$ be a Dung's framework, $FR^F = (A, R)$ its corresponding AFRA obtained by Translation 18 and $\sigma \in \{$admissible, complete, preferred, grounded, stable$\}$ a semantics. If $E \subseteq A$ is a conflict–free extension of F, then $E^{\rightarrow AFRA}$ is a conflict–free extension of FR^F. If $E' \subseteq A \cup R$ is a conflict–free extension of FR^F, then $E = (E' \cap A) \cup \{src(x) \mid x \in E' \cap R\}$ might not be conflict–free in F. If E is a σ–extension of F, then $E^{\rightarrow AFRA}$ is a σ–extension of FR^F. If $E' \subseteq A \cup R$ is a σ–extension of FR^F, then $E = (E' \cap A) \cup \{src(x) \mid x \in E' \cap R\}$ is a σ–extension of F.*

Redefinition of of Theorem 5.7: *Let $\sigma \in \{$admissible, complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the casting functions for it defined as $SC^{X}_{\sigma}(E) = (E' \cap A) \cup \{src(x) \mid x \in E' \cap R\}$ where $X = (A, R) \in Fr^{AF}$[17] and $E \in \sigma(Tr^{AF}_{AFRA}(X))$. The translation Tr^{AF}_{AFRA} is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under conflict–free semantics and the defined casting functions. It is semantics bijective under complete, preferred, grounded and stable semantics and the defined casting functions.*

To show that the translation is not semantics bijective under admissibility and the defined casting function, we can come back to the example from the proof of Theorem 5.6. Among other AFRA admissible extensions, the framework $(\{a, b, c\}, \{(a, b), (b, c)\})$ produces sets $\{a\}$, $\{(a, b)\}$ and $\{a, (a, b)\}$. Using the defined casting function, they will all be mapped to an AF admissible extension $\{a\}$.

Please note that the presented casting function technically qualifies as a two–step one, consisting of removal and extraction components. Consequently, as a whole, it does not qualify as any of the main types. For the admissible semantics, the extensions before and after casting will become incomparable. However, for the semantics that are at least complete, the casting will basically come back to the original approach from the redefinitions of Theorems 5.5 and 5.6. Thus, it can be reclassified as a removal again. Please note that the only reason we permit it is the fact that the attacks show up in the extensions and thus the extraction part of the casting process does not require access to any auxiliary structure that holds the contents of arguments.

Analysis of Translation 18: Under the conflict–free, admissible, complete, preferred, grounded and stable semantics and casting functions in the redefinition of Theorem 5.7, the translation Tr^{AF}_{AFRA} is:

- full, target–subclass and injective
- argument domain and structure preserving
- generic and semantics domain altering
- modular and structural

Under the complete, preferred, grounded and stable semantics and the defined casting functions, the translation is faithful.

The translation Tr^{AF}_{AFRA} is classified as basic under the listed semantics and casting functions.

Explanation. The main properties hold as with the previous analysis, since the translation itself does not change. For the complete semantics the defined casting functions are in fact just removals and behave exactly like the ones from redefinition of Theorems 5.5 and 5.6 (see proof of Theorem 5.7). Therefore, we again have faithfulness under complete, preferred, grounded and stable semantics. ■

[17] We can also take (A, R) as an AFRA, since the structure of the two frameworks is identical.

Example 71. Let us consider a Dung's framework $F_1 = (\{a, b, c\}, \{(a, b), (b, c)\})$. The set $\{c, (a, b)\}$ is admissible in its associated AFRA. Using the removal casting function, it was transformed into the set $\{c\}$, which was not admissible in F_1. However, by performing the removal–extraction function, we obtain the set $\{a, c\}$, which is indeed admissible in F_1.

We can now look again at the framework $(\{a, b, c, d, e\}, \{\alpha = (a, b), \beta = (c, b), \gamma = (c, d), \delta = (d, c), \epsilon = (d, e), \zeta = (e, e)\})$ from Example 70. Its complete extensions were $\{a, \alpha\}$, $\{a, c, \alpha, \beta, \gamma\}$ and $\{a, d, \alpha, \delta, \epsilon\}$. Independently of whether we use the original casting function or its improvement, we still obtain the sets $\{a\}$, $\{a, c\}$ and $\{a, d\}$.

At this point it is reasonable to ask whether it is even possible to create an AF–AFRA translation that is faithful for admissible semantics. Our answer is: not likely. There are two main issues that need to be handled in order for that to be possible; we need to make sure that if an attack is accepted, then so is its source, and if an argument is accepted, then so are the attacks it carries out. If we would like to connect the acceptance of one argument with another in AFs, we could use defense against self–attacking auxiliary arguments. However, this approach in AFRAs would only cause including new attacks in an extension and not necessarily new arguments. For example, consider a framework $(\{a, b\}, \{(a, b)\})$. The AF admissible extension $\{a\}$ has two corresponding AFRA extensions using removal casting function – $\{a\}$ and $\{a, (a, b)\}$ – or three with the source including one – $\{a\}$, $\{(a, b)\}$ and $\{a, (a, b)\}$. Trying to tie (a, b) to a by the use of defense would require adding an argument c and attacks (c, c), (a, c) and $(c, (a, b))$. Although from the usual, argument–centered perspective a would be "defending" the (a, b) attack from c, in AFRA it would be the (a, c) attack doing it, and thus presence of the source is again not required in the extension. Thus, we are not aware of any method that would allow us to tackle the first issue and are not convinced that a direct faithful translation for admissible semantics exists. However, it appears that relating AF not to AFRA but AFRA–produced AFNs (see Translation 35) could address this problem. This approach is weak under the admissible semantics, i.e. the produced AFN does not return all the original extensions. However, the ones that actually are created, appear to be exhibiting the property that we wanted here (see Section 7.4 for a discussion).

Considering the possibility of creating a translation and casting function that would be exact under any semantics, it does not appear to be very likely, perhaps with the exception of unique status ones such as grounded. For exactness we need identity casting functions, which implies the same semantics domains. Unless we produce AFRAs that contain no attacks, there is no chance for that. Moreover, having no attacks means a framework can give us only a single extension, which drastically limits our choice of semantics. The way to produce an AF–AFRA exact translation for the grounded semantics would be to compute the grounded AF extension first and delete any other arguments and all of the attacks from the framework. We thus have an exact, but lossy translation, and semantical to the point of being practically useless.

5.3 AF as EAF

The additional attacks – defense attacks – in EAFs were introduced as means to express and argue about preferences in the framework. Without them, the framework is just a Dung's framework and not surprisingly, AFs translate into EAFs with the empty set of defense attacks.

Translation 19. *Let $F = (A, R)$ be a Dung's framework. The corresponding EAF is $EF^F = (A, R, \emptyset)$.*

Just like in the previous translations, not every type of EAF can be produced from an AF. Only those without defense attacks will be created, and for every such EAF with $D = \emptyset$, we can construct an AF producing it. Thus, $NDef^{EAF}$ is the most accurate codomain description for this translation. The fact that there are no defense attacks in the framework also means that the target EAS belongs to two convenient subclasses by Lemmas 4.67:

Theorem 5.8. *Let $F = (A, R)$ be an AF and $EF^F = (A, R, \emptyset)$ its corresponding EAF obtained through Translation 19. EF^F is in $NDef^{EAF}$. It is also bounded hierarchical and (strongly) consistent.*

Since there are no defense attacks in the framework, every attack will result in a defeat w.r.t. any set of arguments. Moreover, various definitions can be simplified:

Theorem 5.9. *Let $EF = (A, R, D) \in NDef^{EAF}$ be an EAF without defense attacks. The following holds:*

- *an argument a defeats an argument b w.r.t. any set of arguments E iff $(a, b) \in R$*
- *a set of arguments $E \subseteq A$ is conflict–free extension of EF iff there are no $a, b \in E$ s.t. aRb*
- *given a set of arguments $E \subseteq A$, a set containing a pair $\{(x, y)\}$ s.t. x defeats$_E$ y is a reinstatement set on E for the defeat$_E$ by x on y iff $x \in E$.*
- *an argument $a \in A$ is acceptable w.r.t. a set of arguments $E \subseteq A$ iff for every argument b s.t. bRa, there is $c \in E$ s.t. cRb*
- *a set of arguments E is a stable extension of EF iff for every argument $b \notin E$, $\exists a \in E$ s.t. aRb*

This brings us to a rather natural result concerning the behavior of the semantics:

Theorem 5.10. *Let $F = (A, R)$ be a Dung's framework and $EF^F = (A, R, \emptyset)$ its corresponding EAF obtained through Translation 19. A set of arguments $E \subseteq A$ is a σ–extension of F, where $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ iff it is a σ–extension of EF^F.*

With the translation at hand, we can show how the results look when put into our system.

Redefinition of Translation 19: *Let Fr^{AF} be the collection of all Dung's frameworks an $NDef^{EAF}$ the collection of all EAFs with empty set of defense attacks, both on domain $\mathcal{U}$. The translation $Tr^{AF}_{EAF} : Fr^{AF} \rightarrow NDef^{EAF}$ is defined as $Tr^{AF}_{EAF}((A, R)) = (A, R, \emptyset)$ for $(A, R) \in Fr^{AF}$.*

Redefinition of Theorem 5.10: *Let $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} its identity casting functions. The translation Tr^{AF}_{EAF} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*

Analysis of Translation 19: Under the conflict–free, admissible, complete, preferred, grounded and stable semantics and identity casting functions, the translation Tr^{AF}_{EAF} is:

- full, target–subclass and injective
- argument domain and structure preserving
- generic, semantics domain preserving and exact
- modular and structural

The translation Tr^{AF}_{EAF} is classified as basic under the listed semantics and casting functions.

Explanation. The fact that the translation is full and target–subclass can be already seen in the redefinition of Translation 19. Since the only change to the framework is the addition of the empty set accounting for the defense attacks, it is easy to see that no two different AFs will be assigned a single EAF. Thus, our translation can be easily shown to be injective, argument domain and structure preserving. The amount of handled translations shows that the approach is also generic; since the argument domain stays the same and in both frameworks extensions contain arguments only, the semantics domains also stay the same. Exactness follows straightforwardly from Theorem 5.10. Modularity can also be easily shown due to the fact that the set of arguments and attacks is in no way affected by the translation. The simplicity of this translation qualifies it as basic. ■

Example 72. Let us come back to the framework $F = (A, R)$ previously described in Example 1, where $A = \{a, b, c, d, e\}$ and $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$. The associated EAF is simply $EF^F = (A, R, \emptyset)$. Due to the lack of any defense attacks, every attack becomes a defeat$_E$ w.r.t. any $E \subseteq A$. Moreover, every attack forms a trivial reinstatement set for itself. Therefore, we can easily verify that $\emptyset$, $\{a\}$, $\{c\}$, $\{d\}$, $\{a, c\}$ and $\{a, d\}$ are admissible extensions of EF^F, which is exactly what F produces. $\{a\}$ is the grounded extension and $\{a, c\}$, $\{a, d\}$ are preferred. These three sets are also the complete extensions of EF^F. Finally, $\{a, d\}$ is the stable extension of EF^F. We can observe that the produced sets are exactly the same as the ones we would obtain from F.

Please note that translating an AF into an EAFC proceeds in exactly the same manner as for EAFs and yields the same results. Thus, we will not give the full definitions here. It is only worth noticing that due to the fact that the produced EAFCs do not have any defense attacks, the framework is also in minimal form.

5.4 AF as BAF

We now reach the first framework with support. The way BAFs generalize AFs is quite straightforward. Although it is only briefly stated in [28], we retrieve the Dung's framework by just adding the empty support relation:

Translation 20. *Let $F = (A, R)$ be a Dung's framework. The corresponding BAF is $BF^F = (A, R, S)$, where $S = \emptyset$.*

We can observe that due to the lack of any support in the framework, no indirect attacks of any type will be created. Consequently, trying to parametrize +conflict–freeness and defense with any types of conflicts will always boil down to using the direct attacks in R only. Moreover, we can recall Lemma 4.69 to see that we no longer need to consider safety and support closure:

Lemma 4.69. *Let $BF = (A, R, S)$ be a BAF with $S = \emptyset$. Then $R^{ind} = \emptyset$. Every set of arguments $E \subseteq A$ is closed and inverse closed under S and if E is +conflict–free, it is also safe.*

Therefore, we can observe that the d–/s–/c–/i–classification of the BAF semantics is not really necessary here. However, technically speaking, only the d–family is fully defined under the constraint that we use the same parametrization for +conflict–freeness and defense. Therefore, we will provide results only for these semantics, though it should be clear that the other approaches will behave in the same way.

Theorem 5.11. *Let $F = (A, R)$ be an AF and $BF^F = (A, R, S)$ its corresponding BAF obtained through Translation 20. Let $R', R'' \subseteq R^{ind}$ be arbitrary sets of indirect attacks. A set of arguments $E \subseteq A$ is conflict–free in F iff it is +conflict–free in BF^F w.r.t. R'. $E \subseteq A$ is stable in F iff it is stable in BF^F w.r.t. R'. E is a σ–extension of AF, where $\sigma \in \{$admissible, complete, preferred, grounded$\}$ iff it is a d–σ–extension of BF^F w.r.t. (R', R'').*

We can now put the results into our system.
Redefinition of Translation 20: *Let Fr^{AF} be the collection of all Dung's frameworks and $NSup^{BAF}$ the collection of all BAFs the empty support relation, both on domain $\mathcal{U}$. The translation $Tr^{AF}_{BAF} : Fr^{AF} \rightarrow NSup^{BAF}$ is defined as $Tr^{AF}_{BAF}((A, R)) = (A, R, \emptyset)$ for $(A, R) \in Fr^{AF}$.*
Redefinition of Theorem 5.11: *Let $\sigma^{AF} \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ be an AF semantics semantics and $\sigma^{BAF} \in \{$+conflict–free,*

d–admissible, d–complete, d–preferred, d–grounded, d–stable$\}$ a similar BAF semantics with arbitrary $R', R'' \subseteq R^{ind}$. Let SC^{Tr}_{σ} be the identity casting functions for σ. The translation Tr^{AF}_{BAF} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.

Analysis of Translation 20: Under the (+) conflict–free, (d–) admissible, (d–) complete, (d–) preferred, (d–) grounded and stable semantics with arbitrary parameterizing $R', R'' \subseteq R^{ind}$ and identity casting functions, the translation Tr^{AF}_{BAF} is:

- full, target–subclass and injective
- argument domain and structure preserving
- generic, semantics domain preserving and exact
- modular and structural

The translation Tr^{AF}_{BAF} is classified as basic under the listed semantics and casting functions.

Explanation. The functional and structural properties of the translation can be easily observed from the redefinition of Translation 20. The amount of handled semantics qualify the approach as generic. Its exactness follows from the redefinition of Theorem 5.11. The translation is trivially structural. Since the only change is the addition of the empty set accounting for the support relation and no arguments or attacks are added or removed, modularity of the approach follows easily. All of this qualifies the translation Tr^{AF}_{BAF} as basic. ■

Example 73. Let us come back to the framework $F = (A, R)$ from Example 1, where $A = \{a, b, c, d, e\}$ and $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$. The associated BAF is $BF^F = (A, R, \emptyset)$. We can observe that no indirect conflicts of any type can be produced in this framework. Therefore, independently of the used parametrization, the conflict–freeness and defense in BF^F will boil down to considering only direct conflicts, which is equivalent to conflict–freeness and defense in F. Consequently, we can easily verify that $\emptyset$, $\{a\}$, $\{c\}$, $\{d\}$, $\{a, c\}$ and $\{a, d\}$ are d–/i–/s–/c–admissible in BF^F. The last two sets are also d–/i–/s–/c–preferred, with $\{a, d\}$ being stable. Finally, $\{a\}$, $\{a, c\}$ and $\{a, d\}$ are d–complete and $\{a\}$ is d–grounded. We can observe that the produced extensions correspond to the ones we would obtain from F.

5.5 AF as AFN

AFNs retrieve AFs in a fashion very similar to BAFs and EAFs [69]. Again, it suffices to assume that no support relation occurs in the framework:

Translation 21. *Let $F = (A, R)$ be a Dung's framework. The corresponding AFN is $FN^F = (A, R, N)$, where $N = \emptyset$.*

The AF–produced AFNs belong to the $NSup^{AFN}$ subclass, which exhibits all of the normal forms we have analyzed in Section 4. All of them follow trivially from the fact that no support is present:

Theorem 5.12. *Let $F = (A, R)$ be an AF and $FN^F = (A, R, N)$ its corresponding AFN obtained through Translation 21. FN^F is minimal, (strongly) consistent, and weakly, relation and strongly valid.*

Hence, our AFNs are both elementary and well–structured. It is also worth mentioning, that in AFNs without support, conflict–free and strongly coherent extensions coincide and thus AF conflict–free extensions can be obtained from two semantics. Moreover, the coherent extensions become somewhat pointless – they simply return the power set of arguments.

Theorem 5.13. *Let $F = (A, R)$ be a Dung's framework and $FN^F = (A, R, N)$ its corresponding AFN obtained through Translation 21. A set of arguments $E \subseteq A$ is a σ–extensions of F, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$ iff it is a σ–extension of FN^F.*

We can now put the existing results into our system. Please note that the AF–AFN translation exhibits the same properties as the AF–BAF one. Therefore, we will omit further explanations of the properties.
Redefinition of Translation 21: *Let Fr^{AF} be the collection of all Dung's frameworks and $NSup^{AFN}$ the collection of all AFNs with empty support relation, both on argument domain $\mathcal{U}$. The translation $Tr^{AF}_{AFN} : Fr^{AF} \rightarrow NSup^{AFN}$ is defined as $Tr^{AF}_{AFN}((A, R)) = (A, R, \emptyset)$ for $(A, R) \in Fr^{AF}$.*
Redefinition of Theorem 5.13: *Let $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} its identity semantics casting functions. The translation Tr^{AF}_{AFN} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*
Analysis of Translation 21: Under the conflict–free, admissible, complete, preferred, grounded and stable semantics and identity casting functions, the translation Tr^{AF}_{AFN} is:

- full, target–subclass and injective
- argument domain and structure preserving
- generic, semantics domain preserving and exact
- modular and structural

The translation Tr^{AF}_{AFN} is classified as basic under the listed semantics and casting functions.

Example 74. We will again continue with the AF from Example 1, i.e. $F = (A, R)$ where $A = \{a, b, c, d, e\}$ and $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$. The associated AFN is $FN^F = (A, R, \emptyset)$. We can observe that every argument $a \in A$ has a trivial (minimal) coherent set $\{a\}$. Therefore, a set of arguments $E \subseteq A$ attacks all coherent sets of a given argument in FN^F iff it attacks this argument directly. Moreover, any subset of A is coherent in FN^F. Consequently, it is easy to check that the sets $\emptyset$, $\{c\}$, $\{d\}$, $\{a\}$, $\{a, c\}$ and $\{a, d\}$ are both strongly coherent and defend their members. Thus, they are the admissible extensions of FN^F. The latter three are also complete, with $\{a\}$ being grounded and $\{a, c\}$, $\{a, d\}$ preferred. Finally, $\{a, d\}$ is stable in FN^F. These answers are in agreement with the extensions produced by F.

5.6 AF as EAS

Converting an AF into an EAS requires two things. First of all, we need to transform the binary attack to group form, which is a very simple modification. Then, unlike in any other framework, we need to adapt the support relation. In EASs, the unsupported arguments are not considered valid, and leaving an empty support relation would give us a framework returning $\{\eta\}$ or $\emptyset$ as our extensions. Validity comes from evidence, and therefore we need to both add the evidence argument to the structure and include the support from it to every AF argument. The translation is basically the same as first transforming the AF into SETAF, and then exploiting the SETAF–EAS translation (see [73] or Section 6.4). This brings us to the following formulation:

Translation 22. *Let $F = (A, R)$ be a Dung's framework. The corresponding EAS is $ES^F = (A \cup \{\eta\}, R', E)$, where $R' = \{(\{a\}, b) \mid (a, b) \in R\}$ and $E = \{(\{\eta\}, a) \mid a \in A\}$.*

The EASs produced by AFs belong to the intersection of various subclasses of EASs. First of all, we produce attack binary and support binary EASs. Moreover, every supporting set is just $\{\eta\}$ and every argument receives support from it. The intersection of $ABin^{EAS}$, $EvSup^{EAS}$ and $AllSup^{EAS}$ finally gives us the description of the subclass AF^{EAS} of AF–produced EASs. Due to Lemma 4.73, AF^{EAS} satisfies all of the EAS normal forms from Section 4:

Theorem 5.14. *Let $F = (A, R)$ be a Dung's framework and $ES^F = (A \cup \{\eta\}, R', E)$ its corresponding EAS obtained through Translation 22. ES^F is minimal, consistent, weakly, relation and strongly valid.*

We can observe that the original extensions can be retrieved by removing η from the target answers:

Theorem 5.15. *Let $F = (A, R)$ be a Dung's framework and $ES^F = (A \cup \{\eta\}, R', E)$ its corresponding EAS obtained through Translation 22. A set of arguments $S \subseteq A$ is a σ–extensions of F, where $\sigma \in$ {conflict–free, admissible, preferred, complete, grounded, stable} if $S \cup \{\eta\}$ is a σ–extension of ES^F. Moreover, if $S \subseteq A$ is conflict–free in F, then it is also conflict–free in ES^F. A set of arguments $S' \subseteq A \cup \{\eta\}$ is a σ–extension of ES^F if $S' \setminus \{\eta\}$ is a σ–extension of F.*

Please note that $\emptyset$ is also trivially admissible and conflict–free in ES^F. Moreover, due to the additional argument η, a bigger number of conflict–free extensions will be created. However, filtering out η brings back exactly the sets created in F. We can also observe that adding η to the sets in the powerset of A will give us all the self–supporting sets in the target ES. From the properties of the evidential frameworks, it is easy to see that every complete extension will contain η, which is not the case in admissible semantics (note the empty set). Therefore, from complete semantics on, the relation between the target and source extensions is in fact one–to–one.

Redefinition of Translation 22: *Let Fr^{AF} be the collection of all Dung's frameworks on domain $\mathcal{U}$ and AF^{EAS} the EASs with single element attacks and every argument being supported by and only by evidence on domain $\mathcal{U} \cup \{\eta\}$. The translation $Tr^{AF}_{EAS} : Fr^{AF} \rightarrow AF^{EAS}$ is defined as $Tr^{AF}_{EAS}((A, R)) = (A', R', E)$ for $(A, R) \in Fr^{AF}$, where $A' = A \cup \{\eta\}$, $R' = \{(\{a\}, b) \mid (a, b) \in R$ and $E = \{(\{\eta\}, a) \mid a \in A\}$.*

Redefinition of Theorem 5.15: *Let $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} their removal casting functions defined as $SC^{X}_{\sigma}(S) = S \cap A$, where $X = (A, R) \in Fr^{AF}$ and $S \in \sigma(Tr^{AF}_{EAS}(X))$. The translation Tr^{AF}_{EAS} is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is semantics bijective under the complete, preferred, grounded and stable semantics and their removal casting functions.*

Analysis of Translation 22: Under the conflict–free, admissible, complete, preferred, grounded and stable semantics and the removal casting functions, the translation Tr^{AF}_{EAS} is:

- full, target–subclass and injective
- weakly argument domain altering, argument introducing, induced support introducing and attack relation preserving
- generic and weakly semantics domain altering
- modular and semi–structural

Under the complete, preferred, grounded and stable semantics and the removal casting functions, the translation is faithful.

The translation Tr^{AF}_{EAS} is classified as basic under the listed semantics and casting functions.

Explanation. Since any Dung's framework can be translated into an EAS and AF^{EAS} clearly does not account for all the possible evidential frameworks, the translation is full and target–subclass. Clearly, it cannot be the case that two different AFs produce the same EAS – they would differ in the same manner as their source frameworks do. Thus, our approach is also injective. The structural properties of Tr^{AF}_{EAS} can be easily observed from the way Translation 22 is defined. Due to the amount of handled semantics, the translation is generic. From Theorem 5.15 we can also see that the semantics domain is only weakly altered. The transformation can be classified as semi–structural. Although adding the evidence argument is just a structural modification that is required by the definition of EASs, the fact that we add support from it to every other argument comes from our knowledge on how unsupported arguments would be treated by the EAS semantics. Faithfulness is a result of the redefinition of Theorem 5.15. Tr^{AF}_{EAS} is also easily seen to be modular. ■

Please note that under the admissible semantics, the translation Tr^{AF}_{EAS} is actually weakly faithful according to the classification from [42] (see also Section 3.2.3). This means that there is a bijection between the admissible extensions of a source AF and the admissible extensions of the produced EAS if we exclude the $\emptyset$ from the latter.

Example 75. We will again continue with the AF from Example 1, i.e. $F = (A, R)$ where $A = \{a, b, c, d, e\}$ and $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$. The associated EAS is $ES^F = (A \cup \{\eta\}, R, E)$, where $E = \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, c), (\{\eta\}, d), (\{\eta\}, e)\}$. For every argument $a \in A$, $\{\eta, a\}$ is a minimal self–supporting set. In fact, empty set and every subset of $A \cup \{\eta\}$ that contains η are self–supporting in ES^F. Since η cannot be attacked and every attack in ES is binary, acceptability in ES^F boils down to verifying whether every attacker of a given argument is also (directly) attacked. Therefore, we can show that $\emptyset$, $\{\eta\}$, $\{\eta, c\}$, $\{\eta, d\}$, $\{\eta, a\}$, $\{\eta, a, c\}$ and $\{\eta, a, d\}$ are admissible in ES^F. By removing η, we obtain all the admissible extensions of F, though we can observe that $\emptyset$ is obtained both from $\emptyset$ and $\{\eta\}$. All of the sets $\{\eta, a\}$, $\{\eta, a, c\}$ and $\{\eta, a, d\}$ are complete in ES^F. Removing the evidence argument again retrieves the desired extensions of F. However, this time we can observe that the relation is one–to–one. The first extension is also the grounded one and the remaining two are preferred. Finally, $\{\eta, a, d\}$ is the stable set of ES^F; we can observe that $\{\eta, a, c\}$ does not attack e, even though it possesses a self–supporting set on $A \cup \{\eta\}$.

5.6.1 Improvements

Although faithful translations are quite satisfactory, we could ask ourselves whether it is possible to create a full and exact translation from AFs to EASs. The general answer is: no. EASs require the presence of an evidence argument in the framework. In our approach, we have added it as a separate argument and weakly altered the argument domain in order to make sure that the source framework did not contain such an argument before. The reason for that is that not in every AF we can find a suitable argument that could play the role of evidence. For example, we can imagine a simple framework $(\{a, b\}, \{(a, b), (b, a)\})$ with two symmetrically attacking arguments. The η argument is not allowed to attack any other argument and thus neither a nor b can take over its function. Consequently, the introduction of another argument is unavoidable. This brings us to the following result, which in fact holds for any other framework we are considering in this work:

Theorem 5.16. *Let Fr^{AF} be the collection of all AFs on a domain $\mathcal{U}^{AF}$ and Fr^{EAS} the collection of all EASs on a domain $\mathcal{U}^{EAS}$. There exists no full translation from Fr^{AF} to Fr^{EAS} that is exact under admissible, complete, preferred, grounded and stable semantics and their identity casting functions.*

However, we can observe that since conflict–freeness does not take into account the support relation, an exact approach is possible in this case – we simply do not include η or any support from it.

5.7 AF as ADF

Although it was already stated before that ADFs properly generalize the Dung's framework, not in all cases formal proofs were given. By "properly" we mean that the extensions under a σ–semantics of a given Dung's framework are the same as of the corresponding ADF. In this section we will show how to translate AFs into ADFs and that the extensions of both structures coincide under the semantics we have presented in Section 2.3.5. Results for other semantics families can be found in [21] and are based on the same translation that we will present here.

The translation of AFs to ADFs is pretty straightforward. Let $a \in A$ be an argument and $X = \{x_1, .., x_n\}$ the set of arguments attacking a. Whenever any of $x_i's$ is present in a given set, a cannot be accepted. Only when all of them are absent, we can assume a. Consequently, for any set $Y \subseteq A$ s.t. $Y \cap X \neq \emptyset$, the condition of a is out. Since the acceptance condition is technically defined only for the set of parents, which in this case is precisely X, it is easy to see that only $\emptyset$ will be mapped to in. The boolean version is just $att_a = \neg x_1 \wedge ... \wedge \neg x_n$; we will abbreviate this construction with $\bigwedge \neg X$.

Translation 23. *Let $F = (A, R)$ be a Dung's framework and $X^a = \{x_1, ..., x_{n_a}\}$ the collection of attackers of an argument $a \in A$. The ADF corresponding to F is $D^F = (A, R, C)$, where $C = \{C_a\}_{a \in A}$ and every C_a is created in the following way:*

- *Functional form: $C_a(\emptyset) = in$ and for all nonempty $B \subseteq X^a$, $C_a(B) = out$*
- *Propositional form: $C_a = att_a = \bigwedge \neg X^a$. In case X^a is empty, it is simply $\top$*

Remark. Let (A, R, C) be an ADF obtained from a Dung's framework (A, R) by the translation above. The acceptance condition for an argument $s \in S$ can be equivalently described as $\bigwedge \neg par(s)$, or simply $\top$ if $par(s) = \emptyset$. It is easy to see that every decisively in interpretation will map all arguments of $par(s)$ to $\mathbf{f}$ and a minimal one will consist only of these mappings. Consequently, any decisively out interpretation will assign $\mathbf{t}$ to at least one element of $par(s)$. If the condition is $\top$ (and $par(s) = \emptyset$) then obviously no decisively out interpretation exists. It is also worth noticing that in this case, every interpretation that falsifies the condition of s is at the same time decisive for s.

Before we continue showing how semantics behave after the translation, we would first like to note that the sub–semantics classification of ADFs collapses on Dung–style frameworks (see Definition 4.74). Since the system is built around the positive dependency cycles and the Dung's framework does not use any support relations, the fact that Dung–style ADFs are also AADF$^+$s should not be surprising. Moreover, the are also BADFs, and the simplicity of AF produced ADFs qualify them for all of the normal forms we have described in Section 4. Thus, by Lemma 4.77, we can conclude the following:

Theorem 5.17. *Let $F = (A, R)$ be a Dung's framework and $D^F = (A, C)$ its corresponding ADF obtained through Translation 23. Then D^F is an AADF$^+$ and a BADF. It is also redundancy–free, weakly, relation and strongly valid and cleansed.*

We can now introduce certain theorems that will simplify the proof on the relation between the extensions of AFs and their corresponding ADFs. Since all the extension–based sub–semantics coincide by Theorem 2.172, we do not need to analyze the acyclic semantics.

Theorem 5.18. *Let $F = (A, R)$ be a Dung's framework, $D^F = (A, R, C)$ its corresponding ADF obtained through Translation 23 and $E \subseteq A$ a set of arguments. The following holds:*

- *E^- in F equals the union of parents of all arguments in E in D^F.*
- *E is conflict–free in F iff it is conflict–free in D^F.*
- *if E is conflict–free, then E^+ in F coincides with the discarded set of E in D^F.*

Theorem 5.19. *Let $F = (A, R)$ be a Dung's framework, $D^F = (A, R, C)$ its corresponding ADF obtained through Translation 23. Let $E \subseteq A$ be a conflict–free set of F and D^F and $a \in A$ an argument. E defends a in F iff a is decisively in w.r.t. v_E in D^F.*

The above brings out to the following, final result. Please note that by Theorems 2.172 and 5.17 our classification collapses, it does not really matter what type of prefixing we assume.

Theorem 5.20. *Let $F = (A, R)$ be a Dung's framework and $D^F = (A, R, C)$ its corresponding ADF obtained through Translation 23. A set of arguments $E \subseteq A$ is a conflict–free extension of F iff it is (pd–acyclic) conflict–free in D^F. $E \subseteq A$ is a stable extensions of F iff it is (stable) model of D^F. $E \subseteq A$ is a grounded extensions of F iff it is (acyclic) grounded in D^F. $E \subseteq A$ is a σ extensions of F, where where $\sigma \in \{$admissible, preferred, complete$\}$ iff it is an xy–σ–extension of D^F for $x, y \in \{a, c\}$.*

We can now put the results into our system:
Redefinition of Translation 23: *Let Fr^{AF} be the collection of all Dung's frameworks and ADF^{AF} the AF–style ADFs, both on domain $\mathcal{U}$. The translation $Tr^{AF}_{ADF} : Fr^{AF} \rightarrow ADF^{AF}$ is defined as $Tr^{AF}_{ADF}((A, R)) = (A, R, C)$ for $(A, R) \in Fr^{AF}$, where $C = \{C_a \mid a \in A\}$ and given the set $X \subseteq A$ of attackers of a, C_a is defined as a) $C_a(\emptyset) = in$ and for all nonempty $B \subseteq X$, $C_a(B) = out$; or b) $C_a = \top$ if $X = \emptyset$ and $C_a = \bigwedge \neg X$ otherwise.*

Please note that due to the difference in the way semantics are called and the fact that a given AF semantics can coincide more than with one ADF semantics, we need to use the similarity notion (see Definition 3.2).
Redefinition of Theorem 5.20: *Let $\sigma^{AF} \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ be an AF semantics and let $\sigma^{ADF} \in \{$conflict–free, pd–acyclic*

conflict–free, xy–admissible, xy–complete, xy–preferred, grounded, acyclic grounded, model, stable$\}$ for $x, y \in \{a, c\}$ be a similar ADF semantics. Let SC^{Tr}_{σ} the identity casting functions for σ. The translation Tr^{AF}_{ADF} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.

Analysis of Translation 23: Under the (pd–acyclic) conflict–free, (xy–) admissible, (xy–) complete, (xy–) preferred, (acyclic) grounded and (stable) model semantics with $x, y \in \{a, c\}$ and the identity casting functions, the translation Tr^{AF}_{ADF} is:

- full, target–subclass and injective
- argument domain preserving and structure preserving
- generic, semantics domain preserving and exact
- $\otimes$–modular and structural

The translation is not $\oplus$–modular. The translation Tr^{AF}_{ADF} is classified as basic under the listed semantics and casting functions.

Explanation. Any Dung's framework can be translated into an ADF and the produced ADF^{AF} class does not account for all the possible frameworks. Thus, the translation is full and target–subclass. There is clearly a one to one relation between the set of attackers of a given argument and the propositional formula (or function) created from them. Since no arguments are added or removed, a given ADF can be produced only by a single AF and the translation is injective. For similar reasons, the approach is also structure preserving – no argument or relation is removed or added, and the nature of the relation stays the same (see Theorem 5.17 and Lemma 4.77). Further structural and semantical properties follow easily from the definition of Translation 23 and Theorem 5.20 and their redefinitions. The translation is also structural – although the attacks need to be transformed into acceptance conditions, no ADF semantics or any knowledge on them is required to do this.

We will now discuss modularity. Let $F_1 = (A_1, R_1)$, $F_2 = (A_2, R_2)$ and $F_3 = (A_1 \cup A_2, R_1 \cup R_2)$. We can observe that conditions of arguments in AF–style ADFs are simply conjunctions of the negations of their parents. Moreover, the parents of a given argument are the same in $Tr^{AF}_{ADF}(A_1) \otimes Tr^{AF}_{ADF}(A_2)$ as in $Tr^{AF}_{ADF}(A_3)$. Consequently, using Definition 3.10 we can show that the conjunction of given two conditions will be the same as the condition produced just from F_3 in the first place. Thus, $\otimes$–modularity is quite clear. On the other hand, $\oplus$–joining can produce a condition that is not even AF–style and thus we cannot speak of $\oplus$–modularity. ■

Example 76. Let us consider the framework Example 1 for the last time. Our AF is $F = (A, R)$ where $A = \{a, b, c, d, e\}$ and $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$. The associated ADF is $D^F = (A, R, C)$, where $C = \{C_a = \top, C_b = \neg a \wedge \neg c, C_c = \neg d, C_d = \neg c, C_e = \neg d \wedge \neg e\}$ (see Figure 50). We can observe that our ADF is strongly valid. Hence, it suffices to focus on acyclic evaluations. The minimal evaluations for our

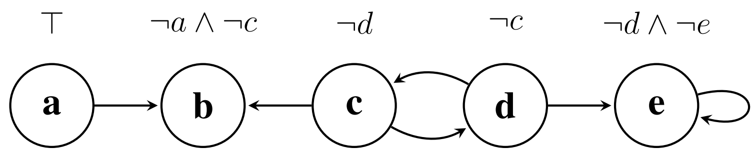


Figure 50: ADF associated with F

arguments a, b, c, d and e are $((a), \emptyset)$, $((b), \{a, c\})$, $((c), \{d\})$, $((d), \{c\})$ and $((e), \{d, e\})$ respectively. The conflict–free (and pd–acyclic conflict–free) extensions of D^F are $E_1 = \emptyset$, $E_2 = \{a\}$, $E_3 = \{b\}$, $E_4 = \{c\}$, $E_5 = \{d\}$, $E_6 = \{a, c\}$, $E_7 = \{a, d\}$ and $E_8 = \{b, d\}$. Their associated discarded sets of any type are $E_1^+ = \emptyset$, $E_2^+ = \{b\}$, $E_3^+ = \emptyset$, $E_4^+ = \{b, d\}$, $E_5^+ = \{c, e\}$, $E_6^+ = \{b, d\}$, $E_7^+ = \{b, c, e\}$ and $E_8^+ = \{c, e\}$. We can observe that only $E_7 = \{a, d\}$ is a model; w.r.t. the extensions E_5 and E_8, the condition C_a is satisfied. For others, C_e evaluates to in. E_7 is also our stable extension. With the exception of E_3 and E_8, all of our sets are admissible extensions of any type of D^F. We can observe that b is not decisively in w.r.t. the ranges of E_3 and E_8 (i.e. $\{a, c\}$ is not a subset of E_3^+ and E_8^+). From there, only E_2, E_6 and E_7 are complete. In the case of the remaining admissible sets E_1, E_4 and E_5, we can see that a is decisively in w.r.t. their ranges, but is not contained in them. Finally, E_2 is the grounded extension of D^F (both acyclic and standard), and E_6 and E_7 are the preferred sets of any type. These answers correspond exactly to the extensions produced by F.

5.8 Summary

In this section we have presented how AFs can be handled by every other argumentation framework we work with in this report – SETAF, AFRA, EAF, EAFC, BAF, AFN, EAS and ADF. We could have observed that all of the translations were quite straightforward and classified as basic. Only in two cases (AFRAs and EASs) we did not obtain exactness. The summary on the properties of our translations is visible in Table 7.

Table 7: Translations from AFs to other frameworks

<table>
<tr><th colspan="2">Properties</th><th>SETAF</th><th colspan="2">AFRA</th><th>EAF</th><th>BAF</th><th>AFN</th><th>EAS</th><th>ADF</th></tr>
<tr><td colspan="2">Translation</td><td>17</td><td colspan="2">18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td></tr>
<tr><td rowspan="6">Strength</td><td>cf</td><td>exact</td><td colspan="2">⊆–weak</td><td>exact</td><td>exact</td><td>exact</td><td>strong</td><td>exact</td></tr>
<tr><td>adm</td><td>exact</td><td>⊆–weak</td><td>strong</td><td>exact</td><td>exact</td><td>exact</td><td>strong</td><td>exact</td></tr>
<tr><td>comp</td><td>exact</td><td colspan="2">faithful</td><td>exact</td><td>exact</td><td>exact</td><td>faithful</td><td>exact</td></tr>
<tr><td>pref</td><td>exact</td><td colspan="2">faithful</td><td>exact</td><td>exact</td><td>exact</td><td>faithful</td><td>exact</td></tr>
<tr><td>grd</td><td>exact</td><td colspan="2">faithful</td><td>exact</td><td>exact</td><td>exact</td><td>faithful</td><td>exact</td></tr>
<tr><td>stb</td><td>exact</td><td colspan="2">faithful</td><td>exact</td><td>exact</td><td>exact</td><td>faithful</td><td>exact</td></tr>
<tr><td colspan="2" rowspan="3">Functional</td><td>full</td><td colspan="2">full</td><td>full</td><td>full</td><td>full</td><td>full</td><td>full</td></tr>
<tr><td>target–subclass</td><td colspan="2">target–subclass</td><td>target–subclass</td><td>target–subclass</td><td>target–subclass</td><td>target–subclass</td><td>target–subclass</td></tr>
<tr><td>injective</td><td colspan="2">injective</td><td>injective</td><td>injective</td><td>injective</td><td>injective</td><td>injective</td></tr>
<tr><td colspan="2" rowspan="3">Syntactical</td><td>argument domain preserving</td><td colspan="2">argument domain preserving</td><td>argument domain preserving</td><td>argument domain preserving</td><td>argument domain preserving</td><td>weakly argument domain altering</td><td>argument domain preserving</td></tr>
<tr><td>structure preserving</td><td colspan="2">structure preserving</td><td>structure preserving</td><td>structure preserving</td><td>structure preserving</td><td>argument introducing</td><td>structure preserving</td></tr>
<tr><td></td><td colspan="2"></td><td></td><td></td><td></td><td>induced support introducing</td><td></td></tr>
<tr><td colspan="2" rowspan="2">Semantical</td><td>generic</td><td colspan="2">generic</td><td>generic</td><td>generic</td><td>generic</td><td>generic</td><td>generic</td></tr>
<tr><td>semantics domain preserving</td><td colspan="2">semantics domain altering</td><td>semantics domain preserving</td><td>semantics domain preserving</td><td>semantics domain preserving</td><td>weakly semantics domain altering</td><td>semantics domain preserving</td></tr>
<tr><td colspan="2" rowspan="2">Computational</td><td>structural</td><td colspan="2">structural</td><td>structural</td><td>structural</td><td>structural</td><td>semi–structural</td><td>structural</td></tr>
<tr><td>modular</td><td colspan="2">modular</td><td>modular</td><td>modular</td><td>modular</td><td>modular</td><td>⊗–modular</td></tr>
</table>

6 Translating SETAFs

In this section we will present five translations from SETAFs to other argumentation frameworks. The translations to EASs and ADFs will not be much different than the approaches from AFs to these structures – they are both capable of expressing group attack easily. However, the translations from SETAFs to AFs and AFNs are more complicated and these are the first coalition and defender approaches we will analyze in this work. In fact, as we will see, it is not possible to create easy, exact translations for these frameworks. We thus obtain our first structure clearly going beyond Dung's in terms of expressiveness. Our running example in this section will be Example 3 from Section 2.1.2. We restate it here for the readers' convenience:

Example 3. Let us consider the SETAF $SF = (A, R)$ with $A = \{a, b, c, d, e\}$ and the attack relation $R = \{(\{a\}, c), (\{a\}, b), (\{b\}, a), (\{c\}, d), (\{e\}, a), (\{b, d\}, e)\}$, as depicted in Figure 2. The admissible extensions of this framework are $\emptyset$, $\{b\}$, $\{b, c\}$, $\{c, e\}$ and $\{b, c, e\}$. Only $\emptyset$ and $\{b, c, e\}$ are complete. The grounded extension is $\emptyset$, while $\{b, c, e\}$ is both preferred and stable.

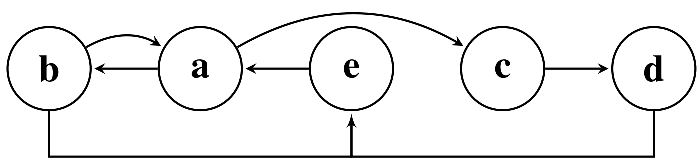


Figure 51: Sample SETAF

6.1 SETAF as AF

In this section we will discuss two translations from SETAFs to AFs – one inspired by the coalition, the other by the defender approach. We will then propose some improvements to the first translation and show that no full exact SETAF–AF transformation can exist.

6.1.1 Basic Coalition Translation

One way of translating a SETAF to an AF is by combining the already existing SETAF–EAS and EAS–AF transformations from Sections 6.4 and 11.1. However, this direction introduces the support from evidence, which is not that intuitive when shifting between purely attack–based frameworks. Consequently, the approach can be somewhat simplified.

What we will consider first is a SETAF–AF translation inspired by the coalition approach. The Dung's framework handles only the binary, not the group attack. Thus, our approach is to "hide" the advanced conflicts within the arguments. The AF arguments will now correspond to sets of SETAF arguments, including both the single–element sets to represent the original arguments and multi–element sets for the collections of arguments that carry out attacks.

What needs to be handled now is the propagation of conflicts. Naturally, the attacks carried out against a given argument in SETAF need to be propagated to the group conflict–arguments containing this argument in the corresponding AF. Moreover, we can also say that a given (AF) argument carries out an attack if any of its (SETAF) argument subsets does. This brings us to the following translation:

Translation 24. ***Deprecated*** *Let $SF = (A, R)$ be a SETAF. Its corresponding AF $F^{SF} = (A', R')$ is built the following way:*

- *let $arg(S) = \{\{a\} \mid a \in S\}$, where $S \subseteq A$, be a function returning a collection of single element sets composed of elements of S,*
- *let $att(S) = \{S' \mid S' \subseteq S \wedge |S'| > 1 \wedge \exists y \in A\, s.t.\, (S', y) \in R\}$, where $S \subseteq A$, be a function returning subsets of S of size bigger than 1, which attack some argument in A,*
- *$A' = arg(A) \cup att(A)$, and*
- *$R' = \{(X, Y) \mid \exists y \in Y, X' \subseteq X s.t.(X', y) \in R\}$.*

Please note that we decide to take the size of the attacking set into account when creating $att(A)$ so that it is disjoint from $arg(A)$. Although dropping this restriction would not change the resulting framework, it does make a difference when we consider certain improvements (see Section 6.1.3).

We can observe that the translation does not behave well when it comes to conflict–freeness, i.e. a conflict–free extension of F^{SF} might not be conflict–free in SF. This is due to the fact that that the attack arguments containing arguments in a conflict–free set of F^{SF} might not necessarily be in the set:

Example 77. Consider a SETAF $(\{a, b, c\}, \{(\{a, b\}, c)\})$ with a single group attack. The set $\{a, b, c\}$ is not conflict–free in this case. The corresponding AF is now $(\{\{a\}, \{b\}, \{c\}, \{a, b\}\}, \{(\{a, b\}, c)\})$. Even though the set $\{\{a\}, \{b\}, \{a, b\}, \{c\}\}$ is not conflict–free, $\{\{a\}, \{b\}, \{c\}\}$ is.

Since conflict–freeness is not preserved anyway, it means that not all of the conflicts we have created in AFs are really necessary – they can reduce the number of undesirable extensions produced by F^{SF}, but do not "fix" the issue. Consequently, we can get rid of the subset–propagation step and simplify the attack relation a bit. The fact that introducing additional attacks from the "bigger" sets in the deprecated Translation 24 is not vital follows from the properties of the minimal normal form for SETAFs. Even though the amount of attacks an argument receives can change, due to a subset relation between them the defense itself is not affected. This brings us to a slightly simpler formulation:

Translation 25. *Let $SF = (A, R)$ be a SETAF. Its corresponding AF $F^{SF} = (A', R')$ is built the following way:*

- *let $arg(S) = \{\{a\} \mid a \in S\}$, where $S \subseteq A$, be a function returning a collection of single element sets composed of elements of S,*

- *let* $att(S) = \{S' \mid S' \subseteq S \wedge |S'| > 1 \wedge \exists y \in A\, s.t.\, (S', y) \in R\}$*, where* $S \subseteq A$*, be a function returning subsets of* S *of size bigger than 1, which attack some argument in* A*,*

- $A' = arg(A) \cup att(A)$*, and*

- $R' = \{(X, Y) \mid \exists y \in Y s.t.(X, y) \in R\}$.

Nevertheless, we can observe that the translation still produces a large number of additional arguments. The amount of attacks we can have in a SETAF is bounded by $n \times |A|$, i.e. every non–empty subset of arguments can attack every argument. Even if we remove the single–element attacks, we are left with a possible exponential blow up. This explicitly shows that we are leaving the domain of purely basic translations.

Theorem 6.1. *Let* $SF = (A, R)$ *be a SETAF and* F^{SF} *its corresponding AF obtained by Translation 25. If* $E \subseteq A$ *is a* σ*–extension of* SF*, where* $\sigma \in$ *{conflict–free, admissible, preferred, complete, grounded, stable}, then* $arg(E) \cup att(E)$ *is a* σ*–extension of* F^{SF}*. If a set of arguments* $E' \subseteq A'$ *is a* σ'*–extension of* F^{SF}*, where* $\sigma' \in$ *{admissible, preferred, complete, grounded, stable}, then* $\bigcup E'$ *is a* σ*–extension of* SF*.*

The redefinitions of our translations and theorems are now as follows. Please note that in this case, we will not provide a description of the subclass of produced AFs:
Redefinition of Translation 25: *Let* Fr^{SETAF} *be the collection of all SETAFs based on domain* $\mathcal{U}$ *and* Fr^{AF} *the collection of all AFs based on the domain* $2^{\mathcal{U}}$*. The translation* $Tr_{AF}^{SETAF} : Fr^{SETAF} \rightarrow Fr^{AF}$ *is defined as* $Tr_{AF}^{SETAF}((A, R)) = (A', R')$*, where* $A' = \{\{a\} \mid a \in A\} \cup \{S' \mid S' \subseteq A \wedge |S'| > 1 \wedge \exists y \in A\, s.t.\, (S', y) \in R\}$ *and* $R' = \{(X, Y) \mid \exists y \in Y\, s.t.(X, y) \in R\}$ *for a framework* $(A, R) \in Fr^{SETAF}$*.*
Redefinition of Theorem 6.1: *Let* $\sigma \in$ *{admissible, preferred, complete, grounded, stable} be a semantics and* SC_{σ}^{Tr} *the union casting functions for* σ*. The translation* Tr_{AF}^{SETAF} *is strong under* $(\sigma, SC_{\sigma}^{Tr})$ *and* $\subseteq$*–weak under the conflict–free semantics and union casting functions. It is semantics bijective under the complete, preferred, grounded and stable semantics and union casting functions.*
Analysis of Translation 25: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and union casting functions, the translation Tr_{AF}^{SETAF} is:

- full, target–subclass and injective

- argument domain altering, argument introducing, induced attack relation introducing

- generic and semantics domain altering

- structural

The translation Tr_{AF}^{SETAF} is not modular. We classify Tr_{AF}^{SETAF} as basic–coalition hybrid under the listed semantics and casting functions.

Explanation. Any SETAF can be transformed into an AF and thus our approach is classified as full. Structurally speaking, any type of an AF can be produced by our translation. We can observe that if we translated an AF into a SETAF and then "came back", the same framework would have been obtained. However, we no longer use $\mathcal{U}$ as the target domain, but $2^{\mathcal{U}}$. This means that if we take into account the nature of the arguments in AFs, we can construct frameworks that cannot be produced by our translation. For example an AF $(\{\{a,b,c\}\}, \emptyset)$ consisting of a single (attack style) argument $\{a,b,c\}$ and no conflict whatsoever is not a structure the translation Tr_{AF}^{SETAF} can produce. Therefore, the presented approach is target–subclass.

We can observe that the translation is injective. Two different frameworks producing the same AF would have to have the same set of arguments, otherwise the argument set in the target AF would not agree. Moreover, we can observe that for a given SETAF attack (X,y), $(X,\{y\})$ is presented in the AF, and there are no attacks in R' carried out against single–element sets that would not originate from the source SETAF. This means that the two SETAFs would have to have the same attacks as well. Since they also had to have the same arguments, our two "different" frameworks could not have been different after all. Thus, Tr_{AF}^{SETAF} is injective.

The translation is clearly both argument and semantics domain altering. Moreover, we also introduce new arguments that would take over the group attack. However, we can observe that every group attack from the source SETAF is in fact represented in the target AF – this means we do not remove any information. Nevertheless, we have additional conflicts in the target AF, which although derivable from the existing ones, are used to account for the shift in the group attack representation. Since they occur between the added arguments, they can be classified as induced. The fact that we can consider those attacks really as additional can be more seen in the explanation concerning the lack of modularity in the translation. The amount of handled semantics (in a strong manner) makes the translation generic. The approach is also clearly a structural one.

The reason why our translation is not modular is the creation of additional attacks between att arguments. Let us consider two SETAFs $SF_1 = (\{a,b,c,d\}, \{(\{a,b\},c)\})$ and $SF_2 = (\{a,b,c,d\}, \{(\{c,d\},a)\})$. The translation produces $F^{SF_1} = (\{\{a\},\{b\},\{c\},\{d\},\{a,b\}\}, \{(\{a,b\},\{c\})\})$ and $F^{SF_2} = (\{\{a\},\{b\},\{c\},\{d\},\{c,d\}\}, \{(\{c,d\},\{a\})\})$. The AF associated with $SF_1 \cup SF_2$ is $(\{\{a\}, \{b\},\{c\}, \{d\},\{a,b\}, \{c,d\}\}, \{(\{a,b\},\{c\}), (\{c,d\},\{a\}), (\{a,b\},\{c,d\}),(\{c,d\},\{a,b\})\})$, which is not the same as $F^{SF_1} \cup F^{SF_2}$ – in this case, the $(\{a,b\},\{c,d\})$ and $(\{c,d\},\{a,b\})$ attacks are not present. This difference is not just a "cosmetic" change – without the additional conflicts, there would be no set defending c against the $\{a,b\}$ attack and no set defending a against the $\{c,d\}$ attack. As a result, the admissible extensions would not be retrieved.

The target argument domain is a powerset of the source one. Moreover, we use the additional arguments to take over the group attack not handled by the Dung's framework. This classifies Tr_{AF}^{SETAF} as a coalition translation. However, the simplicity of the translation w.r.t. typical coalition approaches brings it closer to basic translations. Therefore, we decide to classify Tr^{SETAF}_{AF} as a hybrid. ■

What is worth mentioning is that the deprecated version of Translation 25 does not have exactly the same properties as the new version, even though in both cases the Theorem 6.1 holds (the version for the other translation just has a slightly modified proof). The deprecated approach is overlapping. The addition of subset attacks means that the frameworks in which they did and did not occur in the first place are translated into the same AF. For example, the SETAFs $(\{a,b,c,d\}, \{(\{a\},b), (\{a,c\},d)\})$ and $(\{a,b,c,d\}, \{(\{a\},b), (\{a,c\},b), (\{a,c\},d)\})$ would both be transformed into the AF $(\{\{a\},\{b\},\{c\},\{d\},\{a,c\}\}, \{(\{a\},\{b\}), (\{a,c\},\{b\}), (\{a,c\},\{d\})\})$.

Example 78. Let us consider the SETAF $SF = (A, R)$ with $A = \{a, b, c, d, e\}$ and the attack relation $R = \{(\{a\}, c)$, $(\{a\}, b)$, $(\{b\}, a)$, $(\{c\}, d)$, $(\{e\}, a)$, $(\{b, d\}, e)\}$, previously analyzed in Example 3. The associated AF created using Translation 25 is $F^{SF} = (A', R')$, where $A' = \{\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$, $\{b, d\}\}$ and $R' = \{(\{a\}, \{c\})$, $(\{a\}, \{b\})$, $(\{b\}, \{a\})$, $(\{c\}, \{d\})$, $(\{e\}, \{a\})$, $(\{b, d\}, \{e\})$, $(\{c\}, \{b, d\})$, $(\{a\}, \{b, d\})\}$. We can observe that the set $\{\{b\}, \{d\}, \{e\}\}$ is conflict–free in F^{SF}, even though $\{b, d, e\}$ is not conflict–free in F. The admissible extensions of our framework are $\emptyset$, $\{\{b\}\}$, $\{\{b\}, \{c\}\}$, $\{\{c\}, \{e\}\}$ and $\{\{b\}, \{c\}, \{e\}\}$. They correspond to the sets $\emptyset$, $\{b\}$, $\{b, c\}$, $\{c, e\}$ and $\{b, c, e\}$, which were our original extensions. The complete extensions are $\emptyset$ and $\{\{b\}, \{c\}, \{e\}\}$, with the former being grounded and the latter preferred and stable. This again is in agreement with the sets produced by SF.

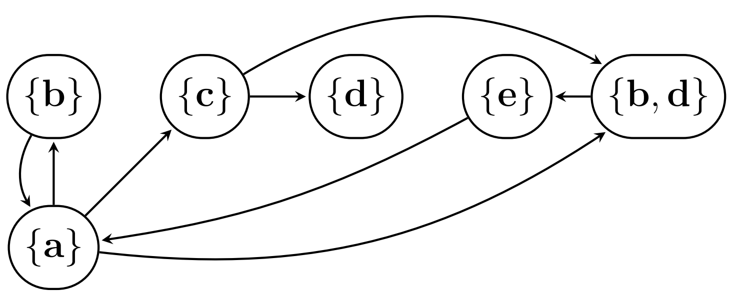
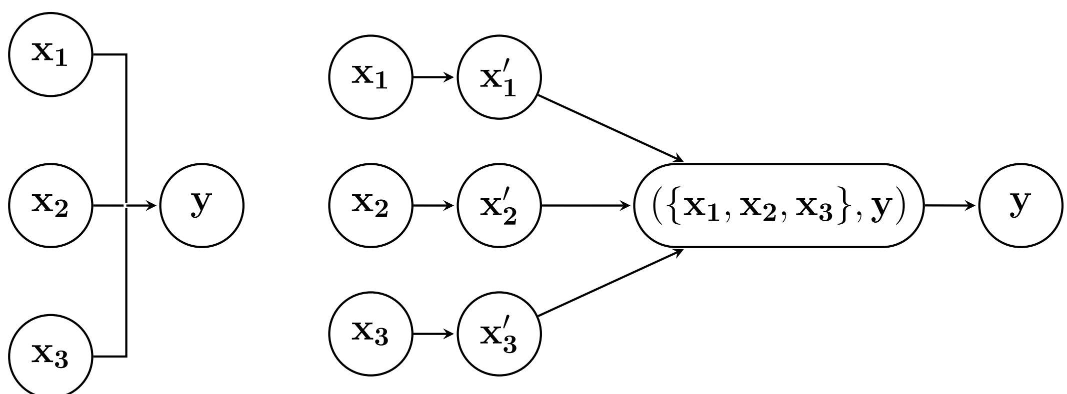


Figure 52: Coalition AF for a SETAF

6.1.2 Basic Defender Translation

In [64], a meta–level argumentation framework has been introduced. Among others, the work also provided translations from SETAFs to this structure. The target arguments are now statements about the elements of the source framework, such as whether an argument is justified, rejected, or defeats another argument. As an intermediary step, the framework undergoes an "expansion". The chosen attacks, independently of whether they are binary or group ones, now become arguments that carry out a conflict against the previous target. Furthermore, with the use of auxiliary arguments, they need to be defended by the arguments that were the source of the attack (see Figure 53). Only then the arguments are transformed into meta–level statements – an argument x becomes x *is justified*, x' is made into x *is rejected*, and $(\{x\}, y)$ is assigned $\{x\}$ *defeats* y.

In what follows we will present a translation corresponding to the expansion step of the meta–level approach with a minor modification. Although the logic–based meta–level AFs introduced in [64] are powerful tools, our focus is on explaining and understanding the way the translation works. Moreover, in this way we avoid a more intrusive domain change and thus can create a faithful translation. We therefore refer the reader to the original paper for further details and will use a (structurally speaking) simplified version of the framework. We will also focus on the case where all of the attacks are substituted in the expansion step and adapt the definitions and theorems from [64] accordingly [18]. Finally, while the primed arguments are introduced only for those arguments that participate in attacks during the expansion step, the meta–level framework contains the rejected version for every argument. Although this difference does not affect the strength of the translation in any way, we choose not to make a distinction between attackers and non–attackers and follow the meta approach.



(a) Sample SETAF

(b) SETAF expansion

Figure 53: Group attack expansion

We can observe that the construction we have just described is in fact what we have classified as the defender approach (see Section 3.3). Moreover, it bears a striking resemblance to the results of a chaining the SETAF–AFN and the defender AFN–AF (strongly valid version) translations (see Translations 29 and 66). Please note that in this approach, both group and binary attacks undergo a transformation, not just the group ones, like in Translation 25. Moreover, while previously we only stored the attacking set, here the whole conflict (i.e. both source and target) becomes the new argument.

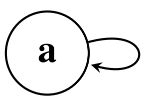
Translation 26. *Let $SF = (A, R)$ be a SETAF. Its corresponding defender AF is $F^{SF}_{def} = (A', R')$, where:*

[18] Please note that although expanding a SETAF in [64] still, technically speaking, gives us a SETAF, all the attacks are in fact binary and thus adapting it to an AF form is trivial.

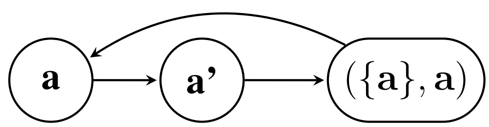
- $A' = A \cup R \cup X'$, *where* $X' = \{x' \mid x \in A\}$, *and*
- $R' = \{(x, x') \mid x' \in X'\} \cup \{(x', (X, y)) \mid x \in X, (X, y) \in R\} \cup \{((X, y), y) \mid (X, y) \in R\}$.

We can observe that just like in Translation 25, the conflict–free semantics is not preserved. However, while in the previous case conflict–freeness still held if the translated SETAF was actually a Dung's framework, in this approach it is lost completely:

Example 79. Let $(\{a\}, \{(\{a\}, a)\})$ be the AF–style SETAF consisting of a single self–attacking argument that is depicted in Figure 54a. The corresponding defender AF is now $(\{a, a', (\{a\}, a)\}, \{(a, a'), (a', (\{a\}, a)), ((\{a\}, a), a)\})$ (see Figure 54b). We can observe that while $\{a\}$ is not conflict–free in the source framework, it is in the target one. However, it is not admissible in any of the structures.



(a) SETAF with self–attacker



(b) SETAF expansion

Figure 54: Self–attack expansion

Let us now focus on analyzing the semantics. As expected, the translation gains strength starting from the admissible semantics, not the conflict–free. Please note that in the construction of the target extensions we will use the concept of the discarded set (see Definition 2.25).

Theorem 6.2. *Let* $SF = (A, R)$ *be a SETAF,* $F^{SF}_{def} = (A', R')$ *its corresponding defender AF obtained through Translation 26 and* $\sigma \in \{$*admissible, preferred, complete, grounded, stable*$\}$ *a semantics. If* $E \subseteq A$ *is conflict–free in* SF*, then* $E \cup \{(X, y) \mid X \subseteq E\} \cup \{x' \mid x' \in X', x \in (A \setminus E)\}$ *is conflict–free in* F^{SF}_{def}*. If* $E \subseteq A$ *is a* σ*–extension of* SF*, then* $E \cup \{(X, y) \mid X \subseteq E\} \cup \{x' \mid x' \in X', x \in E^{+}_{SF}\}$ *is a* σ*–extension of* F^{SF}_{def}*. If* $E' \subseteq A'$ *is a* σ*–extension of* F^{SF}_{def}*, then* $E' \cap A$ *is a* σ*–extension of* SF*.*

We can now put the expansion translation into our classification system:

Redefinition of Translation 26: *Let* Fr^{SETAF} *be the collection of all SETAFs on domain* $\mathcal{U}$ *and* Fr^{AF} *the collection of all AFs based on domain* $\mathcal{U} \cup \mathcal{U}' \cup ((2^{\mathcal{U}} \setminus \emptyset) \times \mathcal{U})$*. The translation* $def\text{-}Tr^{SETAF}_{AF} : Fr^{SETAF} \rightarrow Fr^{AF}$ *is defined as* $def\text{-}Tr^{SETAF}_{AF}((A, R)) = (A', R')$*, where* $A' = A \cup R \cup X'$*, where* $X' = \{x' \mid \exists X \subseteq A, y \in A\, s.t.\, x \in X$ *and* $(X, y) \in R\}$ *and* $R' = \{(x, x') \mid x' \in X'\} \cup \{(x', (X, y)) \mid x \in X, (X, y) \in R\} \cup \{((X, y), y) \mid (X, y) \in R\}$.

Redefinition of Theorem 6.2: *Let* $\sigma \in \{$*admissible, preferred, complete, grounded, stable*$\}$ *be a semantics and* SC^{Tr}_{σ} *the removal casting functions for* σ *defined as*

$SC_{\sigma}^{X}(E) = E \cap A$, where $X \in Fr^{SETAF}$ is a framework with set of arguments A and $E \in \sigma(def\text{-}Tr_{AF}^{SETAF}(X))$. The translation $def\text{-}Tr_{AF}^{SETAF}$ is strong under $(\sigma, SC_{\sigma}^{Tr})$ and $\subseteq$–weak under conflict–free semantics and the removal casting functions. It is semantics bijective under the complete, preferred, grounded and stable semantics and the defined removal casting functions.

Analysis of Translation 26: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and removal casting functions, the translation $def\text{-}Tr_{AF}^{SETAF}$ is:

- full, target–subclass and injective
- argument domain altering, argument introducing and induced attack introducing
- generic, semantics domain altering
- semi–structural and modular

Under the complete, preferred, grounded and stable semantics and removal casting functions, $def\text{-}Tr_{AF}^{SETAF}$ is faithful. Translation $def\text{-}Tr_{AF}^{SETAF}$ is classified as basic–defender hybrid under the listed semantics and casting functions.

Explanation. Any SETAF can be translated into an AF and thus the translation is full. However, not every AF constructed with the given argument domain can be produced by a SETAF. Let us consider a very simple attack $(\{a\}, b)$. It becomes expanded into (a, a'), $(a', (a, b))$ and $((a, b), b)$. Consequently, a sequence of attacks of length 1 is transformed into a sequence of length 3. It is not possible to obtain an AF with a sequence of length 2. This is only one of the examples of Dung's frameworks that we cannot obtain through the defender translation. Consequently, $def\text{-}Tr_{AF}^{SETAF}$ is target–subclass, but for now we cannot describe the subclass fully. Let us now analyze whether the translation is injective. Let us assume it is not and that we have two different SETAFs $SF_1 = (A_1, R_1)$ and $SF_2 = (A_2, R_2)$ that produce the same AF $F = (A', R')$. A' is obtained by joining arguments, attacks, and primed versions of arguments that took part in carrying out the attacks. Consequently, it has to be the case that $A_1 \cup R_1 \cup X_1' = A_2 \cup R_2 \cup X_2'$. Since R_1 and R_2 are defined over a domain different than A and X sets, then $R_1 = R_2$ and thus $A_1 \cup X_1' = A_2 \cup X_2'$. Since X_1' and X_2' are induced by the attack relations, then $R_1 = R_2$ implies $X_1' = X_2'$. Since due to the domain change we assumed that the set of arguments is disjoint from the set of primed arguments, $A_1 = A_2$. Thus, our two SETAFs are in fact not different and the translation is injective.

The fact that the translation is domain altering and argument introducing can be observed from the definition itself. Again, we do not deal with relation removal; the attack arguments and the conflict they carry out at the target represent the relation in the source SETAF. However, we also include the primed versions of arguments, which as such are auxiliary and require additional conflicts. Thus, we say that the translation introduces induced attacks.

Our approach is clearly generic and semantics domain altering. We also exploit defense in order to tie a conflict argument to the conflict sources; consequently, we qualify our approach as semi–structural. We can observe our translation is modular; every attack is expanded separately and the union of two SETAFs will not produce a conflict that was not in one of the frameworks before.

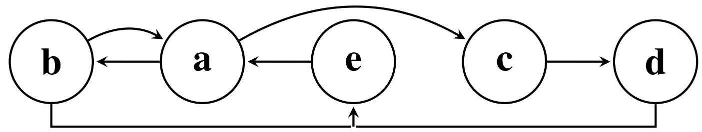
From the redefinition of Theorem 6.2 we can see that our approach is semantics bijective, and since we are dealing with removal casting functions, $def\text{-}Tr_{AF}^{SETAF}$ is faithful for complete, preferred, grounded and stable semantics. Please note that the translation is not bijective under the admissible semantics. We can look at the framework $(\{x_1, x_2, x_3, x_1', x_2', x_3', (\{x_1, x_2, x_3\}, y), y\}, \{(x_1, x_1'), (x_2, x_2'), (x_3, x_3'), (x_1', (\{x_1, x_2, x_3\}, y)), (x_2', (\{x_1, x_2, x_3\}, y)), (x_3', (\{x_1, x_2, x_3\}, y)), ((\{x_1, x_2, x_3\}, y), y)\})$ from Figure 53. In this case, both $\{x_1, x_2, x_3\}$ and $\{x_1, x_2, x_3, (\{x_1, x_2, x_3\}, y)\}$ are admissible extensions of the target AF, and at the same time they correspond to the set $\{x_1, x_2, x_3\}$ in the source SETAF.

We have already discussed that the translation follows the defender pattern. However, since it is a relatively simple one, we decide to classify it as a basic–defender hybrid. ■

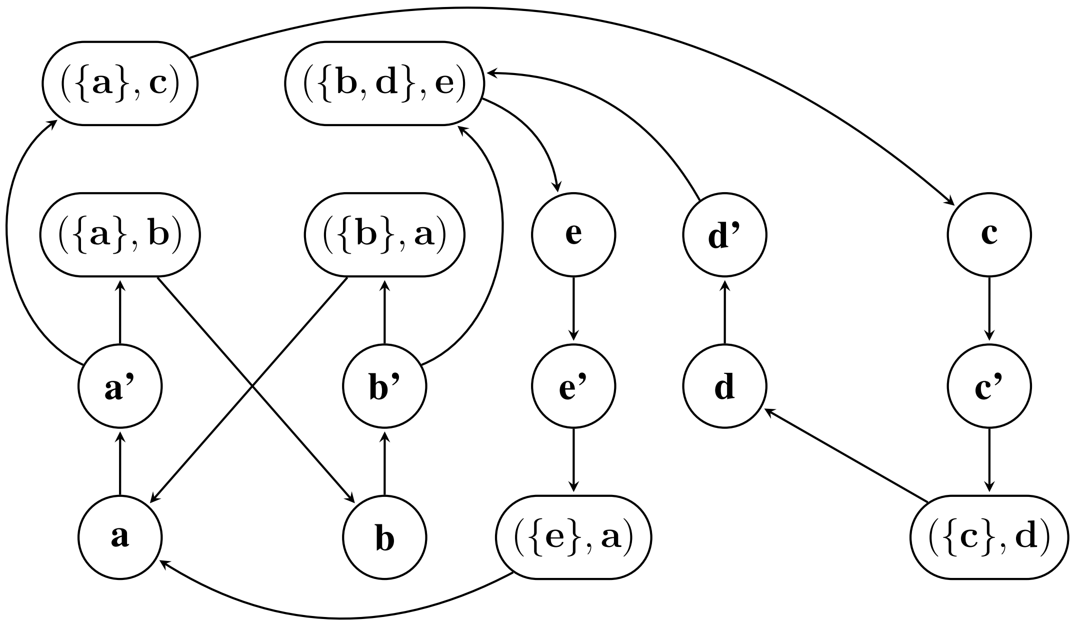
Example 80. Let us come back to the $SF = (A, R)$ with the set of arguments $A = \{a, b, c, d, e\}$ and the attack relation $R = \{(\{a\}, c), (\{a\}, b), (\{b\}, a), (\{c\}, d), (\{e\}, a), (\{b, d\}, e)\}$, previously analyzed in Example 3 and visible in Figure 55a. The associated AF created using Translation 26 is $F^{SF} = (A', R')$, where $A' = \{a, b, c, d, e, a', b', c', d', e', (\{a\}, c), (\{a\}, b), (\{b\}, a), (\{c\}, d), (\{e\}, a), (\{b, d\}, e)\}$ and $R' = \{(a, a'), (b, b'), (c, c'), (d, d'), (e, e'), (a', (\{a\}, c)), (a', (\{a\}, b)), (b', (\{b\}, a)), (c', (\{c\}, d)), (e', (\{e\}, a)), (b', (\{b, d\}, e)), (d', (\{b, d\}, e)), ((\{a\}, c), c), ((\{a\}, b), b), ((\{b\}, a), a), ((\{c\}, d), d), ((\{e\}, a), a), ((\{b, d\}, e), e)\}$. We can see it depicted in Figure 55b. The admissible extensions of F^{SF} are $\emptyset$, $\{b, a', (\{b\}, a)\}$, $\{b, c, a', (\{b\}, a)\}$, $\{b, c, a', (\{b\}, a), (\{c\}, d)\}$, $\{b, c, a', d', (\{b\}, a), (\{c\}, d)\}$, $\{c, e, a', d', (\{c\}, d), (\{e\}, a)\}$, $\{b, c, e, a', d', (\{b\}, a), (\{c\}, d)\}$ and $\{b, c, e, a', d', (\{b\}, a), (\{c\}, d), (\{e\}, a)\}$. They correspond to the sets $\emptyset$, $\{b\}$, $\{b, c\}$, $\{c, e\}$ and $\{b, c, e\}$, which are the admissible extensions of SF. We can observe that the sets $\{b, c\}$ and $\{b, c, e\}$ can be produced by more than one extension of F^{SF}. Out of all of these sets, $\emptyset$ and $\{b, c, e, a', d', (\{b\}, a), (\{c\}, d), (\{e\}, a)\}$ are complete, which is the desired answer. The first set is also grounded, while the other one is preferred and stable.

6.1.3 Improvements

In this section we have presented two approaches towards translating SETAFs into AFs. We have observed that the first translation (Translation 25) was strong and semantics bijective, while the other (Translation 26) faithful. Often it holds that strong and bijective transformations can be upgraded to faithful ones, and the first approach is not an exception. Consequently, we will explain how it can be enhanced and what are the side effects of the improvement. However, most importantly, we will show that for certain semantics



(a) Sample SETAF SF



(b) Meta–level AF for SF

Figure 55: A SETAF and its meta–level AF

it is not possible to create an exact SETAF-AF translation at all. Thus, the results we have presented can, in a certain sense, be the best we can hope for.

In the proof of Theorem 6.1 we could have observed that the complete extensions of AFs obtained through translation Tr_{AF}^{SETAF} always include the sources of the attackers. Consequently, the information carried by the att arguments is in fact redundant under complete, and thus preferred, grounded and stable semantics. Therefore, we can simply consider removing those arguments from an extension, and adapt our approach to work with removal, not coalition casting functions. Unfortunately, this means that the admissible extensions are no longer preserved.

Translation 27. *Let $SF = (A, R)$ be a SETAF. Its corresponding AF $F^{SF} = (A', R')$ is built the following way:*

- *let $att(S) = \{S' \mid S' \subseteq S \wedge |S'| > 1 \wedge \exists y \in A \, s.t. \, (S', y) \in R\}$, where $S \subseteq A$, be a function returning subsets of S of size bigger than 1, which attack some argument* *in* A*,*

- $A' = A \cup att(A)$*, and*
- $R' = \{(X, Y) \mid \exists y \in Y s.t.(X, y) \in R$ *or* $(\{X\}, y) \in R\}$.

Theorem 6.3. *Let* $SF = (A, R)$ *be a SETAF and* $F^{SF} = (A', R')$ *its corresponding AF obtained by Translation 27. If* $E \subseteq A$ *is a* σ*–extension of* SF*, where* $\sigma \in \{conflict-$ $-free, admissible, complete, preferred, grounded, stable\}$ *then* $E \cup att(E)$ *is a* σ*–extension of* F^{SF}*. If a set of arguments* $E' \subseteq A'$ *is a* σ*–extension of* F^{SF}*, where* $\sigma \in \{complete, preferred, grounded, stable\}$*, then* $E' \cap A$ *is a* σ*–extension of* SF*.*

The proof of this theorem is a simple adaptation of the original one from Theorem 6.1. When we redefine this translation into our system, we obtain the following:
Redefinition of Translation 27: *Let* Fr^{SETAF} *be the collection of all SETAFs based on domain* $\mathcal{U}$ *and* Fr^{AF} *the collection of all AFs based on the domain* $\mathcal{U} \cup 2^{\mathcal{U}}$*. The translation* $imp\text{-}Tr_{AF}^{SETAF} : Fr^{SETAF} \rightarrow Fr^{AF}$ *is defined as* $imp\text{-}Tr_{AF}^{SETAF}((A, R)) = (A', R')$*, where* $A' = A \cup \{S' \mid S' \subseteq A \wedge |S'| \geq 1 \wedge \exists y \in A\, s.t.\, (S', y) \in R\}$ *and* $R' = \{(X, Y) \mid \exists y \in Y s.t.(X, y) \in R$ *or* $(\{X\}, y) \in R\}$ *for a framework* $(A, R) \in Fr^{SETAF}$*.*
Redefinition of Theorem 6.3: *Let* $\sigma \in \{$*complete, preferred, grounded, stable*$\}$ *be a semantics and* SC_σ^{Tr} *the removal casting functions for* σ *defined as* $SC_\sigma^X(E) = E \cap A$*, where* $X \in Fr^{SETAF}$ *is a framework with set of arguments* A *and* $E \in \sigma(imp\text{-}Tr_{AF}^{SETAF}(X))$*. The translation* $imp\text{-}Tr_{AF}^{SETAF}$ *is strong and semantics–bijective under* (σ, SC_σ^{Tr})*. It is* $\subseteq$*–weak under the conflict–free and admissible semantics and removal casting functions.*

By combining the existing properties of the original form of this translation and adding the removal casting functions, we can conclude that $imp\text{-}Tr_{AF}^{SETAF}$ is faithful under complete, preferred, grounded and stable semantics and the defined casting functions. However, we decide not to change the classification of this translation from basic–coalition to e.g. basic. The group arguments still represent the attacks, even though we remove them from extensions. We could decide to use the auxiliary arguments of the same type as the existing ones and thus go with a basic and weakly domain altering translation. However, we then lose the conflict information and thus our approach would be relation removing and lossy. Consequently, the choice is left to the reader, as depending on the application of the translation, one property can be more desirable than another.

Although for the time being we are not sure whether a faithful translation for admissible semantics can be created, we know that no exact one can exist. Let us first describe the SETAF we are going to use [37].

Example 81. Let $SF = (\{x, y, z\}, \{(\{x, y\}, z), (\{y, z\}, x), (\{x, z\}, y)\})$ be a SETAF depicted in Figure 56. The conflict–free extensions of SF are $\emptyset$, $\{x\}$, $\{y\}$, $\{z\}$, $\{x, y\}$, $\{x, z\}$ and $\{y, z\}$, out of which $\emptyset$, $\{x, y\}$, $\{x, z\}$ and $\{y, z\}$ are admissible. They are also complete. Furthermore, the last three are preferred and stable. $\emptyset$ is the grounded extension.

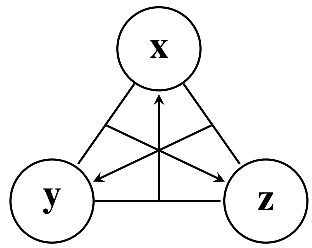


Figure 56: Framework SF

We can now start with the analysis of the sets of extensions (Definitions 2.175, 2.176 and 2.177). Let us focus on the collection $S_1 = \{\emptyset, \{x\}, \{y\}, \{z\}, \{x, y\}, \{x, z\}, \{y, z\}\}$. The arguments in S_1 are $Arg_{S_1} = \{x, y, z\}$ and the pair collection is $Pair_{S_1} = \{(x, x), (y, y), (z, z), (x, y), (x, z), (y, z), (y, x), (z, x), (z, y)\}$. We can observe that S_1 is downward–closed. However, it is not tight; for example, $\{x, y\} \cup \{z\} \notin S_1$, but both $\{z, x\}$ and $\{z, y\}$ are in $Pair_{S_1}$. Therefore, S_1 does not meet the signature requirements for conflict–freeness in AFs.

We can now focus on the collection $S_2 = \{\emptyset, \{x, y\}, \{x, z\}, \{y, z\}\}$. The set of arguments of S_2 and the $Pair_{S_2}$ collection are the same as in the S_1 case. We can observe that S_2 is not adm–closed. Consider the sets $\{x, y\}$ and $\{x, z\}$; for every $a, b \in \{x, y, z\}$, $(a, b) \in Pair_{S_2}$. However, $\{x, y, z\} \notin S_2$. This means that our collection does not fit the admissible signature in AFs. Moreover, we can observe that S_2 is not com–closed. We can consider the subset $\{\{x, y\}, \{y, z\}\}$ of S_2. Every pair made out of arguments in $\{x, y, z\}$ is in $Pair_{S_2}$. However, the completion–sets collection C_s of $\{x, y, z\}$ is empty, and thus the com–closed requirements are not met. Although the complete signature is not yet fully analyzed, being com–closed is one of the necessary properties for a set of extensions.

We can now consider $S_3 = \{\emptyset, \{x, y\}, \{x, z\}, \{y, z\}\}$. The same analysis as in the S_2 case holds; the collection is still not adm–closed. Thus, even though it is incomparable, it does not fit the preferred semantics signature. We can reiterate the explanation for S_1 to show that S_3 is not tight. Consequently, the stable signature requirements are not satisfied.

In summary, we can observe that the extensions of our framework SF from Example 81 fail every AF signature we have analyzed. This means that there are no chances for a strong and full translation that would use the identity casting functions. Consequently, no exact and full translation from SETAFs to AFs can be constructed.

Theorem 6.4. *Let Fr^{SETAF} be the collection of all SETAFs on a domain $\mathcal{U}^{SETAF}$ and Fr^{AF} the collection of all AFs on a domain $\mathcal{U}^{AF}$. There exists no full translation from Fr^{SETAF} to Fr^{AF} that is exact under conflict–free, admissible, complete, preferred and stable semantics and identity casting functions for them.*

Please note that an exact translation for the grounded semantics will exist. Since only a single extension is produced, every argument not present in it can be removed from the framework. We thus obtain a trivial, though expensive and extremely semantical translation.

6.2 SETAF as BAF

We can now try to compare SETAFs to BAFs. In the coalition–style SETAF–AF Translation 25 we could have observed that the arguments representing attacks were somehow detached from the arguments that originally carried them out. We also had to propagate conflicts from the attack arguments to attack arguments. In Translation 26, this was addressed by connecting the relevant framework elements through defense. However, another way to approach it is to connect attack arguments and standard arguments through support:

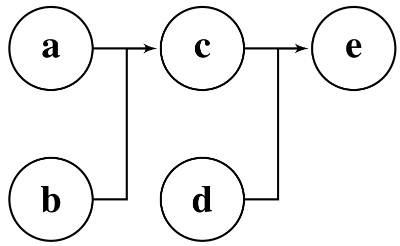
Example 82. Let us consider a simple SETAF depicted in Figure 57a. In order to transform it to a BAF, we can add the attack arguments to the framework and add support to them from the arguments that carry them out, as visible in Figure 57b.

The fact that in order to accept e.g. $\{a, b\}$, we need to accept both a and b, can be read as necessary support from a and b to $\{a, b\}$. We can now choose to include the secondary attack in our analysis, which in this case would permit $\{a, b\}$ to attack $\{c, d\}$, which is a desirable reading. Additionally, we would demand inverse closure of the BAF extensions in order to make sure that the supporters of a given argument are in fact present in an extension. However, what would also be very useful is including a certain "group" version of supported attack and closure.

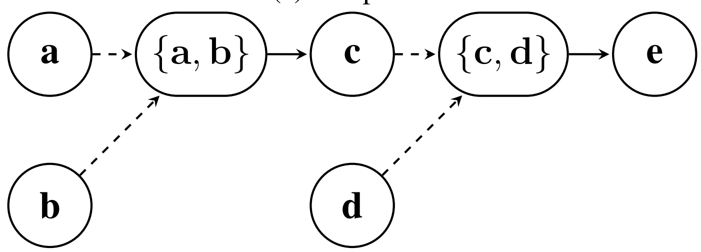
One of the issues we had before in conflict–freeness was the fact that e.g. a, b and c could be jointly accepted in the target framework. However, if we observe that the presence of a and b is sufficient for the $\{a, b\}$ argument, then it makes sense to derive a supported conflict from a and b to c. Nevertheless, this is a group form of attack, and as such it is not present in BAFs.

The previously analyzed translations were not semantics bijective under admissibility. This was due to the fact that we did not always have to accept attack arguments in an extension even though it was perfectly possible. For example, in our case both $\{a, b, \{a, b\}\}$ and $\{a, b\}$ could be considered d and i–admissible. By enforcing closure, we could ensure that only the first set was produced, and thus retrieve a one–to–one relation with the SETAF extensions. However, since more than one supporter can be required in order to accept an attack argument, the closure we would need would be a group one, not the one present in Definition 2.70.

This example has shown that it might be possible to create a SETAF–BAF translation that would behave better than any of the SETAF–AF approaches we have considered. It has also shown that even though supported attack and closure are more associated with deductive support rather than necessary [30], certain applications motivate their use with this interpretation of support as well. Nevertheless, the group versions that we would like to use in our method are not yet present in BAFs. Those aspects that are already defined reflect the way necessary support is modeled in AFNs. Consequently, we will continue with this approach in the next section.



(a) Sample SETAF



(b) Sample BAF representation

Figure 57: Sample SETAF and its possible BAF representation

6.3 SETAF as AFN

The translation from SETAFs to AFNs is simpler than into AFs, even though both of the structures use binary attack. Since we have support at hand, the argument representing a given group attack is no longer detached from the ones that originally carry out the attack. Moreover, attacking the argument contained in a group attack is sufficient for defending ourselves from the actual attack argument. Consequently, the conflict propagation that we have observed in the SETAF–AF translation (Translation 25) is no longer an issue. This brings us to the following definition:

Translation 28. ***Deprecated*** *Let $SF = (A, R)$ be a SETAF. Its corresponding AFN $FN^{SF} = (A', R', N)$ is built the following way:*

- *let $arg(S) = \{\{a\} \mid a \in S\}$, where $S \subseteq A$, be a function returning a collection of single element sets composed of elements of S,*
- *let $att(S) = \{S' \mid S' \subseteq S \wedge |S'| > 1 \wedge \exists y \in A\, s.t.\, (S', y) \in R\}$, where $S \subseteq A$, be a function returning subsets of S of size bigger than 1, which attack an argument in A,*
- *$A' = arg(A) \cup att(A)$,*
- *$R' = \{(X, \{y\}) \mid (X, y) \in R\}$, and*
- *$N = \{(\{X\}, Y) \mid X \in arg(A), Y \in att(A), X \subseteq Y\}$.*

However, the described approach can be further simplified, similarly to Translation 27. Due to the fact that an attack argument cannot appear without its supporters in a coherent (and thus admissible) extension, we do not "lose" any of the semantics as it was in the AF case. Therefore, we will work with the following formulation:

Translation 29. *Let $SF = (A, R)$ be a SETAF. Its corresponding AFN $FN^{SF} = (A', R', N)$ is built the following way:*

- *let $att(S) = \{S' \mid S' \subseteq S \wedge |S'| > 1 \wedge \exists y \in A\, s.t.\, (S', y) \in R\}$, where $S \subseteq A$, be a function returning subsets of S of size bigger than 1, which attack an argument in A,*
- *$A' = A \cup att(A)$,*
- *$R' = \{(X, y) \mid (X, y) \in R$ or $(\{X\}, y) \in R\}$, and*
- *$N = \{(\{X\}, Y) \mid X \in A, Y \in att(A), X \in Y\}$.*

The SETAF–produced AFNs have a number of interesting properties. First of all, it is support binary, i.e. every supporting set will consist of precisely one argument. This means that the minimality of the target AFN is independent of the minimality of the source SETAF. By tracing the support edges in the graph we obtain paths of length 1 – attack arguments are only supported by the normal arguments carrying out the conflict, and no normal argument needs support through N. We can also observe that the attack arguments cannot be directly attacked. Moreover, we have no support cycles, and by using Theorem 4.34 we can show that our framework is strongly valid. Therefore, our AFN is on the intersection of the $SBin^{AFN}$, Sup_1^{AFN}, $SCons^{AFN}$ and SV^{AFN} classes, which puts it in the group of well–structured and elementary AFNs of depth 1. However, please note that it is still not the most accurate description of the produced frameworks. The listed classes do not grasp the fact that an argument can be supported or attacked, but not both at the same time, and that every support argument has to carry out an attack. Nevertheless, by Lemma 4.71, we already have all of the desired normal forms:

Theorem 6.5. *Let $SF = (A, R)$ be a SETAF and $FN^{SF} = (A', R', N)$ its corresponding AFN obtained through Translation 29. FN^{SF} is minimal, (strongly) consistent, weakly, relation and strongly valid.*

We can now move on to the semantics analysis. Please note that this translation does not straightforwardly preserve conflict–freeness. Let us explain it on an example.

Example 83. Let $SF_1 = (\{a, b, c\}, \{(\{a, b\}, c)\})$ be a SETAF. In total, we create four arguments for the corresponding AFN: $A_1 = \{a\}$, $A_2 = \{b\}$, $A_3 = \{c\}$ and $A_4 = \{a, b\}$. This gives us a framework $FN_1^{SF} = (\{A_1, A_2, A_3, A_4\}, \{(A_4, A_3)\}, \{(\{A_1\}, A_4), (\{A_2\}, A_4)\})$. While the set $\{a, b, c\}$ is not conflict free in SF_1, $\{A_1, A_2, A_3\}$ (i.e. $\{\{a\}, \{b\}, \{c\}\}$) is conflict–free in FN_1^{SF}. Only if we take into account the attack arguments induced by the members of the set, we will have that $\{A_1, A_2, A_3, A_4\}$ (i.e. $\{\{a\}, \{b\}, \{a, b\}, \{c\}\}$) is not AFN conflict–free.

Theorem 6.6. *Let $SF = (A, R)$ be a SETAF and $FN^{SF} = (A', R', N)$ its corresponding AFN obtained by Translation 29. If E is conflict–free in SF, then E is conflict–free in FN^{SF}. If E is a σ–extension of SF, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, then $E' = E \cup att(E)$ is a σ–extension of FN^{SF}. If $E' \subseteq A'$ is conflict–free in FN^{SF}, then $E = E' \cap A$ might not be conflict–free in SF. If E' is a σ'–extension of FN^{SF}, where $\sigma' \in \{$admissible, preferred, complete, grounded, stable$\}$, then $E = E' \cap A$ is a σ'–extension of SF.*

Please note that the same proof, but with minor adjustments, can be used to show that Theorem also holds for the deprecated Translation 28.

We can now enter our translation into the system.

Redefinition of Translation 29: *Let Fr^{SETAF} be the collection of all SETAFs on domain $\mathcal{U}$ and $WSt^{AFN} \cap SEle_1^{AFN}$ the collection of well–structured and elementary AFNs of depth 1 on domain $\mathcal{U} \cup (2^{\mathcal{U}} \setminus \emptyset)$. The translation $Tr_{AFN}^{SETAF} : Fr^{SETAF} \rightarrow (WSt^{AFN} \cap SEle_1^{AFN})$ is defined as $Tr_{AFN}^{SETAF}((A, R)) = (A', R', N)$ for $(A, R) \in Fr^{SF}$, where $A' = A \cup att(A)$ for $att(A) = \{S' \mid S' \subseteq A \wedge |S'| > 1 \wedge \exists y \in A \, s.t. \, (S', y) \in R\}$, $R' = \{(X, y) \mid (X, y) \in R$ or $(\{X\}, y) \in R\}$ and $N = \{(\{X\}, Y) \mid X \in A, Y \in att(A), X \in Y\}$.*

Redefinition of Theorem 6.6: *Let $\sigma \in \{$admissible, complete, preferred, grounded, stable$\}$ be a semantics and SC_σ^{Tr} the removal casting functions for σ defined as $SC_\sigma^X(E) = E \cap A$, where $X \in Fr^{SETAF}$ is a framework with set of arguments A and $E \in \sigma(Tr_{AFN}^{SETAF}(X))$. The translation Tr_{AFN}^{SETAF} is strong under (σ, SC_σ^{Tr}). It is $\subseteq$–weak under the conflict–free semantics and removal casting functions. It is semantics bijective under complete, preferred, grounded and stable semantics and their removal casting functions.*

Analysis of Translation 29: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and removal casting functions, the translation Tr_{AFN}^{SETAF} is:

- full, target–subclass and injective
- argument domain altering, argument introducing, induced support relation introducing, attack relation preserving
- generic and semantics domain altering
- structural and modular

Translation Tr_{AFN}^{SETAF} is faithful under the complete, preferred, grounded and stable semantics and removal casting functions. We classify the translation as a basic–coalition approach under the listed semantics and casting functions.

Explanation. Many of the properties are similar as in Translation 25. Thus, we will only remark on some of them. Let us show why the translation is injective. We can consider two SETAFs $SF_1 = (A_1, R_1)$ and $SF_2 = (A_2, R_2)$ producing the same AFN $FN = (A, R, N)$. From the construction of A we can observe that it has to be the case that A_1 = A_2. The attack relation R in FN also shows that R_1 and R_2 have to be the same and thus $SF_1 = SF_2$. The framework adds a support relation between the new attack arguments and normal arguments, thus we can talk about induced introducing. It is however attack relation preserving, even though the group attacks can now be expressed with arguments.

Unlike the Translation 25, Tr_{AFN}^{SETAF} is easily shown to be modular. Joining two SETAFs will not produce arguments or attacks that were not present in at least one of the frameworks. Consequently, in the case of Tr_{AFN}^{SETAF}, the set of arguments and attacks in the AFN created by the union of translated SETAFs or translation of the union will be the same. Similar follows for the support relation. The fact that the translation is faithful under complete, preferred, grounded and stable semantics and the defined removal casting functions follows from the redefinition of Theorem 6.6. ■

Please note we can try to reclassify this translation as basic. However, the same explanation as in the case of Translation 27 holds. We would have to discard the content of the attack arguments and accept that the translation would become lossy.

Example 84. Let us come back to the $SF = (A, R)$ with the set of arguments $A = \{a, b, c, d, e\}$ and the attack relation $R = \{(\{a\}, c)$, $(\{a\}, b)$, $(\{b\}, a)$, $(\{c\}, d)$, $(\{e\}, a)$, $(\{b, d\}, e)\}$, previously analyzed in Example 3. The associated AFN created using Translation 29 is $FN^{SF} = (A', R', N)$, where $A' = \{a, b, c, d, e, \{b, d\}\}$, $R' = \{(a, c)$, (a, b), (b, a), (c, d), (e, a), $(\{b, d\}, e)\}$ and $N = \{(\{b\}, \{b, d\}), (\{d\}, \{b, d\})\}$. We can see it depicted in Figure 58. It is worth nothing that $\{b, d\}$ possesses only one minimal coherent set on A', namely $\{b, d, \{b, d\}\}$. Consequently, both $\{a\}$ and $\{c\}$ attack all coherent sets of the attack argument $\{b, d\}$. The admissible extensions of FN^{SF} include $\emptyset$, $\{b\}$, $\{b, c\}$, $\{c, e\}$ and $\{b, c, e\}$, which were the desired answers. In this case, no filtering is needed. The complete extensions are $\emptyset$ and $\{b, c, e\}$, with the first being grounded and the other both preferred and stable. Thus, we retrieve the original extensions of SF.

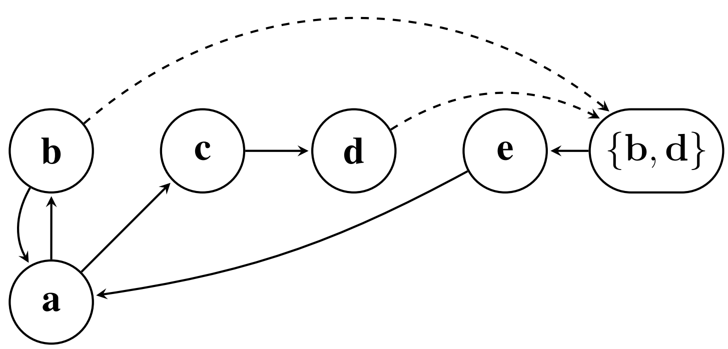


Figure 58: An AFN created for a SETAF

6.3.1 Improvements

At this point we can ask ourselves the question whether it is possible to create a full and exact translation from SETAFs to AFNs. Due to the fact that the signatures of the AFN semantics are not yet researched, we cannot give a complete answer at this point. However, we can show that a collection of admissible (preferred) extensions of a given AFN in fact meets the signature requirements of the AF admissible (preferred) semantics (please consult Section 10.1.1). This, jointly with the fact that SETAF semantics go beyond what AFs can express (see Section 6.1.3), means that the following holds:

Theorem 6.7. *Let Fr^{SETAF} be the collection of all SETAFs on the domain $\mathcal{U}^{SETAF}$ and Fr^{AFN} the collection of all AFNs on the domain $\mathcal{U}^{AFN}$. There does not exist a full translation from Fr^{SETAF} to Fr^{AFN} that is exact under the admissible (preferred) semantics and identity casting functions.*

Please note that due to the fact that the stable semantics for AFNs do not conform to the stable AF signature, it is possible that an exact translation from SETAFs to AFNs can be created in this case. We leave answering this question for future work.

6.4 SETAF as EAS

The translation of SETAFs into EASs was proposed in [73]. Since EASs structurally permit set conflict in the same way as SETAFs, the only thing required for preserving the behavior of semantics is including the support from evidence to every argument. The approach is quite straightforward and the same analysis as done in case of AF–EAS translation (see Section 5.6) holds here.

Translation 30. *Let $SF = (A, R)$ be a SETAF. The corresponding EAS is $ES^{SF} = (A', R, E)$, where $A' = A \cup \{\eta\}$ and $E = \{(\{\eta\}, a) \mid a \in A\}$.*

The EASs produced by SETAFs are not that far from those produced by AFs and again we find ourselves at an intersection of various subclasses. The only difference is that in the SETAF case, there are no restrictions concerning the attack relation. Thus, we still deal with support binary EASs, where every supporting set is just $\{\eta\}$ and every argument receives support from it. Consequently, the intersection of $EvSup^{EAS}$ and $AllSup^{EAS}$ finally gives us the description of the subclass $SETAF^{EAS}$ of SETAF–produced EASs. This class satisfies all of the EAS normal forms from Section 4 with the exception of the minimal one:

Theorem 6.8. *Let $SF = (A, R)$ be a SETAF and $ES^{SF} = (A', R, E)$ its corresponding EAS obtained through Translation 30. ES^{SF} is consistent, weakly, relation and strongly valid. If SF is minimal, then so is ES^{SF}.*

Please note that just like in the AF–EAS case, $\emptyset$ is trivially admissible and conflict–free in ES^{SF}. Consequently, the $\emptyset$ extension in SF can be obtained both from $\emptyset$ and

$\{\eta\}$. The amount of conflict–free extensions also increases due to the fact that η cannot attack (or be attacked) in the framework – in other words, every SETAF conflict–free set will have two corresponding EAS ones. We can also observe that every subset of SETAF arguments (extended with η) will be self supporting in the target framework. We can now recall the theorem concerning the behavior of the semantics and proceed with entering the translation into our system:

Theorem 6.9. *Let $SF = (A, R)$ be a SETAF and $ES^{SF} = (A', R, E)$ its corresponding EAS obtained by Translation 30. A set of arguments $S \subseteq A$ is a σ–extension of SF, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$* [19]*, iff $S \cup \{\eta\}$ is a σ–extension of ES^{SF}.*

Redefinition of Translation 30: *Let Fr^{SETAF} be the collection of all SETAFs on domain $\mathcal{U}$ and $SETAF^{EAS}$ the EASs with every argument being supported by and only evidence on domain $\mathcal{U} \cup \{\eta\}$. The translation $Tr^{SETAF}_{EAS} : Fr^{SETAF} \rightarrow SETAF^{EAS}$ is defined as $Tr^{SETAF}_{EAS}((A, R)) = (A', R, E)$, where $A' = A \cup \{\eta\}$ and $E = \{(\{\eta\}, a) \mid a \in A\}$.*

The reason for our choice of the domains is to make sure that η, or however we want to designate the evidence argument, is not in $\mathcal{U}$. Although EASs can be translated into SETAFs (see Translations 73 and 75) and SETAFs back to EASs, the η argument cannot be "reused". During the EAS–SETAF shift, the existing evidence can start carrying out attacks. Consequently, it does not meet EAS evidence requirements anymore and during the SETAF–EAS translation, a new argument playing this role needs to be added.

Redefinition of Theorem 6.9: *Let $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} their removal casting functions defined as $SC^{X}_{\sigma}(S) = S \cap A$, where $X = (A, R) \in Fr^{SETAF}$ and $S \in \sigma(Tr^{SETAF}_{EAS}(X))$. The translation Tr^{SETAF}_{EAS} is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is semantics bijective under the complete, preferred, grounded and stable semantics and the removal casting functions.*

The same explanations as in the analysis of Translation 22 hold; thus, we will omit them here.

Analysis of Translation 30: Under the conflict–free, admissible, complete, preferred, grounded and stable semantics and the removal casting functions, the translation Tr^{SETAF}_{EAS} is:

- full, target–subclass and injective
- weakly argument domain altering, argument introducing, induced support introducing and attack relation preserving
- generic and weakly semantics domain altering
- modular and semi–structural

[19] Although the complete semantics was defined only later [78], the results still hold.

Under the complete, preferred, grounded and stable semantics and the removal casting functions, the translation is faithful.

The translation Tr^{SETAF}_{EAS} is classified as basic under the listed semantics and casting functions.

Again, similarly as in the AF–EAS approach, translation Tr^{SETAF}_{EAS} is in fact weakly faithful (see Section 3.2.3) under the admissible semantics. This means that if we exclude the empty set from the EAS extensions, there is a bijection between what is left and the original SETAF answers. Moreover, for the same reasons as in Section 5.6.1, our results cannot be further improved. The translation will always add the η argument that will need to be removed:

Theorem 6.10. *Let Fr^{SETAF} be the collection of all SETAFs on domain $\mathcal{U}^{SETAF}$ and Fr^{EAS} the collection of all EASs on domain $\mathcal{U}^{AF}$. There exists no full translation from Fr^{SETAF} to Fr^{EAS} that is exact under conflict–free, admissible, complete, preferred and stable semantics and identity casting functions for them.*

Example 85. We will again continue with the framework $SF = (A, R)$ with the set of arguments $A = \{a, b, c, d, e\}$ and the attack relation $R = \{(\{a\}, c)$, $(\{a\}, b)$, $(\{b\}, a)$, $(\{c\}, d)$, $(\{e\}, a)$, $(\{b, d\}, e)\}$, previously analyzed in Example 3. The associated EAS is quite straightforward; it is simply $ES^{SF} = (A \cup \{\eta\}, R, E)$, where $E = \{(\{\eta\}, a)$, $(\{\eta\}, b)$, $(\{\eta\}, c)$, $(\{\eta\}, d)$, $(\{\eta\}, e)\}$. We can observe that for every argument $a \in A$, $\{\eta, a\}$ is a minimal self–supporting set for a. This also means that every set of arguments that contains η also e–supports every other argument in ES^{SF}. To every conflict (X, y) in R, we can create an associated minimal e–supported attack $X \cup \{\eta\}$. Since η cannot be attacked, then defense can occur only through attacking an argument in X. We can therefore show that $\emptyset$, $\{\eta\}$, $\{\eta, b\}$, $\{\eta, b, c\}$, $\{\eta, c, e\}$ and $\{\eta, b, c, e\}$ are admissible in ES^{SF}. By filtering out η, we obtain the admissible extensions of SF. We can observe that $\emptyset$ can be produced both from $\emptyset$ and $\{\eta\}$. The complete extensions of ES^{SF} are $\{\eta\}$ and $\{\eta, b, c, e\}$. This gives us the desired SF sets. This time, the relation is one–to–one. We can easily show that the grounded, stable and preferred extensions of both frameworks are also in correspondence.

6.5 SETAF as ADF

The hidden conjunctive nature of the Dung's framework has already been noted by the authors of SETAF. In a certain sense, it is reflected by the translation from AFs to ADFs (see Section 5.7). The intuition behind SETAF was to somehow relax this constraint by allowing a disjunctive behavior, which again has its counterpart in ADFs. Let a be an argument and $X = \{X_1, ..., X_n\}$ the collection of all and only sets attacking a, i.e. sets s.t. $X_i Ra$. Only the presence of all members of X_i, not just some of them, renders a unacceptable w.r.t. a set of arguments. Therefore given any set of arguments Y that does not include any of the attackers, i.e. there is no X_i s.t. $X_i \subseteq Y$, the acceptance condition of a is in. Consequently, for any other set it is out. The propositional version is simply $att_a = \bigvee \neg X_1 \wedge ... \wedge \bigvee \neg X_n$, where $X_i = \{x^i_1, ..., x^i_{n_i}\}$ and $\bigvee \neg X_i = \neg x^i_1 \vee ... \vee \neg x^i_{n_i}$.

Translation 31. *Let $SF = (A, R)$ be a SETAF. Its corresponding ADF $D^{SF} = (A, L, C)$ is the following:*

1. *for every $x, y \in A$ s.t. $\exists B \subseteq A$, $x \in B$ and BRy, add (x, y) to L, and*
2. *for every argument $a \in A$ create an acceptance condition C_a. Let $X = \{X_1, ..., X_n\}$ be the collection of all sets of arguments s.t. $X_i Ra$:*
   - *the functional acceptance condition maps to out all and only subsets $B \subseteq \bigcup X$ s.t. $\exists X_i \in X$ for which $X_i \subseteq B$. All remaining subsets are in.*
   - *the propositional acceptance condition is $C_a = att_a = \bigvee \neg X_1 \land ... \land \bigvee \neg X_n$. In case X is empty, it is simply $\top$.*

Remark. Let $a \in A$ be an arbitrary argument, $X^a = \{X_1^a, ..., X_n^a\}$ the collection of the sets of arguments that attack it and let C_a be its acceptance condition created as above. If we focus on the functional representation, we can see that an interpretation v is decisively in for a iff $\forall B \subseteq par(s)$ s.t. $C_a(B) = out$, $\exists b \in B$, $v(b) = \mathbf{f}$. This means we prevent an attacking set from appearing by falsifying at least one member of every such set, which naturally corresponds to the way we would defend in SETAFs. In terms of propositional representation, we want to make sure that all disjunctions corresponding to the elements of X^a will always evaluate to true, which is achieved by setting at least one argument of every clause to $\mathbf{f}$. In other words, assuming that the condition is of the form $C_a = \bigvee \neg X_1^a \land ... \land \bigvee \neg X_n^a$, it would be the case that for a decisively in interpretation v, $\forall X_i^a \in X^a$, $v^{\mathbf{f}} \cap X_i^a \neq \emptyset$.

Any minimal interpretation v will try to minimize $v^{\mathbf{f}}$ and naturally does not contain any $\mathbf{t}$ mappings, i.e. $v^{\mathbf{t}} = \emptyset$. Consequently, a pd–evaluation built with v would be of the form $((a), v^{\mathbf{f}})$. Interpretations for which a would be decisively out map to $\mathbf{t}$ all members of at least one attacking set, i.e. given an interpretation z, $\exists X_i^a \in X^a$ s.t. $\forall x \in X_i^a$, $z(x) = \mathbf{t}$. This also naturally means that z outs the condition, i.e. $C_a(z^{\mathbf{t}} \cap par(a)) = out$. Finally, any interpretation z' which outs C_a, is also decisively out for a.

Just like in the case of AFs, it is no surprise that SETAF–style ADFs will be AADF$^+$s and BADFs as well. They will also be in the cleansed form and weakly valid form. However, since the redundancy–freeness of the produced ADF depends on the minimality of the source SETAF and as such is required for relation and strong validity (see the discussion in Section 4.3.2.3), further forms require some assumptions on the source framework.

Theorem 6.11. *Let $SF = (A, R)$ be a SETAF and $D^{SF} = (A, L, C)$ its corresponding ADF obtained through Translation 31. Then D^{SF} is an AADF$^+$ and a BADF. It is also cleansed and weakly valid. If SF is minimal, then D^{SF} is redundancy–free, relation and strongly valid.*

Please note that it can happen that D^{SF} is redundancy–free, even if SF is not, as already noted in Section 4.1.5. Just because the source framework is not minimal and some attacks are going to be removed, it does not necessarily mean that the related arguments will not be attackers of a given argument anymore.

We can now show the relation between a given SETAF and its corresponding ADF.

Theorem 6.12. *Let $SF = (A, R)$ be a SETAF and $D^{SF} = (A, L, C)$ its corresponding ADF obtained through Translation 31. A set of arguments E is a conflict–free extension of SF iff it is a conflict–free extension of D^{SF}.*

Lemma 6.13. *Let $SF = (A, R)$ be a SETAF and $D^{SF} = (A, L, C)$ its corresponding ADF obtained through Translation 31. Let E be a conflict–free extension of SF (and thus of D^{SF}). The discarded set of E in SF coincides with the discarded set of E in D^{SF}.*

Lemma 6.14. *Let $SF = (A, R)$ be a SETAF and $D^{SF} = (A, L, C)$ its corresponding ADF obtained through Translation 31. A conflict–free set of arguments E defends an argument $a \in A$ in SF iff a is decisively in w.r.t. v_E in D^{SF}.*

With this at hand, we come to the final result. Please note that since by Theorems 2.172 and 6.11 our classification collapses, it does not really matter what type of prefixing we assume.

Theorem 6.15. *Let $SF = (A, R)$ be a SETAF and $D^{SF} = (A, L, C)$ its corresponding ADF obtained through Translation 31. A set of arguments $E \subseteq A$ is a conflict–free extensions of SF iff it is (pd–acyclic) conflict–free in D^F. $E \subseteq A$ is a stable extensions of SF iff it is (stable) model of D^F. $E \subseteq A$ is a grounded extensions of SF iff it is (acyclic) grounded in D^F. $E \subseteq A$ is a σ extensions of SF, where where $\sigma \in \{$admissible, preferred, complete$\}$ iff it is an xy–σ–extension of D^F for $x, y \in \{a, c\}$.*

We can now put the translation into our system.

Redefinition of Translation 31: *Let Fr^{SETAF} be the collection of all SETAFs and $SETAF^{ADF}$ the collection of all SETAF–style ADFs, both based on argument domain $\mathcal{U}$. The translation $Tr_{ADF}^{SETAF} : Fr^{SETAF} \rightarrow SETAF^{ADF}$ is defined as $Tr_{ADF}^{SETAF}((A, R)) = (A, L, C)$ for a framework $(A, R) \in Fr^{SETAF}$, where $L = \{(x, y) \mid \exists X \subseteq A, x \in X, (X, y) \in R\}$ and $C = \{C_a \mid a \in A\}$ and given the collection $X = \{X_1, ..., X_n\}$ of all sets of arguments s.t. $X_i Ra$, C_a is defined as a) $C_a(B) = out$ for $B \subseteq \bigcup X$ s.t. $\exists X_i \in X, X_i \subseteq B$ and $C_a(B) = in$ for remaining $B \subseteq \bigcup X$; or b) $C_a = \bigvee \neg X_1 \wedge ... \wedge \bigvee \neg X_n$ if $X \neq \emptyset$ and $C_a = \top$ otherwise.*

Redefinition of Theorem 6.15: *Let $\sigma^{SETAF} \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ be a SETAF semantics and $\sigma^{ADF} \in \{$conflict–free, pd–acyclic conflict–free, xy–admissible, xy–complete, xy–preferred, grounded, acyclic grounded, model, stable$\}$ a similar ADF semantics with $x, y \in \{a, c\}$. Let SC_σ^{Tr} be the identity casting functions for σ. The translation Tr_{ADF}^{SETAF} is strong and semantics bijective under (σ, SC_σ^{Tr}).*

Analysis of Translation 31: Under the (pd–acyclic) conflict–free, (xy–) admissible, (xy–) complete, (xy–) preferred, (acyclic) grounded and (stable) model semantics with $x, y \in \{a, c\}$ and the identity casting functions, the translation Tr_{ADF}^{SETAF} is:

- full, target–subclass and overlapping
- argument domain preserving and structure preserving
- generic, semantics domain preserving and exact
- $\otimes$–modular and structural

The translation is not $\oplus$–modular. The translation Tr_{ADF}^{SETAF} is classified as basic under the listed semantics and casting functions.

Explanation. Any SETAF can be translated into an ADF and clearly SETAF–style ADFs do not account for all possible frameworks. Consequently, our approach is full and target–subclass. Unfortunately, it is also overlapping.

Consider the SETAFs $SF_1 = (\{a,b,c,d\}, \{(\{a,b\},c), (\{a,d\},c), (\{a,b,d\},c)\})$ and $SF_2 = (\{a,b,c,d\}, \{(\{a,b\},c), (\{a,d\},c)\})$. We can observe that SF_2 is in fact the minimal form of SF_1. Upon being translated into ADFs, the argument c would have the same (functional) condition in both frameworks, namely $C_c = \{\emptyset = in, \{a\} = in, \{b\} = in, \{d\} = in, \{a,b\} = out, \{a,d\} = out, \{a,b,d\} = out\}$. Please note the propositional versions would not be exactly the same, though still equivalent. Nevertheless, it appears that the translation can be made injective if we limit our domain only to minimal SETAFs. Clearly, Tr_{ADF}^{SETAF} is argument and semantics domain preserving and generic. We can observe that no arguments are added or removed during the translation. Moreover, as seen from the definition of L, all connections are preserved between the arguments. The links are also considered attacking (see Theorem 6.11), even though some are redundant. Therefore, despite the fact that the translation is overlapping, it is structure preserving, and this state of affairs is unavoidable due to the differences between ADFs and SETAFs (see Section 2.3.9).

Our translation is clearly structural. Its exactness follows easily from Theorem 6.15 and its redefinition. The lack of $\oplus$–modularity follows from the fact that AFs are special cases of SETAFs and the AF–ADF translation is not $\oplus$–modular (see analysis of Translation 23). Please note that this type of modularity can come in handy when different, not union–based, joining of frameworks is considered. However, the approach is $\otimes$–modular, and the explanation is similar to the AF case. ■

Example 86. We now come back to our SETAF for the last time. Let $SF = (A, R)$ be a framework with the set of arguments $A = \{a,b,c,d,e\}$ and the attack relation $R = \{(\{a\},c), (\{a\},b), (\{b\},a), (\{c\},d), (\{e\},a), (\{b,d\},e)\}$, previously analyzed in Example 3. The associated ADF is $D^{SF} = (A, L, C)$, where $L = \{(a,b), (b,a), (a,c), (c,d), (e,a), (b,e), (d,e)\}$ and the set acceptance conditions C is as follows; $C_a = \neg b \wedge \neg e$, $C_b = \neg a$, $C_c = \neg a$, $C_d = \neg c$ and $C_e = \neg b \vee \neg d$ (see Figure 59). D^{SF} is an AADF$^+$ and therefore we can focus on acyclic evaluations only. With the exception of e, every argument has a single minimal evaluation. For a we produce $((a), \{b,e\})$, for b we have $((b), \{a\})$, then $((c), \{a\})$ for c and $((d), \{c\})$ for d. Concerning e, we have two evaluations, namely $((e), \{b\})$ and $((e), \{d\})$. The conflict–free (and at the same time, pd–acyclic conflict–free) extensions of D^{SF} are $E_1 = \emptyset$, $E_2 = \{a\}$, $E_3 = \{b\}$, $E_4 = \{c\}$, $E_5 = \{d\}$, $E_6 = \{e\}$, $E_7 = \{a, d\}$, $E_8 = \{b, c\}$, $E_9 = \{b, d\}$, $E_{10} = \{b, e\}$, $E_{11} = \{c, e\}$, $E_{12} = \{d, e\}$ and $E_{13} = \{b, c, e\}$. Their discarded sets of any type are respectively $E_1^+ = E_5^+ = \emptyset$, $E_2^+ = E_7^+ = \{b, c\}$, $E_3^+ = E_6^+ = E_{10}^+ = E_{12}^+ = \{a\}$, $E_4^+ = \{d\}$, $E_8^+ = E_{11}^+ = E_{13}^+ = \{a, d\}$ and $E_9^+ = \{a, e\}$. Therefore, from these conflict–free extensions, only E_1, E_3, E_8, E_{11} and E_{13} are admissible in D^{SF} (independently of the used prefixing). The sets E_1 and E_{13} are also complete. For every other extension, we can find an argument decisively in w.r.t. its range that is outside the set; for E_3, it is c, for E_8 it's e, for E_{11} it is b. E_1 is also our standard and acyclic grounded extension, while E_{13} is preferred, model and stable. These answers are exactly the same as in the original SETAF SF.

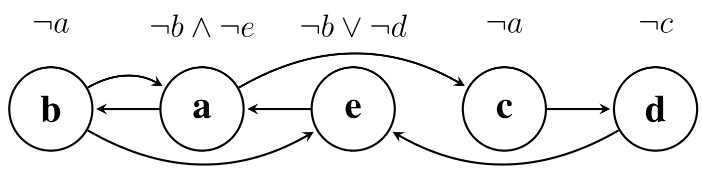


Figure 59: ADF associated with SF

6.6 SETAF as Other Frameworks

Although we have discussed various translations, we have not analyzed the conversions from SETAFs to AFRAs, EAFs and EAFCs. To the best of our knowledge, these approaches were not discussed in the literature. Moreover, if they were and we have just overlooked them, we do not think that the resulting SETAF–AFRA and SETAF–AF translations would be much different from the SETAF–AF transformations. Furthermore, even though EAFCs permit group relations, their nature is more positive than negative (see Section 8). Therefore, it is quite probable that again we would come back to the described construction. Consequently, for now we propose that the SETAF–AF–AFRA/EAF/EAFC chained translation should be used.

6.7 Summary

In this section we have focused on translating SETAFs, the first of our frameworks that cannot be translated exactly into AFs. We could have observed that out of all the approaches, it is the coalition SETAF–AF translation that was the weakest. Moreover, it suffered from the loss of modularity, which is not the case in any other approach. The only case in which we have managed to obtain exact results is the SETAF–ADF translation. However, please note that even though formally the SETAF–EAS approach is classified as faithful, it in fact satisfies the weakly exact restrictions from [42] that were mentioned in

Section 3.2.3. After all, for any (non–empty) evidential framework, it suffices to remove the evidence argument to retrieve the source SETAF. Although the presented SETAF–AFN translation is only faithful, it might be possible that an exact translation can be obtained for the stable semantics. Due to the lack of research on semantics signatures in AFNs, this task is left for future work. The summary of our results can be seen in Table 8.

Table 8: Translations from SETAFs to other frameworks

Properties		AF		AFN	EAS	ADF
Translation		25	26	29	30	31
Strength	cf	$\subseteq$–weak		$\subseteq$–weak	strong	exact
	adm	strong	strong	strong	strong	exact
	comp	strong	faithful	faithful	faithful	exact
	pref	strong	faithful	faithful	faithful	exact
	grd	strong	faithful	faithful	faithful	exact
	stb	strong	faithful	faithful	faithful	exact
Functional		full	full	full	full	full
		target–subclass	target–subclass	target–subclass	target–subclass	target–subclass
		injective	injective	injective	injective	overlapping
Syntactical		argument domain altering	argument domain altering	argument domain altering	weakly argument domain altering	argument domain preserving
		argument introducing	argument introducing	argument introducing	argument introducing	structure preserving
		induced attack introducing	induced attack introducing	induced support introducing	induced support introducing	
Semantical		generic	generic	generic	generic	generic
		semantics domain altering	semantics domain altering	semantics domain altering	weakly semantics domain altering	semantics domain preserving
Computational		structural	semi–structural	structural	semi–structural	structural
			modular	modular	modular	$\otimes$–modular

7 Translating AFRAs

7.1 AFRA as AF

In this section we will show how AFRAs can be translated to AFs, BAFs and AFNs. Additionally, we will also provide a discussion on the differences between recursive and defense attacks (i.e. EAFs). As we might recall, the most prominent difference between the semantics of AFRA and any other frameworks concerned the explicit presence of attacks in the extensions. Consequently, in order to obtain the desired correspondence, the AFRA conflicts will be transformed into arguments in the target frameworks. The indirect defeats can then be turned into actual attacks or simulated with indirect conflicts derived from support. Depending on how (if at all) we choose to connect the AFRA attacks to their sources in the target frameworks, we will differentiate between basic, attack propagation and defender approaches.

7.1.1 Standard Translation

The translation from AFRAs into AFs was introduced in [9]. Recall that AFRAs, unlike AFs, elevate attacks to the level of arguments and allow them to appear in the extensions. Consequently, in order to go back to AFs – or in fact, any other more "traditional" framework – the attacks need to be transformed into arguments so that we can simulate the way they are treated in AFRAs. However, in the Dung's framework, the attack argument can become detached from it sources, and thus the propagation of indirect conflicts becomes necessary:

Translation 32. *Let $FR = (A, R)$ be an AFRA. The corresponding AF $F^{FR} = (A', R')$ is defined as follows:*

- *$A' = A \cup R$, and*
- *$R' = \{(V, W) \mid V, W \in A \cup R$ and V defeats $W\}$.*

Theorem 7.1. *Let $FR = (A, R)$ be an AFRA and $F^{FR} = (A', R')$ its corresponding AF obtained through Translation 32. Then $S \subseteq A \cup R$ is a σ–extension of FR, where $\sigma \in$ {conflict–free, admissible, preferred, complete, stable, grounded} iff S is a σ–extension of F^{FR}.*

The redefinitions of the available results are now the following:
Redefinition of Translation 32: *Let Fr^{AFRA} be the collection of all AFRAs on the domain $\mathcal{U}$ and Fr^{AF} the collection of all AFs on the domain $\bigcup_{i=1}^{\infty} \mathcal{U}^i$, where $\mathcal{U}^1 = \mathcal{U}$ and $\mathcal{U}^i = \mathcal{U} \times \mathcal{U}^{i-1}$ for $i > 1$. The translation $Tr_{AF}^{AFRA} : Fr^{AFRA} \to Fr^{AF}$ is defined as $Tr_{AF}^{AFRA}((A, R)) = (A', R')$, where $A' = A \cup R$ and $R' = \{(V, W) \mid V, W \in A \cup R$ and V defeats $W\}$.*

Redefinition of Theorem 7.1: *Let $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ be a semantics and SC_σ^{Tr} the identity casting functions for σ. The translation Tr_{AF}^{AFRA} is strong and semantics bijective under (σ, SC_σ^{Tr}).*
Analysis of Translation 32: Under the conflict–free, admissible, complete, preferred, grounded and stable semantics and identity casting functions, the translation Tr_{AF}^{AFRA} is:

- full, target–subclass and injective
- argument domain altering, argument introducing and induced attack relation introducing
- generic, semantics domain preserving and exact
- semi–structural

Translation Tr_{AF}^{AFRA} is not modular. It is classified as basic–attack propagation hybrid under the listed semantics and casting functions.

Explanation. Since every AFRA can be translated into an AF, the translation is full. However, it is target–subclass, and there are two reasons for this situation. First of all, taking into account how the domain for Fr^{AF} is defined, we can observe that not every choice of arguments can represent an AFRA (for example, argument (c, d) cannot appear on its own, c and d ought to be present). Similar follows for attacks – for example, if argument c is attacked, then so should any argument (c, x). However, putting the domain aside, we cannot produce an AF that would consist only of two (different) arguments a and b and an attack (a, b). Let us assume the opposite and try to "reconstruct" the possible AFRA. Since b does not attack anything, it has to be an argument. Thus, it can only be the case that a represents an attack argument. The only candidate for its source is b and thus we have an AFRA $(\{b\}, \{a = (b, b)\})$. However, Tr_{AF}^{AFRA} is $(\{a, b\}, \{(a, b), (a, a)\})$, not $(\{a, b\}, \{(a, b)\})$. Therefore, it appears that with this translation we cannot produce an AF that would consist only of two arguments and a single conflict between them.

The translation is clearly injective due to the way the argument set is defined in the target AF – it cannot be the case that two different AFRAs would produce the same union of arguments and attacks.

Clearly, the argument domain is not the same between the two frameworks – AFs now have to include conflicts as arguments. Nevertheless, the semantics domain is the same. The translation introduces new arguments that represent the attacks and as every conflict receives such representation, the approach is not relation removing. It is however induced attack relation introducing, as the previously indirect defeats between conflicts in AFRAs become direct attacks between their AF representations. From the redefinition of Theorem 7.1 we can see that Tr_{AF}^{AFRA} is generic and exact. However, please note that this result is not entirely uncontroversial. After all, we need to be able to "read back" some arguments as conflicts, which is something we wanted to avoid in exact approaches. Consequently, even though for now we classify the AFRA–AF translation as an exact one, it is a property open for discussion.

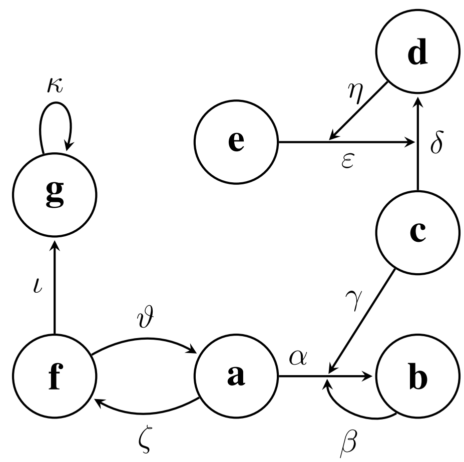
The translation is quite simple and the only reason we are not classifying it as semi–structural is the fact that it is not just direct defeats, but also indirect ones, that are taken into account in the construction of the new conflict set. Indirect defeats between two conflicts are as such not represented in the structure of AFRA and are a semantical notion, even though a minor one. Nevertheless, without being aware of them and using only the direct defeats we would not have a translation preserving the behavior of the semantics. This is also the reason why this approach exhibits the behavior of attack propagation translations and thus is classified as a hybrid.

It is also the indirect conflicts that cause the loss of modularity in Tr_{AF}^{AFRA}. Let us consider an AFRA $FR_1 = (\{a, b, c\}, \{(a, b), (b, c)\})$ and two of its subframeworks $FR_2 = (\{a, b, c\}, \{(a, b)\})$ and $FR_3 = (\{a, b, c\}, \{(b, c)\})$. Their associated AFs are $F_1 = (\{a, b, c, (a, b), (b, c)\}, \{((a, b), b), ((b, c), c), ((a, b), (b, c))\})$, $F_2 = (\{a, b, c, (a, b)\}, \{((a, b), b)\})$ and $F_3 = (\{a, b, c, (b, c)\}, \{((b, c), c)\})$ respectively. We can observe that the union of F_2 and F_3 is different from F_1 – the $((a, b), (b, c))$ conflict is not present. Consequently, Tr_{AF}^{AFRA} is not modular. ■

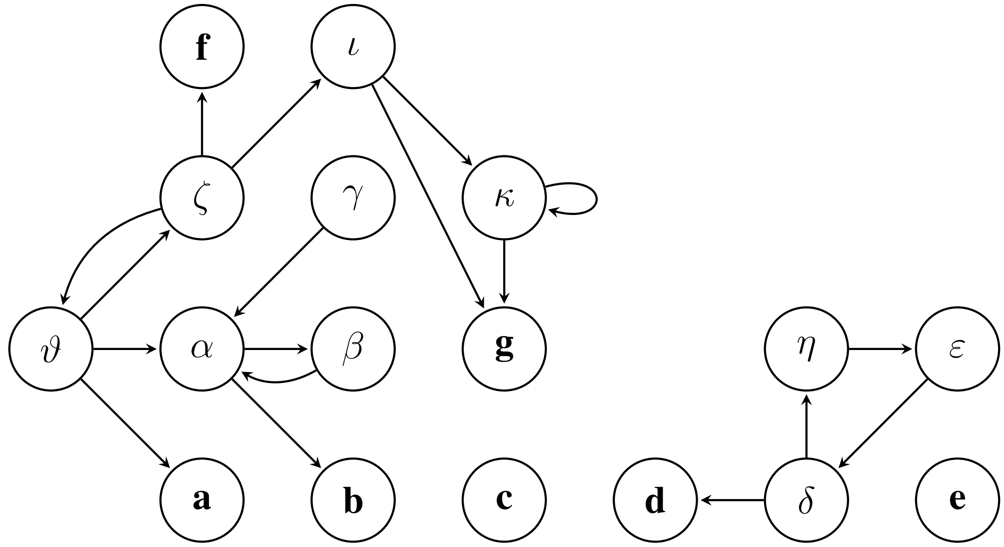
Example 87. Let us consider the AFRA $FR = (A, R)$ from Example 4, where $A = \{a, b, c, d, e, f, g\}$ and $R = \{\alpha, \beta, \gamma, \delta, \varepsilon, \eta, \zeta, \vartheta, \iota, \kappa\}$, with $\alpha = (a, b)$, $\beta = (b, \alpha)$, $\gamma = (c, \alpha)$, $\delta = (c, d)$, $\varepsilon = (e, \delta)$, $\eta = (d, \varepsilon)$, $\zeta = (a, f)$, $\vartheta = (f, a)$, $\iota = (f, g)$ and $\kappa = (g, g)$. We depict it again in Figure 60a for readers' convenience. The direct defeats in FR are $D = \{(\alpha, b), (\beta, \alpha), (\gamma, \alpha), (\delta, d), (\varepsilon, \delta), (\eta, \varepsilon), (\zeta, f), (\vartheta, a), (\iota, g), (\kappa, g)\}$. The indirect defeats are $I = \{(\vartheta, \alpha), (\vartheta, \zeta), (\alpha, \beta), (\delta, \eta), (\zeta, \vartheta), (\zeta, \iota), (\iota, \kappa), (\kappa, \kappa)\}$. According to Translation 32, the associated AF is therefore $F^{FR} = (A \cup R, D \cup I)$. We can see it depicted in Figure 60b. Due to the huge amount of possible admissible extensions, we will focus on more advanced semantics. We can observe that c, e and γ are the only unattacked arguments in this framework. γ defends β and b. This gives us our grounded extension $\{b, c, e, \beta, \gamma\}$, which is the same as in FR. The η, ε and δ conflict arguments form an odd length cycle. Neither them nor d will appear in any of extension, and thus we can ignore this part of F^{FR}. We can now select ϑ, which leads to the acceptance of ι and f, thus producing another complete extension $\{b, c, e, f, \beta, \gamma, \vartheta, \iota\}$. Alternatively, we can accept ζ and a and obtain the set $\{a, b, c, e, \beta, \gamma, \zeta\}$. This gives us the three complete extensions of F^{FR}. We can observe that the latter two are preferred. Due to the aforementioned cycle, no subset of $A \cup R$ is stable. We therefore retrieve all and only extensions produced by FR.

7.1.2 Defender Translation

Although the standard AFRA–AF translation is already exact, it is not the only approach available in the literature. In what follows we would like to recall the metalevel translations that, although mostly used in the context of frameworks that have second order attacks (i.e. single recursion), can be extended to AFRAs. In the first approach [18], the target arguments make statements about the source arguments and conflicts; this includes



(a) Sample AFRA FR



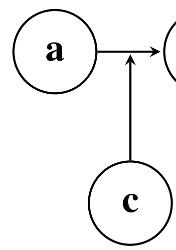
(b) AF representing FR

Figure 60: AFRA FR and its associated AF

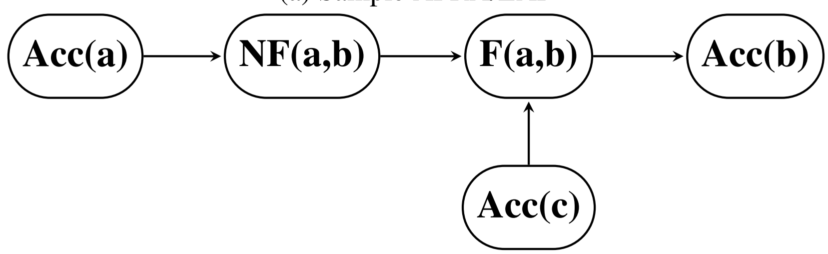
whether an argument can be "accepted" **Acc** and if a given attack is "in force" **F** or "not in force" **NF**, as seen in Figure 61[20].

However, this approach is not entirely sufficient for AFRAs. The "deepest" attack is not made into an argument, which means it will not show up in the extension of the target AF. Since it can appear in an extension of the source AFRA, we have a semantical mismatch. In the other approach, the meta–arguments make statements whether an argument is "justified" **j**, "rejected" **r**, or "defeats" **def** either another argument or conflict [64], as seen in Figure 62.

[20]Please note that the presented frameworks correspond to the definitions given in [18], not to the provided examples, which are closer to the method later introduced in [64].



(a) Sample AFRA/EAF



(b) Corresponding meta–level AF

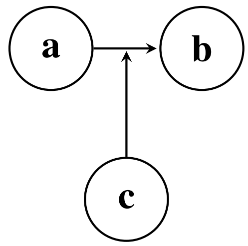
Figure 61: Sample AFRA/EAF and its meta–level AF from [18]

We can observe that all conflicts in the framework receive their respective arguments in this approach. However, the translation was analyzed only in the context of EAFs, in which recursion is much more limited than in AFRAs. The general approach was only sketched out and thus we will now prove that it is indeed correct. However, please note that just like in the SETAF–AF case (see Section 6.1.2), we will not recall the structure of the logic–based meta–level AFs used in [64] and use a simpler framework. The $j(x)$ and $def(x, y)$ arguments will revert to the arguments and conflicts that they represent and we will use x' to stand for $r(x)$. For further meta–level analysis we refer the reader to the original paper. Our interest is in adjusting the approach in a way we can create a faithful translation.

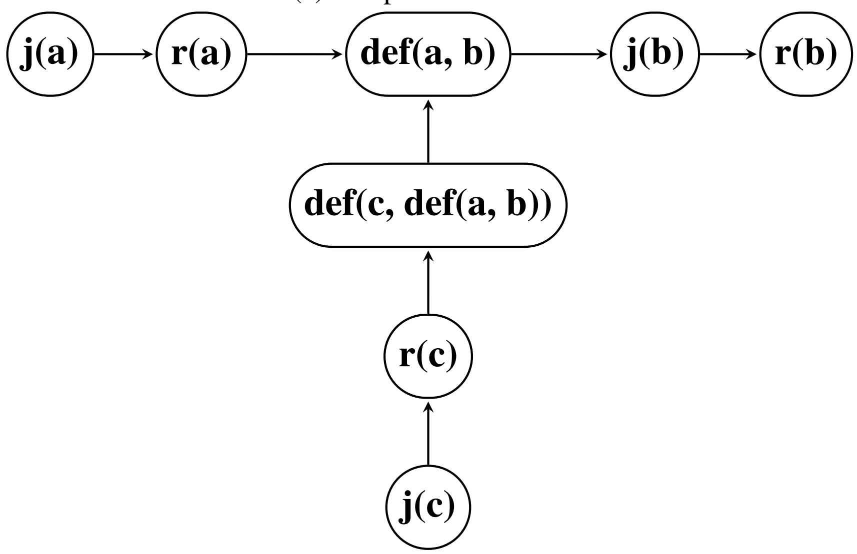
The transformation is now as follows; arguments, their primed versions and conflicts become the target arguments. We now have three types of conflicts in the target AF. The first type consists of attacks by arguments on their primed versions. In the second type, if a given argument is a source of an AFRA attack, then the primed argument corresponding to it attacks this conflict. Finally, we connect the conflict arguments with their targets. This brings us to the following formulation:

Translation 33. *Let $FR = (A, R)$ be an AFRA. The corresponding AF is $F^{FR}_m = (A', R')$ for $A' = A \cup R \cup X'$, where $X' = \{x' \mid x \in A\}$, and $R' = \{(x, x') \mid x \in A\} \cup \{(x', a) \mid a \in R, src(a) = x\} \cup \{(a, b) \mid a \in R, b \in A \cup R, trg(a) = b\}$.*

Please note that unlike in the standard translation, not all of the semantics are preserved by the meta–level one. This comes from the fact that in an AFRA admissible extensions, an attack does not need to be accompanied by its source. However, since in the meta–level framework the attack source becomes the only defender against the Rej arguments, its



(a) Sample AFRA/EAF



(b) Corresponding meta–level AF

Figure 62: Sample AFRA/EAF and its meta–level AF from [64]

presence is forced. Similar issues arise e.g. in the AFRA–AFN translation (see Section 7.4). Moreover, unlike in the previous approach, the indirect defeats do not become direct conflicts. This leads to the fact that also the conflict–free semantics is preserved only "one way".

Theorem 7.2. *Let $FR = (A, R)$ be an AFRA and $F_m^{FR} = (A', R')$ its corresponding AF obtained through Translation 33. If $E \subseteq A \cup R$ is a σ–extension of FR, where $\sigma \in$ $\{$conflict–free, complete, preferred, grounded, stable$\}$, then $E' = E \cup \{x' \mid x \in (A \cap E^+)\}$ is a σ–extension of F_m^{FR}, where $E^+ = \{x \mid \exists y \in E$ s.t. y defeats $x\}$ is the discarded set of E in FR. This does not necessarily hold for admissible semantics. If $E' \subseteq A'$ is a σ'–extension of F_m^{FR}, where $\sigma' \in \{$admissible, complete, preferred, grounded, stable$\}$, then $E = E' \cap (A \cup R)$ is a σ'–extension of FR. This does not necessarily hold for conflict–free semantics.*

The redefinitions of the available results are now the following:

Redefinition of Translation 33: *Let Fr^{AFRA} be the collection of all AFRAs on the domain $\mathcal{U}$ and Fr^{AF} the collection of all AFs on the domain $\mathcal{U}' \cup \bigcup_{i=1}^{\infty} \mathcal{U}^i$, where $\mathcal{U}^1 = \mathcal{U}$*

and $\mathcal{U}^i = \mathcal{U} \times \mathcal{U}^{i-1}$ *for* $i > 1$. *The translation* $m\text{-}Tr_{AF}^{AFRA} : Fr^{AFRA} \to Fr^{AF}$ *is defined as* $m\text{-}Tr_{AF}^{AFRA}((A,R)) = (A', R')$, *where* $A' = A \cup R \cup X'$ *for* $X' = \{x' \mid x \in A\}$ *and* $R' = \{(x,x') \mid x \in A\} \cup \{(x',a) \mid a \in R, src(a) = x\} \cup \{(a,b) \mid a \in R, b \in A \cup R, trg(a) = b\}$.

Redefinition of Theorem 7.2: *Let* $\sigma \in \{$*complete, preferred, grounded, stable*$\}$ *be a semantics and* SC_σ^{Tr} *the removal casting functions for* σ *defined as* $E_\sigma^X(E) = E \cap (A \cup R)$ *for a framework* $X = (A,R) \in Fr^{AFRA}$ *and* $E \in \sigma(m\text{-}Tr_{AF}^{AFRA}(X))$. *The translation* $m\text{-}Tr_{AF}^{AFRA}$ *is strong and semantics bijective under* (σ, SC_σ^{Tr}). *It is* $\subseteq$*–weak under the conflict–free semantics and* $\supseteq$*–weak under the admissible semantics and the defined semantics casting functions.*

Analysis of Translation 33: Under the conflict–free, admissible, complete, preferred, grounded and stable semantics and the defined removal casting functions, the translation $m\text{-}Tr_{AF}^{AFRA}$ is:

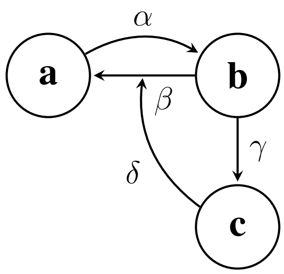
- full, target–subclass and injective
- argument domain altering, argument introducing and induced attack relation introducing
- generic and semantics domain altering
- semi–structural and modular

Translation $m\text{-}Tr_{AF}^{AFRA}$ is faithful under the complete, preferred, grounded and stable semantics and the defined removal casting functions. It is classified as basic–defender under the listed semantics and casting functions.

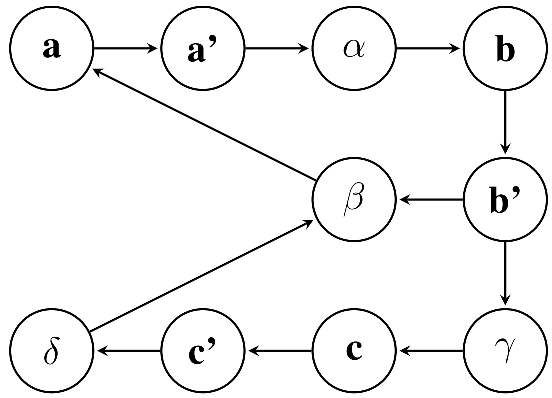
Explanation. Since every AFRA can be translated into an AF, the translation is full. However, even though we do not provide a full description of how the target AFs can look like, there are some frameworks that cannot be produced. For example, the translation $m\text{-}Tr_{AF}^{AFRA}$ cannot create a self–attacker. Moreover, every AFRA uniquely defines the set of arguments in the obtained Dung's framework and the translation has to be injective.

We can observe that neither the argument nor semantics domain is preserved in this translation. Since every conflict is represented by an argument in the target structure, our approach is not relation removing. It is induced attack relation introducing due to the conflicts related to the primed arguments. Because of the amount of the semantics handled in a strong manner, $m\text{-}Tr_{AF}^{AFRA}$ is generic. Moreover, as it is semantics bijective for complete, preferred, grounded and stable semantics and removal casting functions are used (Theorem 7.2), we classify it as faithful.

This translation exploits the concept of defense in order to tie the conflicts to their sources. Consequently, we qualify it as semi–structural. Fortunately, this approach is modular. We can observe there is no attack propagation as in the standard AFRA–AF translation; a given conflict is connected via defense precisely to its source and attacks only its direct target. Therefore, there is no argument or attack in a translation of a union of two frameworks that would not be present in the union of their translations, and no elements are lost in the process. ■



(a) Sample AFRA



(b) The associated AF

Figure 63: Meta–level translation for AFRAs

Example 88. Let us look at the AFRA $FR = (\{a, b, c\}, \{\alpha, \beta, \gamma, \delta\})$ depicted in Figure 63a, where $\alpha = (a, b)$, $\beta = (b, a)$, $\gamma = (b, c)$ and $\delta = (c, \beta)$. According to Translation 33, its associated AF is $F^{FR} = (A', R')$, where $A' = \{a, b, c, a', b', c', \alpha, \beta, \gamma, \delta\}$ and $R' = \{(a, a'), (b, b'), (c, c'), (a', \alpha), (b', \beta), (b', \gamma), (c', \delta), (\alpha, b), (\beta, a), (\gamma, c), (\delta, \beta)\}$. We can see it in Figure 63b. The admissible extensions of FR are $\emptyset$, $\{\alpha\}$, $\{a, \alpha\}$, $\{c, \alpha\}$, $\{a, c, \alpha\}$, $\{\alpha, \delta\}$, $\{a, \alpha, \delta\}$, $\{c, \alpha, \delta\}$, $\{a, c, \alpha, \delta\}$, $\{\beta, \gamma\}$ and $\{b, \beta, \gamma\}$. Out of these sets, $\emptyset$, $\{a, c, \alpha, \delta\}$ and $\{b, \beta, \gamma\}$ are complete. The first extension is the grounded extension, while the other two are preferred and stable.

Let us now focus on F^{FR}. Its admissible sets are $\emptyset$, $\{a, b', \alpha\}$, $\{a, c, b', \alpha\}$, $\{a, c, b', \alpha, \delta\}$ and $\{b, a', c', \beta, \gamma\}$. They correspond to extensions $\emptyset$, $\{a, \alpha\}$, $\{a, c, \alpha\}$, $\{a, c, \alpha, \delta\}$ and $\{b, \beta, \gamma\}$. We can observe that not all of the admissible extensions of FR are retrieved. Due to the fact that defense enforces the presence of the source of an attack in an admissible set, the answers $\{\alpha\}$, $\{c, \alpha\}$, $\{\alpha, \delta\}$, $\{a, \alpha, \delta\}$, $\{c, \alpha, \delta\}$ and $\{\beta, \gamma\}$ are lost. However, the complete extensions of F^{FR} are $\emptyset$, $\{a, c, b', \alpha, \delta\}$ and $\{b, a', c', \beta, \gamma\}$, which corresponds to the sets produced by FR. It is easy to verify that the grounded and preferred extensions between FR and F^{FR} are also related in the desired manner.

7.1.3 Improvements

The Translation 32 is already exact; consequently, the only improvements could concern the argument domain and modularity. Unfortunately, it does not appear that much can be done in this area without the loss of exactness. Even if we used attack semantics for AFs [88] in order to address the issue of domain alteration, it would not solve the issue of recursive attacks. Thus, argument domain change is inevitable. Obtaining modularity would require preventing indirect defeats showing up as attacks in the target AF. However, without them the Translation 32 would no longer preserve the behavior of the semantics. A possible solution is to connect the attacks to their sources (e.g. via defender approach), however, as we have observed in Translation 33, it leads to loss of exactness since the auxiliary primed arguments appear in the extensions. Turning them into self–attackers would prevent them from showing up, but it easy to see that the semantics would no longer be preserved – we would lose the ability to defend from an attack with an indirect defeat. Therefore, we decide not to pursue any further improvements to the AFRA–AF translations.

7.2 AFRA as EAF

Out of all of the frameworks we have looked at in this work, only AFRAs and EAFs allow attacks on attacks, even though in different extent. However, as already noted in [9], the two frameworks are quite different from the semantics perspective. Additionally, in [8], the EAF$^+$ framework was proposed, which added recursion to defense attacks. Nevertheless, the provided results focused on the EAF–AFRA direction, not the other way around. We will now try to show some of the issues in the AFRA–EAF approach.

For the sake of this analysis, we will limit ourselves to the case where only attacks directed at arguments can be attacked. Moreover, we will use frameworks on which both EAF and EAFC semantics would coincide, i.e. we do not permit symmetric attacks between arguments in AFRA. Finally, we will focus our analysis on the semantics that are at least complete. AFRAs produce a lot of admissible extensions, which makes their analysis quite difficult – we could observe that the framework in Example 4 had over two hundred admissible extensions, while only three complete ones. Moreover, EAF extensions contain only arguments, while AFRA semantics produce collections of both arguments and attacks. Therefore, we would like to exploit the fact that defending an attack implies defending its source argument (see Lemma 2.32) and and thus in case of complete semantics we can at least partially disregard the conflicts appearing in extensions in our comparison.

Example 89. Adapted from [62]. Let us consider the AFRA $FR = (\{a, b, c, d, e, f, g\}, \{\alpha = (a, b), \beta = (d, c), \gamma = (b, (d, c)), \delta = (c, (a, b)), \epsilon = (b, e), \zeta = (e, f), \eta = (f, g)\})$ from Figure 64. Please recall that any set of arguments can be considered conflict–free; only the inclusion of attacks into the extension can change the situation. The framework has 108 admissible extensions and we will not list them here. The complete ones are $\{a, d\}$, $\{a, d, e, g, \alpha, \beta, \zeta\}$ and $\{a, b, c, d, f, \gamma, \delta, \epsilon, \eta\}$. The latter two are preferred and stable, while the first one is grounded.

Let us now consider the EAF obtained from our AFRA, where the set of arguments remains the same and the conflicts are split into argument–argument and argument–attack ones:

Example 90. [62] Let us consider the EAF $EF = (\{a, b, c, d, e, f, g\}$, $\{(a, b)$, (d, c), (b, e), (e, f), $(f, g)\}$, $\{(b, (d, c)), (c, (a, b))\})$ from Figure 65. It has a number of conflict–free extensions and thus we will focus only on some of them. We can observe that the sets

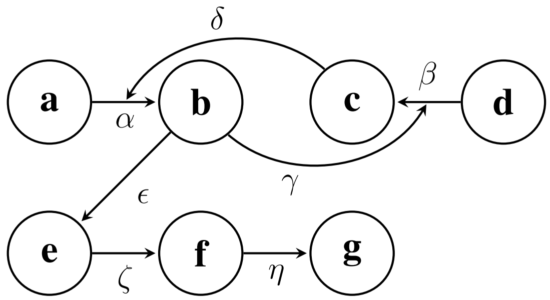


Figure 64: Framework FR

$\{a, b\}$ and $\{c, d\}$ are not conflict–free. However, $\{a, b, c\}$, $\{b, c, d\}$ and $\{a, b, c, d\}$ already are. This is due to the fact that attacks are no longer defeats when the defense attacks are present. Additionally, also $\{a, d, e, g\}$ and $\{b, c, a, d, f\}$ are conflict–free. The admissible extensions of EF include $\emptyset$, $\{a\}$, $\{d\}$, $\{a, d\}$, $\{b, c\}$, $\{a, b, c\}$, $\{b, c, d\}$, $\{a, d, e\}$, $\{b, c, f\}$, $\{a, b, c, f\}$, $\{b, c, d, f\}$, $\{a, d, e, g\}$, $\{a, b, c, d\}$ and $\{a, b, c, d, f\}$. We can observe that the set $E = \{b, c\}$ is admissible. Neither a nor d defeat$_E$ any of its elements, and thus there is nothing to defend from. The set $\{a, d, e\}$ is admissible since the defeat of b by a has a reinstatement set $\{(d, c), (a, b)\}$ – the defense attack on (a, b) by c is reinstated with (d, c), and (d, c), that is defense attacked by b, is reinstated with (a, b). Although the behavior appears cyclic, it suffices for defense.

Out of the listed admissible extensions, $\{a, d, e, g\}$ and $\{a, b, c, d, f\}$ are complete. We can observe they are incomparable and do not follow the typical semilattice structure of complete extensions – this is another consequence of the fact that the characteristic operator of EAFs (and EAFCs) is not monotonic. The grounded extension is $\{a, d, e, g\}$; it is minimal, but not the least complete extensions. Both $\{a, d, e, g\}$ and $\{a, d, b, c, d, f\}$ are stable and preferred.

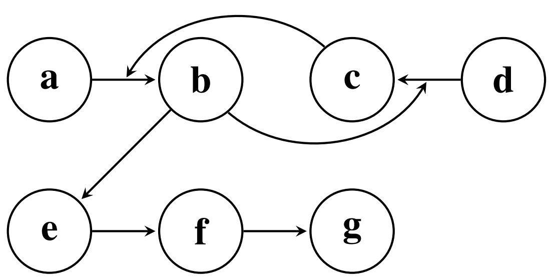


Figure 65: Framework EF associated with FR

We can now compare our two frameworks FR and EF. We can observe that EF does not recreate all of the complete extensions of FR – namely, the extension corresponding to $\{a, d\}$ is missing. This means that that also the grounded extensions between the frameworks are different, even though the preferred and the stable are related. The aforementioned issues are caused by different definitions of acceptability in both frameworks and different treatment of attacks. In particular, in the described example we can observe that the characteristic operator of FR was monotonic, while the one in the associated EF was not. Consequently, we can also expect the relation between the admissible extensions to be somewhat complicated.

Although the stable and preferred semantics were in some correspondence between the source and target frameworks in our example, we have only focused on the single–recursion AFRAs without symmetric attacks. Consequently, creating a dedicated AFRA–EAF translation is not beneficial, particularly taking into account the benefits of the AFRA–AF–EAF chain. Not only all of the desired semantics are preserved and every AFRA can be translated, but the produced EAF belongs to various normal forms and subclasses that can make computation easier and preserve monotonicity of the characteristic operator. Therefore, we have decided not to continue the AFRA–EAF analysis and use the chained approach from now on.

7.3 AFRA as BAF

We have already seen that the translation from AFRAs to AFs is fully "working". However, it is based on defeats, not the attacks in R, and suffers from the loss of modularity. In a certain sense, if we see the relation between an attack and its source as support, direct defeats would correspond to direct attacks and indirect defeats to secondary attacks. However, in AFRA, a conflict can appear without its source in an admissible extension (complete ones always include the sources due to Lemma 2.32). This brings us to the abstract interpretation of support, developed in the early BAFs.

Translation 34. *Let $FR = (A, R)$ be an AFRA. Its corresponding BAF $BF^{FR} = (A', R', S')$ is defined in the following way:*

- $A' = A \cup R$,
- $R' = \{(X, Y) \mid X \in R, Y \in A \cup R, Y = trg(X)\}$, *and*
- $S' = \{(X, Y) \mid Y \in R, X \in A, X = src(Y)\}$.

Let us now focus on the semantics. The proof that the extensions of FR and BF^{FR} are in a close relation is quite straightforward. This is due to the fact that the used BAF semantics do not have any explicit requirements concerning the support relation, unlike e.g. in AFNs and EASs. Everything is handled with the indirect attacks, and as direct and secondary conflicts encompass both types of defeats in AFRAs, the rest follows easily:

Theorem 7.3. *Let $FR = (A, R)$ be an AFRA and $BF^{FR} = (A', R', S)$ its corresponding BAF obtained through Translation 34. Let R^{sec} be the collection of first–tier secondary attacks in BF^{FR}. $E \subseteq A \cup R$ is a conflict–free (stable, d–grounded) extension of FR iff it is +conflict–free (stable, d–grounded) in BF^{FR} w.r.t. R^{sec}. E is a σ–extension of FR, where $\sigma \in \{$admissible, complete, preferred$\}$, iff it is a d–σ–extension of FR w.r.t. (R^{sec}, R^{sec}).*

We can now put this translation into our system. Although we have not introduced too many BAF subclasses in our work, mostly due to the fact that both attacks and support are binary and the semantics do not deal with support cycles, it is easy to see that not all BAFs can be produced by our translation. The support graph can be seen as a forest, where each tree consists precisely of one edge. Clearly, this cannot account for all the possible support relations that can appear in BAFs.

Redefinition of Translation 34: *Let Fr^{AFRA} be the collection of all AFRAs on domain $\mathcal{U}$ and Fr^{BAF} be the collection of all BAFs on the domain $\bigcup_{i=1}^{\infty} \mathcal{U}^i$, where $\mathcal{U}^1 = \mathcal{U}$ and $\mathcal{U}^i = \mathcal{U} \times \mathcal{U}^{i-1}$ for $i > 1$. The translation $Tr_{BAF}^{AFRA} : Fr^{AFRA} \rightarrow Fr^{BAF}$ is defined as $Tr_{BAF}^{AFRA}((A, R)) = (A', R', S)$, where $A' = A \cup R$, $R' = \{(X, Y) \mid X \in R, Y \in A \cup R, Y = trg(X)\}$ and $S' = \{(X, Y) \mid Y \in R, X \in A, X = src(Y)\}$.*

Redefinition of Theorem 7.3: *Let $\sigma^{AFRA} \in$ {conflict–free, admissible, complete, preferred, grounded, stable} be an AFRA semantics and $\sigma^{BAF} \in$ {+conflict–free, d–admissible, d–complete, d–preferred, d–grounded, stable} a similar BAF semantics. Let SC_σ^{Tr} be the identity casting functions for σ. Translation Tr_{BAF}^{AFRA} is strong and semantics bijective under σ and SC_σ^{Tr}.*

Analysis of Translation 34: Under the (+) conflict–free, (d–) admissible, (d–) complete, (d–) preferred, (d–) grounded and stable semantics and identity casting functions, the translation Tr_{BAF}^{AFRA} is:

- full, target–subclass and injective
- argument domain altering, argument introducing, attack relation preserving, induced support relation introducing
- generic, semantics domain preserving and exact
- semi–structural and modular

Translation Tr_{BAF}^{AFRA} is classified as basic under the listed semantics and casting functions.

Explanation. Since any AFRA can be translated into a BAF, the translation is full. Based on the previous discussion, it is also target–subclass. The translation is easily injective for the same reasons the AFRA–AF Translation 32 was – the way the set of arguments is defined in the target BAF clearly describes the source framework. Similarly, the current approach is also argument domain altering, argument introducing, generic and semantics domain preserving. However, this time it is attack relation preserving – although the conflicts are now represented with arguments, only the direct defeats are taken into account. Clearly, the approach is also induced support introducing.

We choose to classify our translation as semi–structural. By adding support, we simulate the indirect defeats in AFRA with indirect conflicts in BAF. Consequently, we are using some semantical notions of BAFs, even though very basic ones. In contrast, a purely structural translation (i.e. producing $(A', R', \emptyset)$ instead of (A', R', S)) would not preserve the extensions between the source and target frameworks in a strong manner.

The fact that the translation is exact follows straightforwardly from Theorem 7.3 and its redefinition. Moreover, the approach can be easily shown to be modular. Let $FR_1 = (A_1, R_1)$ and $FR_2 = (A_2, R_2)$ be two frameworks. Let $FR_3 = FR_1 \cup FR_2 = (A_1 \cup A_2, R_1 \cup R_2)$. Let us now consider the frameworks $Tr^{AFRA}_{BAF}(FR_1) \cup Tr^{AFRA}_{BAF}(FR_2)$ and $Tr^{AFRA}_{BAF}(FR_3)$; we can observe that the set of arguments is identical in both cases and equal to $A_1 \cup R_1 \cup A_2 \cup R_2$. Concerning the attacks, we receive $R'_1 = \{(X, Y) \mid X \in R_1, Y \in A_1 \cup R_1, Y = trg(X)\} \cup \{(X, Y) \mid X \in R_2, Y \in A_2 \cup R_2, Y = trg(X)\}$ for the first structure and $R'_2 = \{(X, Y) \mid X \in (R_1 \cup R_2), Y \in (A_1 \cup R_1 \cup A_2 \cup R_2), Y = trg(X)\}$ for the other. Clearly, $R'_1 \subseteq R'_2$. Assume there is some pair (X, Y) in R'_2, not in R'_1. It can only be the case that $X \in R_1$ and $Y \in (A_2 \cup R_2)$, or $X \in R_2$ and $Y \in (A_1 \cup R_1)$. However, due to the restriction that $Y = trg(X)$, it has to be the case that if $Y \in (A_2 \cup R_2)$, then $Y \in (A_1 \cup R_1)$ and vice versa – the attack X can after all be targeted at an element that appears in its own framework as well. Consequently, if $(X, Y) \in R'_2$, then $(X, Y) \in R'_1$ and the two sets are equal. The support analysis is quite straightforward and similar to this one and we can thus conclude that the translation Tr^{AFRA}_{BAF} is modular. ∎

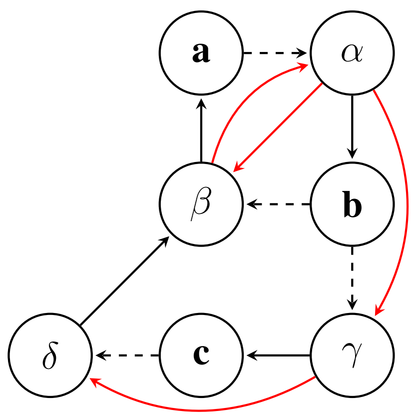


Figure 66: An AFRA–produced BAF

Example 91. Let us come back to the framework from Example 88 and Figure 63a, i.e. $FR = (\{a, b, c\}, \{\alpha, \beta, \gamma, \delta\})$, where $\alpha = (a, b)$, $\beta = (b, a)$, $\gamma = (b, c)$ and $\delta = (c, \beta)$. The BAF associated with FR is $BF^{FR} = (A', R', S')$, where $A' = \{a, b, c, \alpha, \beta, \gamma, \delta\}$, $R' = \{(\alpha, b), (\beta, a), (\gamma, c), (\delta, \beta)\}$ and $S' = \{(a, \alpha), (b, \beta), (b, \gamma), (c, \delta)\}$. The secondary attacks in BF^{FR} are $R^{sec} = \{(\gamma, \delta), (\alpha, \gamma), (\alpha, \beta), (\beta, \alpha)\}$. We can see they correspond to the indirect defeats in FR. BF^{FR}, along with the secondary attacks marked in red, can be seen in Figure 66. From now on we will assume that the BAF semantics are parametrized with secondary attacks. The d–admissible sets of BF^{FR} are therefore $\emptyset$, $\{\alpha\}$, $\{a, \alpha\}$, $\{c, \alpha\}$, $\{a, c, \alpha\}$, $\{\alpha, \delta\}$, $\{a, \alpha, \delta\}$, $\{c, \alpha, \delta\}$, $\{a, c, \alpha, \delta\}$, $\{\beta, \gamma\}$ and $\{b, \beta, \gamma\}$, which is in accordance with the admissible extensions of FR. Due to the fact that every argument in A' is (directly or secondary) attacked, it is easy to see that $\emptyset$ is a d–complete extension of BF^{FR}. $\{\alpha\}$ is unfortunately not d–complete; it defends c and δ from γ and a

from β. Consequently, only $\{a, c, \alpha, \delta\}$ becomes d–complete. Similarly, we can observe that $\{\beta, \gamma\}$ defends b and as such does not qualify as a d–complete extension. However, $\{b, \beta, \gamma\}$ meets all the requirements. We thus obtain the three desired complete extensions. We can observe that all of them are also inverse closed. Our d–grounded extension is $\emptyset$, while $\{a, c, \alpha, \delta\}$ and $\{b, \beta, \gamma\}$ are d–preferred. These two sets are also stable in BF^{FR}; all of the elements of A' not contained in them are either directly or secondary attacked. We can conclude that the extensions produced by BF^{FR} are exactly the same as the ones created with FR.

7.4 AFRA as AFN

In the previous sections we have analyzed the AFRA–AF and AFRA–BAF translations. We have observed that abstract support, joined with secondary attack, is sufficient for an exact and modular transformation from AFRAs to BAFs. However, this type of support is unique to BAFs only, while the necessary one that appears in AFNs is connected to deductive and evidential supports (see Sections 9.2 and 10.5). Moreover, it resembles the positive dependencies in ADFs more closely than abstract support. Consequently, we would like to see whether AFRAs can be conveniently expressed with AFNs.

Translation 35. *Let $FR = (A, R)$ be an AFRA. Its corresponding AFN $FN^{FR} = (A', R', N')$ is defined in the following way:*

- $A' = A \cup R$,
- $R' = \{(X, Y) \mid X \in R, Y \in A \cup R, Y = trg(X)\}$, *and*
- $N' = \{(\{X\}, Y) \mid Y \in R, X \in A, X = src(Y)\}$.

The produced AFNs are not very complicated. First of all, they are support binary and singular, which also makes them minimal by Lemma 4.71. Moreover, they are of support depth 1. Since only the attack arguments can attack and only the standard ones can support, the target frameworks are also (strongly) consistent. Finally, the binary version of the support graph (A, N) is (directed) acyclic, which by Theorem 4.34 means that the produced AFNs are strongly valid. Consequently, our frameworks belong to the subclass of well–structured and elementary AFNs with support depth 1. In other words, all of the normal forms are satisfied:

Theorem 7.4. *Let $FR = (A, R)$ be an AFRA and $FN^{FR} = (A', R', N')$ its corresponding AFN obtained through Translation 35. FN^{FR} is minimal, (strongly) consistent, weakly, strongly and relation valid.*

However, please notice that $WSt^{AFN} \cap SEle_1^{AFN}$ is not the most accurate description of FN^{FR}. The produced frameworks are further restricted with requiring that every argument receiving support carries out an attack and every argument carrying out an attack requires support. Furthermore, every argument carries out at most one attack – we do not have any conflict propagation that was present in the AFRA–AF approach. Nevertheless, the current description is sufficient for our purposes.

We can observe that, structurally, Translation 35 to AFNs is similar to Translation 34 to BAFs. However, different interpretation of support means that the extensions produced by both frameworks will be different. Necessary support leads to the fact that an argument cannot be accepted without its supporters (if they exist). Consequently, unlike in the AFRA–BAF approach, the admissible semantics will not be strongly preserved. However, using only the extensions obtained from the corresponding AFN could be used to strengthen the AFRA semantics and retain only these sets in which attacks are not "detached" from their origins, which was an issue in the AF–AFRA translation (see Section 5.2.1 for a discussion).

Definition 7.5. Let $FR = (A, R)$ be an AFRA and $E \subseteq A \cup R$ be a set its elements. The **source–complete set** of E is $E^{src} = E \cup \{src(V) \mid V \in E \cap R\}$.

Please note that if E^{src} is conflict–free and admissible, then so is E by Lemma 2.32. By the same lemma, it also holds that if E is a complete extensions, then $E = E^{src}$.

Theorem 7.6. *Let $FR = (A, R)$ be an AFRA and $FN^{FR} = (A', R', N')$ its corresponding AFN obtained through Translation 35. If a set $E \subseteq A \cup R$ is a σ–extension of FR, where $\sigma \in \{$conflict–free, complete, preferred, grounded, stable$\}$, then it is a σ–extension of FN^{FR}. If $E = E^{src}$ is admissible in FR, then it is admissible in FN^{FR} and if it is conflict–free in FR, it is strongly coherent in FN^{FR}. It might not be the case for $E \neq E^{src}$.*

Not every conflict–free extension of FN^{FR} is conflict–free in FR. If a set $E' \subseteq A'$ is strongly coherent in FN^{FR}, then it is conflict–free in FR. If a set $E' \subseteq A'$ is a σ'–extension of FN^{FR}, where $\sigma' \in \{$admissible, complete, preferred, grounded, stable$\}$, then E' is a σ'–extension of FR.

We can thus conclude that while complete, preferred, grounded and stable extensions coincide between the two frameworks, the target AFNs produce "too many" conflict–free ones and "not enough" admissible ones. We can now reformulate and analyze our translation:

Redefinition of Translation 35: *Let Fr^{AFRA} be the collection of all AFRAs on domain $\mathcal{U}$ and $WSt^{AFN} \cap SEle_1^{AFN}$ be the collection of well–structured and elementary AFNs with support depth 1 on the domain $\bigcup_{i=1}^{\infty} \mathcal{U}^i$, where $\mathcal{U}^1 = \mathcal{U}$ and $\mathcal{U}^i = \mathcal{U} \times \mathcal{U}^{i-1}$ for $i > 1$. The translation $Tr_{AFN}^{AFRA} : Fr^{AFRA} \rightarrow (WSt^{AFN} \cap SEle_1^{AFN})$ is defined as $Tr_{AFN}^{AFRA}((A, R)) = (A', R', N')$, where $A' = A \cup R$, $R' = \{(X, Y) \mid X \in R, Y \in A \cup R, Y = trg(X)\}$ and $N' = \{(\{X\}, Y) \mid Y \in R, X \in A, X = src(Y)\}$.*

Redefinition of Theorem 7.6: *Let $\sigma \in \{complete, preferred, grounded, stable\}$ be a semantics and SC_σ^{Tr} the identity casting functions for σ. Translation Tr_{AFN}^{AFRA} is strong and semantics bijective under σ and SC_σ^{Tr}. It is $\subseteq$–weak under conflict–free semantics and identity casting functions and $\supseteq$–weak under admissible semantics and identity casting functions.*

Please note the same explanations hold as in the AFRA–BAF translation. Consequently, we will omit further discussion.

Analysis of Translation 35: Under the conflict–free, admissible, complete, preferred, grounded and stable semantics and identity casting functions, the translation Tr_{AFN}^{AFRA} is:

- full, target–subclass and injective
- argument domain altering, argument introducing, attack relation preserving, induced support relation introducing
- generic and semantics domain preserving
- semi–structural and modular

Under the complete, preferred, grounded and stable semantics and identity casting functions, Tr_{AFN}^{AFRA} is exact. It is classified as basic under the listed semantics and casting functions.

Example 92. Let us come back to the framework from Example 88 and Figure 63a, i.e. $FR = (\{a, b, c\}, \{\alpha, \beta, \gamma, \delta\})$, where $\alpha = (a, b)$, $\beta = (b, a)$, $\gamma = (b, c)$ and $\delta = (c, \beta)$. The AFN associated with FR is $FN^{FR} = (A', R', N')$, where $A' = \{a, b, c, \alpha, \beta, \gamma, \delta\}$, $R' = \{(\alpha, b),\ (\beta, a),\ (\gamma, c),\ (\delta, \beta)\}$ and $N' = \{(\{a\}, \alpha), (\{b\}, \beta), (\{b\}, \gamma), (\{c\}, \delta)\}$. Structurally speaking, the framework looks the same as the BAF in Figure 66 without the secondary attacks that were marked in red. For every argument in A', we can create a single minimal coherent set containing it; for every argument $x \in A$, this set is simply $\{x\}$. For every argument that was created from R, this set consists of the attack it represents and its source, i.e. we obtain sets $\{a, \alpha\}$, $\{b, \beta\}$, $\{b, \gamma\}$ and $\{c, \delta\}$. The admissible sets of FN^{FR} are $\emptyset$, $\{a, \alpha\}$, $\{a, c, \alpha\}$, $\{a, c, \alpha, \delta\}$ and $\{b, \beta, \gamma\}$. Although all of these sets are admissible in FR, we can observe that some of the admissible sets of FR, such as $\{\alpha\}$, are not admissible in FN^{FR}. This is due to the interpretation of necessary support, which enforces the presence of the source of a given conflict in the extension. Out of these extensions, $\emptyset$, $\{a, c, \alpha, \delta\}$ and $\{b, \beta, \gamma\}$ are complete in FN^{FR}. This time we retrieve all and only complete extension of FR. Our grounded extension is $\emptyset$, while $\{a, c, \alpha, \delta\}$ and $\{b, \beta, \gamma\}$ are preferred and stable. This again is in accordance with the sets produced by FR.

7.4.1 Improvements

We could have observed that the presented AFRA–AFN translation is weak when it comes to conflict–free and admissible semantics. This can be addressed, however, the required modifications would cause this translation to resemble the AFRA–AF approach. In the proof of Theorem 7.6 we have observed that the loss of strength in the case of conflict–freeness was due to the fact that indirect conflicts are not taken into account in the AFN version of the semantics. This can be addressed by transforming the indirect defeats into direct conflicts. However, this brings us to Translation 32 and causes the loss of modularity. Moreover, the support which ties the attacks to their sources becomes redundant, and its presence is the only thing weakening the admissible semantics. Thus, its removal brings us back to the AFRA–AF translation. Although the AFRA–AF and AFRA–AFN have certain structural differences, it is the change in modularity that appears to be the most prominent. Consequently, one needs to choose between the conflict–free and admissible semantics and modularity. Since the decision depends on the application of the translation, we leave it to the reader to define which of the approaches is more desirable.

7.5 AFRA as ADF

Although it will only become visible in the next section, ADFs can handle a certain level of recursion that appears in EAFs. Unfortunately, the recursion available in AFRAs can be deeper than that and thus the construction becomes more complicated. Moreover, for the time being, there are no semantics available for ADFs that would list links in the extensions. Consequently, an AFRA–ADF translation with auxiliary arguments representing the recursive conflicts is the most reasonable way to proceed. We propose to use a chained translation AFRA–AF–ADF or AFRA–AFN–ADF in order to transforms AFRAs to ADFs. We do not believe there is any gain in pursuing a direct method. Unfortunately, for the reasons mentioned in Section 9.4, we cannot use the BAF bypass.

We can note that independently of the chosen bypass framework (see Theorems 5.17 and 10.19), the produced ADF will have all the desirable normal forms – redundancy–free, cleansed, weak, relation and strongly valid. Moreover, it will be both a BADF and an AADF$^+$, and thus any family of ADF semantics can be used for computing the AFRA ones. Although most of the properties of the produced ADF will be an outcome of the properties of the chained translations, there is an exception in the AFN case. The AFRA–AFN–ADF chain will be an injective translation, even though the AFN–ADF translation in principle is not. This is due to the fact that the input AFN for the AFN–ADF part will be always in minimal form.

7.6 AFRA as Other Frameworks

In this section we have not discussed translating AFRAs to SETAFs and EASs. There appears to be no advantage of group attack over binary attack when it comes to the handling of the recursive conflicts. Consequently, we propose to use a chained translation to SETAFs, be it through AF or AFN bypass. The translation to EASs would be almost identical as to AFNs – the only difference would be the addition of the evidence argument, transforming conflict to group form, and adding evidential support to every normal argument. In other words, modifications similar to the ones from AFN–EAS Translation 68 apply here. Therefore, there appears to be no gain in defining an AFRA–EAS translation directly and thus we decide to use a chained one with AF or AFN as bypass.

7.7 Summary

In this section we have presented four translations from AFRAs to other argumentation frameworks. This included the existing translations from AFRAs to AFs [9, 64] and two new ones to BAFs and AFNs. In all of those cases we had to introduce auxiliary arguments to account for the recursive attacks. The only actual differences between these methods concern the strength of the translation and modularity. We could have observed that out of the available approaches, the AFRA–BAF translation appears to be the most interesting. It is the only transformation that is both modular and exact under any semantics. In the original AFRA–AF approach [9] we lose modularity, while in the AFRA–AFN case the conflict–free and admissible semantics are not that well preserved. Finally, the defender AF translation [64] is not exact under any semantics anymore, even though it is modular. The summary of our results is visible in Table 9.

Table 9: Translations from AFRAs to other frameworks

Properties		AF		BAF	AFN
Translation		32	33	34	35
Strength	cf	exact	$\subseteq$–weak	exact	$\subseteq$–weak
	adm	exact	$\supseteq$–weak	exact	$\supseteq$–weak
	comp	exact	faithful	exact	exact
	pref	exact	faithful	exact	exact
	grd	exact	faithful	exact	exact
	stb	exact	faithful	exact	exact
Functional		full	full	full	full
		target–subclass	target–subclass	target–subclass	target–subclass
		injective	injective	injective	injective
Syntactical		argument domain altering	argument domain altering	argument domain altering	argument domain altering
		argument introducing	argument introducing	argument introducing	argument introducing
		induced attack introducing	induced attack introducing	induced support introducing	induced support introducing
Semantical		generic	generic	generic	generic
		semantics domain preserving	semantics domain altering	semantics domain preserving	semantics domain preserving
Computational		semi–structural	semi–structural	semi–structural	semi–structural
			modular	modular	modular

8 Translating EAFs and EAFCs

In Section 2.1.4 we have extensively discussed the extended argumentation framework and noted that its semantics do not follow the usual pattern. By this we understand that the characteristic operator is not necessarily monotonic, which led to the fact that the complete semantics do not conform to the usual semilattice structure and the grounded extension cannot be defined as the least complete set. Moreover, the stable extensions are not necessarily preferred. These design choices mean that translating an EAF to any framework (and semantics) is not a trivial task. Therefore, most of the transformations are performed only for bounded hierarchical EAFs, a special EAF subclass on which the characteristic operator defined on conflict–free extensions is monotonic, where conflict–freeness can be defined using defeats, and the usual relations between stable–preferred and complete–grounded extensions hold (see Lemma 2.60, Theorem 2.59 and Definition 2.57).

Please note that although we will mainly focus on EAFs, when possible we will also make a note on their collective generalization, EAFCs, where defense attacks can be carried out by groups of arguments. The reason why we do not separate those two frameworks, even though e.g. AFs and SETAFs differ in a similar way and did receive separate sections, is semantics. Due to their unusual nature, most of the analysis for those two frameworks will be the same and thus separating them could blur the picture. Moreover, the conflict–free semantics in EAFs and EAFCs can differ on frameworks with symmetric attacks (see Section 2.1.4 and Example 5). Since conflict–freeness is the most basic semantics, changing this notion affects every other type of extension we can obtain. Consequently, providing translations for EAFs and EAFCs alongside can let the reader choose what is more adequate for EAFs – the original semantics, or the EAFC–style ones. This is of course under the assumption we have an approach that can handle an EAF subclass on which both approaches differ.

We will start this section by showing how certain EAFCs can be compiled back to EAFs and vice versa. Since most of our translations will be limited to bounded hierarchical versions of these frameworks – on which their semantics agree – we will often get an EAFC translation simply through chaining or merging the approaches. We will then move on to showing EAF translations to AFs, AFRAs, SETAFs and AFNs. Finally, we will present the ADF transformation, the only one in which we are not strictly limited to bounded hierarchical frameworks.

8.1 EAFC and EAF

Due to the difference in the conflict–free semantics of EAFs and EAFCs, the translations between the two can only be done on a subclass of these frameworks where the definitions coincide. Therefore, we need to assume that our frameworks belong either to the bounded hierarchical subclasses BH^{EAF} and BH^{EAFC} or to those without symmetric attacks – $NSym^{EAF}$ and $NSym^{EAFC}$. Let us start with th EAF–EAFC translation. The approach is very simple and not unlike the AF–SETAF translation:

Translation 36. *Let $EF = (A, R, D)$ be a bounded hierarchical EAF or an EAF s.t. there are no arguments $a, b \in A$ for which $(a, b), (b, a) \in R$. The EAFC corresponding to EF is $EFC^{EF} = (A, R, D')$, where $D' = \{(\{c\}, (a, b)) \mid (c, (a, b)) \in D\}$.*

The resulting EAFC can be described quite easily, as it will inherit any properties of the source EAF. Moreover, since all defense attacks will be carried out by sets of arguments of size 1, it will be in fact a binary EAFC.

Theorem 8.1. *Let $EF = (A, R, D)$ be an EAF and $EFC^{EF} = (A, R, D')$ its corresponding EAFC obtained through Translation 36. EFC^{EF} is a binary EAFC. If EF is bounded hierarchical, then so is EFC^{EF}. If EF has no symmetric attacks, then neither does EFC^{EF}. If EF is (strongly) consistent, then so is EFC^{EF}.*

Theorem 8.2. *Let $EF = (A, R, D)$ be a bounded hierarchical EAF or an EAF s.t. there are no arguments $a, b \in A$ for which $(a, b), (b, a) \in R$ and EFC^{EF} its corresponding EAFC obtained trough Translation 36. A set $E \subseteq A$ is a σ–extension of EF, where $\sigma \in$ {conflict–free, admissible, complete, preferred, grounded, stable}, iff it is a σ–extension of EFC.*

The redefinition of this translation is quite straightforward. The properties are also easily visible from the transformation and require no further explanations.

Redefinition of Translation 36: *Let $BH^{EAF} \cup NSym^{EAF}$ be the collection of all EAFs that are bounded hierarchical or without symmetric attacks and let $Bin^{EAFC} \cap (BH^{EAF} \cup NSym^{EAF})$ the collection of all binary EAFCs that are bounded hierarchical or without symmetric attacks, both based on argument domain $\mathcal{U}$. The translation $Tr^{EAF}_{EAFC} : (BH^{EAF} \cup NSym^{EAF}) \rightarrow (Bin^{EAFC} \cap (BH^{EAF} \cup NSym^{EAF}))$ is defined as $Tr^{EAF}_{EAFC}((A, R, D)) = (A, R, D')$, where $D' = \{(\{c\}, (a, b)) \mid (c, (a, b)) \in D\}$ for a framework $(A, R, D) \in (BH^{EAF} \cup NSym^{EAF})$.*

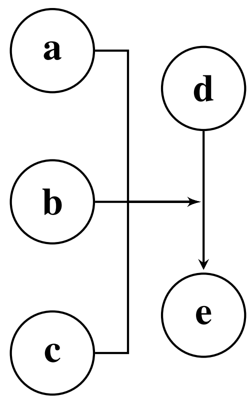
Redefinition of Theorem 8.2: *Let $\sigma \in$ {conflict–free, admissible, preferred, complete, grounded, stable} be a semantics and SC^{Tr}_{σ} the identity casting functions for σ. The translation Tr^{EAF}_{EAFC} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*

Analysis of Translation 36: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and identity casting functions, the translation Tr^{EAF}_{EAFC} is:

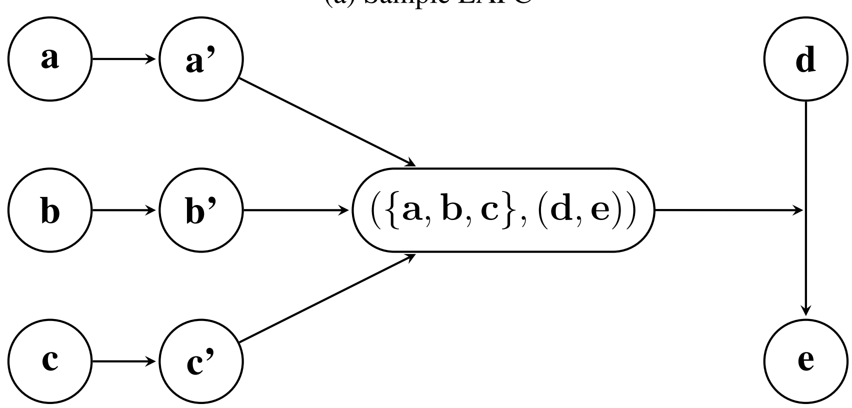
- source–subclass, target–subclass and injective
- argument domain and structure preserving
- generic, semantics domain preserving and exact
- structural and modular

Translation Tr^{EAF}_{EAFC} is classified as basic under the listed semantics and casting functions.

In order to translate EAFCs into EAFs we can reuse the SETAF–AF methods (see Section 6.1). Since most of the EAF translations follow the defender approach, we will create the EAFC–EAF one in the same style. This means that group defense attacks become new arguments that now need to be defended by the arguments carrying them out, as visible in Figure 67.



(a) Sample EAFC



(b) Corresponding EAF

Figure 67: Sample EAFC and its corresponding EAF

Translation 37. *Let $EFC = (A, R, D)$ be a bounded hierarchical EAFC or an EAFC s.t. there are no arguments $a, b \in A$ for which $(a, b), (b, a) \in R$. The corresponding EAF is $EF^{EFC} = (A', R', D')$, where:*

- $A' = A \cup Gr_D \cup X'$, *where* $Gr_D = \{(a, (b, c)) \mid (a, (b, c)) \in D, |a| > 1\}$ *and* $X' = \{x' \mid \exists (a, (b, e)) \in Gr_D, x \in a\}$,
- $R' = R \cup \{(x, x') \mid x \in X'\} \cup \{(x', (a, (b, c))) \mid (a, (b, c)) \in Gr_D, x \in a\}$, *and*
- $D' = D \setminus Gr_D \cup \{((a, b), b) \mid (a, b) \in Gr_D, b \in R\}$.

We can observe that the produced EAF belongs to the same subclass as EAFC. The translation does not introduce any symmetric attacks or cycles that would make it not hierarchical anymore. Furthermore, if a given argument was not attacking another one and defense attacking this conflict, then it does not do so in the produced EAF and thus consistency is also preserved.

Theorem 8.3. *Let $EFC = (A, R, D)$ be an EAFC and $EF^{EFC} = (A', R', D')$ its corresponding EAF obtained through Translation 37. If EFC is bounded hierarchical, then so is EF^{EFC}. If EFC has no symmetric attacks, then neither does EF^{EFC}. If EFC is (strongly) consistent, then so is EF^{EFC}.*

The way we retrieve the extensions in this case resembles the construction from the defender SETAF–AF translation (Translation 26). Unfortunately, just like in this case, the conflict–free semantics is preserved only one way:

Theorem 8.4. *Let $EFC = (A, R, D)$ be a bounded hierarchical EAFC or an EAFC s.t. there are no arguments $a, b \in A$ for which $(a, b), (b, a) \in R$ and EF^{EFC} its corresponding EAFC obtained trough Translation 37. If $E \subseteq A$ is a σ–extension of EFC, where $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ then $E' = E \cup \{(a, (b, c)) \mid (a, (b, c)) \in Gr_D, a \subseteq E\} \cup \{x' \mid E$ defeats$_E$ x and there is a reinstatement set for this defeat on $E\}$ is a σ–extension of EF^{EFC}. If $E' \subseteq A'$ is a σ'–extension of EF^{EFC}, where $\sigma' \in \{$admissible, complete, preferred, grounded, stable$\}$ then $E = E' \cap A$ is a σ'–extension of EFC. This does not necessarily hold for conflict–free semantics.*

The can now redefine and analyze our EAFC–EAF translation. The property analysis is almost the same as in the case of the defender SETAF–AF translation (Translation 26) and thus will be omitted.

Redefinition of Translation 37: *Let $BH^{EAFC} \cup NSym^{EAFC}$ be the collection of all EAFCs that are bounded hierarchical or without symmetric attacks on domain $\mathcal{U}$ and let $BH^{EAF} \cup NSym^{EAF}$ be the collection of all EAFs that are bounded hierarchical or without symmetric attacks on domain $\mathcal{U} \cup GD(\mathcal{U}) \cup \mathcal{U}'$, where $GD(\mathcal{U}) = (2^{\mathcal{U}} \setminus \emptyset) \times (\mathcal{U} \times \mathcal{U})$. The translation $Tr^{EAFC}_{EAF} : (BH^{EAFC} \cup NSym^{EAFC}) \rightarrow (BH^{EAFC} \cup NSym^{EAF})$ is defined as $Tr^{EAFC}_{EAF}((A, R, D)) = (A', R', D')$, where:*

- *$A' = A \cup Gr_D \cup X'$, where $Gr_D = \{(a, (b, c)) \mid (a, (b, c)) \in D, |a| > 1\}$ and $X' = \{x' \mid \exists (a, (b, e)) \in Gr_D, x \in a\}$,*
- *$R' = R \cup \{(x, x') \mid x \in X'\} \cup \{(x', (a, (b, c))) \mid (a, (b, c)) \in Gr_D, x \in a\}$, and*
- *$D' = D \setminus Gr_D \cup \{((a, b), b) \mid (a, b) \in Gr_D, b \in R\}$.*

Redefinition of Theorem 8.4: *Let $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the removal casting functions for σ defined as $SC^{X}_{\sigma}(E) = E \cap A$, where $X = (A, R, D) \in BH^{EAFC} \cup NSym^{EAFC}$ and $E \in$*

$\sigma(Tr^{EAFC}_{EAF}(X))$. The translation Tr^{EAFC}_{EAF} is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under conflict–free semantics and removal casting functions. For complete, preferred, grounded and stable semantics, the translation is semantics bijective.

Analysis of Translation 37: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and removal casting functions, the translation Tr^{EAFC}_{EAF} is:

- source–subclass, target–subclass and injective
- argument domain altering, argument introducing and induced attack introducing
- generic and semantics domain altering
- semi–structural and modular

Under the complete, preferred, grounded and stable semantics and removal casting functions, Tr^{EAFC}_{EAF} is faithful. Translation Tr^{EAFC}_{EAF} is classified as basic–defender hybrid under the listed semantics and casting functions.

8.1.1 Improvements

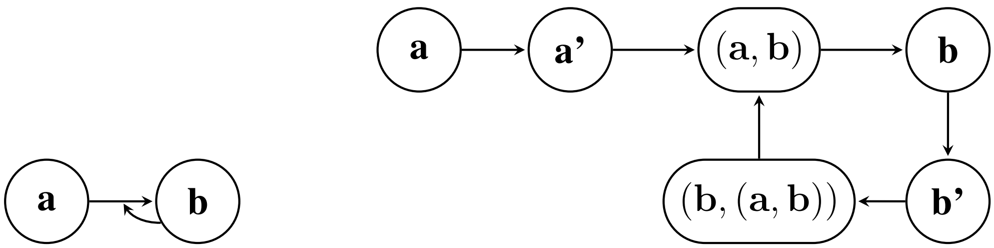
One of the most important improvements that can be done to the presented translations concerns fullness, i.e. devising a way such that every type of EAF(C) can undergo a translation. Unfortunately, we are not aware of any solution for now, and this task is left for future work. The EAF–EAFC translation is already exact and modular; however, the EAFC–EAF is only faithful, and weak under the conflict–free semantics. Due to the lack of research on the semantic signatures of both frameworks, we do not know whether exactness is achievable. We only expect it not to be the case based on the SETAF–AF relation.

8.2 EAF as AF

The translation from EAFs to AFs is similar to the defender SETAF–AF and AFRA–AF approaches (see Sections 6.1.2 and 7.1.2). The conflicts in the source framework become new arguments in the target structure and are connected to the arguments carrying them out via defense [18,47,64]. In Figures 61 and 62 in Section 7.1.2 we have depicted the two main approaches. In the first work [18], only direct attacks would receive corresponding arguments, while in the other method [64] both direct and defense conflicts would undergo a transformation. This difference affected the behavior of the semantics in AFRA–AF transformation due to the fact that conflicts appear in extensions. However, as already noted in [47], any of the methods is acceptable for EAFs. We will follow the construction introduced in [64] due to the fact that we have worked with it in the previous SETAF and AFRA translations.

It is important to note that even though the conflict expansion approach performed well with other argumentation frameworks [18, 64], it misbehaves when it comes to EAFs. By this we understand that the extensions produced by the target AFs are not always related to the ones in the source EAF. Let us look at the following examples:

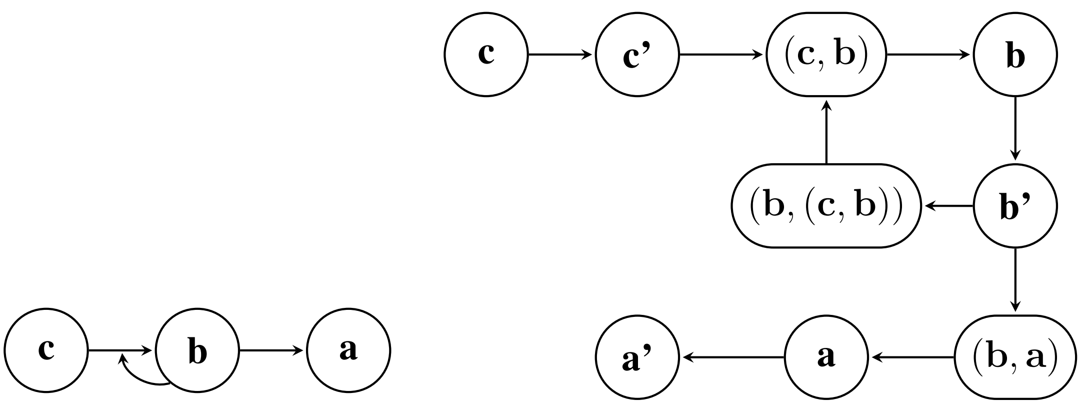
Example 93. Let $EF_1 = (\{a, b\}, \{(a, b)\}, \{(b, (a, b))\})$ be the EAF depicted in Figure 68a [64]. We can observe it is not a hierarchical one. It has a single preferred extension $\{a, b\}$. However, the corresponding AF $(\{a, b, a', b', (a, b), (b, (a, b))\}, \{(a, a'), (b, b'), (a', (a, b)), (b', (b, (a, b))), ((a, b), b), ((b, (a, b)), (a, b))\})$, visible in Figure 68b has two preferred extensions – $\{a, (a, b), b'\}$ and $\{a, b, (b, (a, b))\}$. We can observe the first one does not correspond to the preferred extension of EF_1.



(a) EF_1 (b) Corresponding defender AF

Figure 68: EF_1 and its corresponding AF from [64]

Example 94 (Adapted from [9])**.** Let us consider the framework $EF_2 = (\{a, b, c\}, \{(b, a), (c, b)\}, \{(b, (c, b))\})$ depicted in Figure 69a. The sets $\{a, c\}$ and $\{b, c\}$ are complete extensions of EF_2. The set $\{a, c\}$ is also the grounded extension. However, the complete extensions of the corresponding AF visible in Figure 69b are $\{c\}$, $\{b, c, a', (b, a), (b, (c, b))\}$ and $\{a, c, b', (c, b)\}$, where $\{c\}$ is also grounded. Therefore, again we obtain a mismatch between the EAF and AF extensions.



(a) EF_2 (b) Corresponding defender AF

Figure 69: EF_2 and its corresponding AF

Unfortunately, such issues were to be expected. We have already noted at the beginning of this section that the EAF semantics are structurally different from e.g. the AF and SETAF ones. By this we understand the lack of monotonicity of the characteristic operator and the defeats, i.e. an argument defeated by a given set E is not necessarily defeated by $E' \supseteq E$. The presented conflict expansion does not build a gap between the non–monotonic EAF and monotonic AF semantics. As observed in the example with the self–reinstating argument, the inclusion of defeats in an extension, as done in the presented approach, can make comparable EAF extensions incomparable in the target AF. Therefore, the EAF–AF translation is limited to bounded hierarchical EAFs, on which the characteristic operator becomes monotonic (see Section 2.1.4) and the self–reinstatement is no longer an issue.

We can now proceed with introducing the translation. Please note that just like in the SETAF–AF and AFRA–AF approaches (see Sections 6.1.2 and 7.1.2), we will use a simplified version of the original meta–level framework [64]. However, it is only the naming of the arguments that is, in fact, different, and thus the original results still hold [64]. Please note we will also reuse the src and trg notation from AFRAs. This means that given a (direct or defense) attack $x = (a, b)$, $src(x) = a$ and $trg(x) = b$, where depending on the type of conflict, b is either an argument or a direct attack.

Translation 38. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF. Its corresponding AF is $F^{EF} = (A', R')$ for $A' = A \cup R \cup D \cup X'$, where $X' = \{x' \mid x \in A\}$, and $R' = \{(x, x') \mid x \in A\} \cup \{(x', a) \mid a \in R \cup D, src(a) = x\} \cup \{(a, b) \mid trg(a) = b$, and either $(a \in R$ and $b \in A)$, or $(a \in D$ and $b \in R)\}$.*

The original results concerning the semantics spoke in terms of skeptical and credulous acceptance and focused on semantics that were at least complete. However, the proofs also include the admissible semantics and do in fact show the correspondence between the source and target extensions. Please note that skeptical acceptance as such is not analyzed in the case of admissible semantics – due to the fact that $\emptyset$ is always admissible, no argument can be skeptically justified.

Theorem 8.5. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF, $F^{EF} = (A', R')$ its corresponding AF obtained by Translation 38 and $\sigma \in \{$admissible, complete, grounded, preferred, stable$\}$ a semantics. Argument $a \in A$ is a credulously, respectively skeptically (if applicable), justified argument of $bh - EF$ under the semantics σ iff it is a credulously, respectively skeptically, justified argument of F^{EF} under σ.*

Theorem 8.6. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF and $F^{EF} = (A', R')$ its corresponding AF obtained by Translation 38. If $E \subseteq A$ is a σ–extension of $bh - EF$ for $\sigma \in \{$conflict–free, admissible, complete, grounded, preferred, stable$\}$, then there is a σ–extension $E' \subseteq A'$ of F^{EF} s.t. $E = E' \cap A$. If $E' \subseteq A'$ is a σ'–extension of F^{EF} for $\sigma' \in \{$admissible, complete, grounded, preferred, stable$\}$, then $E = E' \cap A$ is a σ'–extension of $bh - EF$. This does not necessarily hold for conflict–free semantics.*

Please note we have included the results for conflict–free semantics as well. The proof of that is trivial and can be easily adapted from e.g. Theorem 6.2. Although the correspondence between the admissible extensions of $bh-EF$ and F^{EF} is not one–to–one (we have some freedom when it comes to the addition of primed versions of arguments that do not carry out attacks), it is such when we consider the complete semantics. The bounded hierarchical nature of the EAFs we consider means that sooner or later, defense of any conflict or primed argument in the target AF can be tracked back to the usual arguments. Please note it would not necessarily be the case if we considered arbitrary EAFs; we can again look at Example 93, in which the EAF complete extension $\{a\}$ had two corresponding AF complete ones, $\{a\}$ and $\{a, b', (a, b)\}$.

We can now proceed to redefine and analyze the translation:

Redefinition of Translation 38: *Let BH^{EAF} be the collection of all bounded hierarchical EAFs based on domain $\mathcal{U}$ and Fr^{AF} the collection of all AFs based on the domain $\mathcal{U}' \cup \bigcup_{i=1}^{3} \mathcal{U}^i$, where $\mathcal{U}^1 = \mathcal{U}$ and $\mathcal{U}^i = \mathcal{U} \times \mathcal{U}^{i-1}$ for $i > 1$ The translation $Tr_{AF}^{EAF} : BH^{EAF} \rightarrow Fr^{AF}$ is defined as $Tr_{AF}^{EAF}((A, R, D)) = (A', R')$, where $A' = A \cup R \cup D \cup X'$, where $X' = \{x' \mid x \in A\}$, and $R' = \{(x, x') \mid x \in A\} \cup \{(x', a) \mid a \in R \cup D, src(a) = x\} \cup \{(a, b) \mid trg(a) = b$, and either $(a \in R$ and $b \in A)$, or $(a \in D$ and $b \in R)\}$ for a framework $(A, R, D) \in BH^{EAF}$.*

Redefinition of Theorem 8.6: *Let $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC_{σ}^{Tr} the removal casting functions for σ defined as $SC_{\sigma}^{X}(E) = E \cap A$, where $X = (A, R, D) \in BH^{EAF}$ is a bounded hierarchical EAF and $E \in \sigma(Tr_{AF}^{EAF}(X))$. The translation Tr_{AF}^{EAF} is strong under $(\sigma, SC_{\sigma}^{Tr})$. It is $\subseteq$–weak under the conflict–free semantics and the defined casting function. It is semantics bijective under complete, grounded, preferred and stable semantics.*

Analysis of Translation 38: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and union casting functions, the translation Tr_{AF}^{EAF} is:

- source–subclass, target–subclass and injective
- argument domain altering, argument introducing, induced attack relation introducing
- generic and semantics domain altering
- semi–structural and modular

The translation Tr_{AF}^{EAF} is faithful under complete, preferred, grounded and stable semantics and the defined removal casting functions. We classify Tr_{AF}^{EAF} as a basic–defender hybrid under the listed semantics and casting functions.

Explanation. The translation only works with bounded hierarchical EAFs; consequently, it is source–subclass. Although we do not have a precise description of the subclass of AFs that can be obtained with this transformation, there exist frameworks that cannot be produced. For example, an AF containing a self–attacking argument cannot be created with this approach. We can observe that the set of arguments in the target AF is uniquely defined by the structure of the source EAF. Consequently the translation is injective.

Both argument and semantics domain is altered. Due to the presence of the primed arguments, we classify the translation as argument introducing. It is not relation removing – every attack and defense attack is represented in the target structure, even if by the use of auxiliary arguments. Having said that, adding the primed arguments requires new attacks, and thus their introduction is induced. The amount of handled semantics makes the translation generic. We also classify it as semi–structural, as it uses the defense to connect the arguments with the attacks and defense attacks they carry out. Just like in the SETAF and AFRA cases (see Section 6.1.2 and 7.1.2), the translation is modular. Every conflict is separately "expanded" and no propagation is present. ■

8.2.1 EAFC as AF

The extended argumentation frameworks we have used in Examples 93 and 94 do not have symmetric attacks. This means that we can use the defeat–based definition of conflict–freeness to obtain the desired extensions, even though the frameworks are not bounded hierarchical. Consequently, if we represented these structures as collective EAFs, their extensions would remain the same. Therefore, our previous analysis holds for EAFCs as well, and thus a translation from EAFCs to AFs needs to be limited to bounded hierarchical frameworks as well (see Definition 2.61).

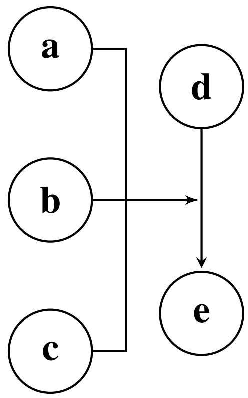
Having said that, an EAFC–AF translation is in fact a merge between the presented EAFC–EAF and EAF–AF approaches. In other words, conflicts still become new arguments that are defended by their sources, with the difference being that now a source can be more than a single argument. However, since group attacks of size bigger than 1 already undergo this treatment in the EAFC–EAF step (see Translation 37), they need to be omitted in the EAF–AF one. Consequently, simple chaining would be overly redundant. An example of this method can be seen in Figure 70.

The EAFC–AF translation is now a simple modification of the EAF–AF one. Like previously, we reuse the AFRA src and trg notation, with the difference that the source of a defense attack is now a set:

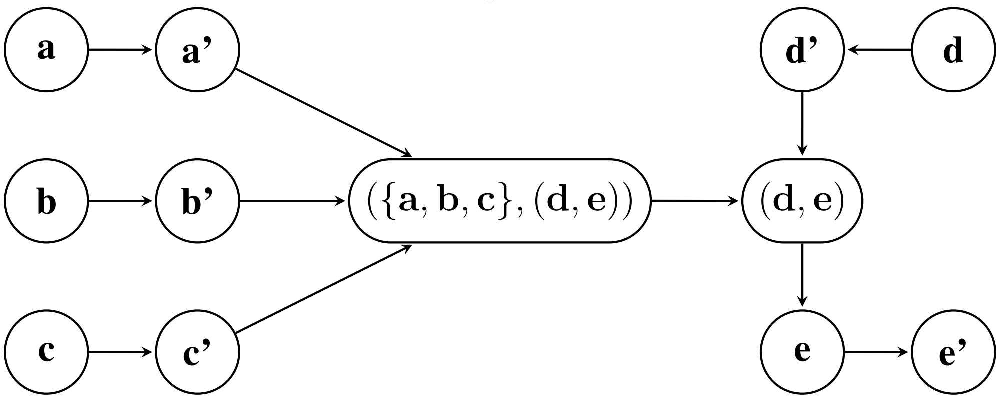
Translation 39. *Let $bh - EFC = (A, R, D)$ be a bounded hierarchical EAFC. Its corresponding AF is $F^{EFC} = (A', R')$ for $A' = A \cup R \cup D \cup X'$, where $X' = \{x' \mid x \in A\}$, and $R' = \{(x, x') \mid x \in A\} \cup \{(x', a) \mid a \in R, src(a) = x\} \cup \{(x', a) \mid a \in D, x \in src(a)\} \cup \{(a, b) \mid trg(a) = b$, and either $(a \in R$ and $b \in A)$, or $(a \in D$ and $b \in R)\}$.*

We can observe that the modification done to the EAF–AF translation is not a drastic one. Thus, the original proof can be adapted, and the existing semantical results carry over:

Theorem 8.7. *Let $bh - EFC = (A, R, D)$ be a bounded hierarchical EAFC and $F^{EF} = (A', R')$ its corresponding AF obtained by Translation 39. If $E \subseteq A$ is a σ–extension of $bh - EFC$ for $\sigma \in \{$conflict–free, admissible, complete, grounded, preferred, stable$\}$, then*



(a) Sample EAFC



(b) Corresponding defender AF

Figure 70: Sample EAFC and its corresponding AF

there is a σ–extension $E' \subseteq A'$ of F^{EFC} s.t. $E = E' \cap A$. If $E' \subseteq A'$ is a σ'–extension of F^{EFC} for $\sigma' \in \{admissible, complete, grounded, preferred, stable\}$, then $E = E' \cap A$ is a σ'–extension of $bh - EFC$. This does not necessarily hold for conflict–free semantics.

The EAFC–AF translation has the same properties as the EAF–AF one and thus we will omit the explanations. Only the target domain needs to be slightly adjusted in order to account for the group defense attacks.

Redefinition of Translation 39: *Let BH^{EAFC} be the collection of all bounded hierarchical EAFCs based on domain $\mathcal{U}$ and Fr^{AF} the collection of all AFs based on the domain $\mathcal{U}' \cup \bigcup_{i=1}^{2} \mathcal{U}^i \cup (\mathcal{U} \times \mathcal{U}^2)$, where $\mathcal{U}^1 = \mathcal{U}$ and $\mathcal{U}^i = \mathcal{U} \times \mathcal{U}^{i-1}$ for $i > 1$. The translation $Tr_{AF}^{EAFC} : BH^{EAFC} \rightarrow Fr^{AF}$ is defined as $Tr_{AF}^{EAFC}((A, R, D)) = (A', R')$, where $A' = A \cup R \cup D \cup X'$, where $X' = \{x' \mid x \in A\}$, and $R' = \{(x, x') \mid x \in A\} \cup \{(x', a) \mid a \in R, src(a) = x\} \cup \{(x', a) \mid a \in D, x \in src(a)\} \cup \{(a, b) \mid trg(a) = b$, and either $(a \in R$ and $b \in A)$, or $(a \in D$ and $b \in R)\}$ for a framework $(A, R, D) \in BH^{EAFC}$.*

Redefinition of Theorem 8.7: *Let $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC_{σ}^{Tr} the removal casting functions for σ defined as $SC_{\sigma}^{X}(E) = E \cap A$, where $X = (A, R, D) \in BH^{EAFC}$ is a bounded hierarchical EAFC* *and $E \in \sigma(Tr_{AF}^{EAFC}(X))$. The translation Tr_{AF}^{EAFC} is strong under $(\sigma, SC_{\sigma}^{Tr})$. It is $\subseteq$–weak under the conflict–free semantics and the defined casting function. It is semantics bijective under complete, grounded, preferred and stable semantics.*

Analysis of Translation 39: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and union casting functions, the translation Tr_{AF}^{EAFC} is:

- source–subclass, target–subclass and injective
- argument domain altering, argument introducing, induced attack relation introducing
- generic and semantics domain altering
- semi–structural and modular

The translation Tr_{AF}^{EAFC} is faithful under the complete, preferred, grounded and stable semantics and the defined removal casting functions. We classify Tr_{AF}^{EAFC} as a basic–defender hybrid under the listed semantics and casting functions.

8.2.2 Improvements

The EAF–AF translation we have introduced is modular and faithful for semantics that are also complete. Unfortunately, the approach is also source–subclass. Consequently, we would like to know whether a full and/or exact translation is even possible and if a wider range of semantics can be obtained.

Let us first start with the general EAFs. We can come back to the previously analyzed Example 90. The framework $EF = (\{a, b, c, d, e, f, g\}, \{(a, b), (d, c), (b, e), (e, f), (f, g)\}, \{(b, (d, c)), (c, (a, b))\})$ had numerous conflict–free extensions. However, what is important is that $\{a, b, c\}$ was one, while $\{a, b\}$ was not. Consequently, the set of extensions $cf(EF)$ is not downward closed and thus there is no AF F s.t. $cf(EF) = cf(F)$ (see Definitions 2.176 and 2.177).

The EF framework had only two complete extensions - $\{a, d, e, g\}$ and $\{a, b, c, d, f\}$. Although the com–closed requirement is satisfied, the intersection of the two sets – $\{a, d\}$ – is not present. Consequently, by Proposition 2.178 there cannot be an AF F s.t. $comp(F) = \{\{a, d, e, g\}, \{a, b, c, d, f\}\}$. Finally, concerning the stable semantics, we can come back to Example 9. The described framework $EF' = (\{a, b, c, d\}, \{(a, b), (d, c)\}, \{(b, (d, c)), (c, (a, b))\})$ had two stable extensions, $\{a, d\}$ and $\{a, b, c, d\}$. They are clearly comparable and thus breach the stable signature of AFs (see Definition 2.177). This brings us to the following result:

Theorem 8.8. *Let Fr^{EAF} be the collection of all EAFs on a domain $\mathcal{U}^{EAF}$ and Fr^{AF} the collection of all AFs on a domain $\mathcal{U}^{AF}$. There does not exist a full translation from Fr^{EAF} to Fr^{AF} that is exact under conflict–free, complete and stable semantics and their identity casting functions.*

Since the EAFC representation of our EAF EF has the same extensions due to lack of symmetric attacks, this result can be reiterated also for them:

Theorem 8.9. *Let Fr^{EAFC} be the collection of all EAFCs on a domain $\mathcal{U}^{EAFC}$ and Fr^{AF} the collection of all AFs on a domain $\mathcal{U}^{AF}$. There does not exist a full translation from Fr^{EAFC} to Fr^{AF} that is exact under conflict–free, complete and stable semantics and their identity casting functions.*

Please note it does not appear that even a faithful translation can be created for the complete semantics – no addition of auxiliary arguments in a way that still a removal casting function can be used will bring e.g. the missing intersection set in our EF.

The question concerning whether there exist exact translations from EAFs and EAFCs to AFs for the admissible and preferred semantics is unfortunately still open. Therefore, we hope to answer it in the future.

We can now limit ourselves to the bounded hierarchical EAFs. We can observe that when it comes to conflict–free semantics, again the answer concerning exactness is "no". The conflict–free extensions of a simple bh–EAF $(\{a, b, c\}, \{(a, b)\}, \{(c, (a, b))\})$ are $\emptyset, \{a\}, \{b\}, \{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}$. Due to the absence of the set $\{a, b\}$, our collection is not downward closed and thus does not fit the AF conflict–freeness signature. It however does appear that due to the fact that now the stable extensions are incomparable, we might be able to realize them. Unfortunately, this is all we are able to say – the question concerning admissible and preferred semantics is still open, and only the necessary conditions are known for complete. Therefore, the search for exact translations for bounded hierarchical frameworks is again left for future work.

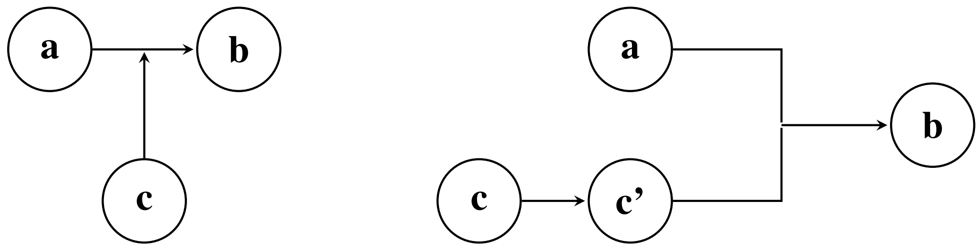
8.3 EAF as SETAF

Although we did not focus on translating group attack into defense attack, there is a way to express the combination of direct and defense conflicts as group conflicts. An EAF argument cannot be accepted if its attacker is present and every relevant defense attacker is absent. Thus, with the use of auxiliary arguments representing the rejection of a given defense attacker, the described situation can be expressed with group attacks. Let us look at the following example:

Example 95. Let us assume a simple framework $EF_1 = (\{a, b, c\}, \{(a, b)\}, \{(c, (a, b))\})$ visible in Figure 71a that will be our main example for defense attack transformation in this section. The set $\{b\}$ is a conflict–free extension, i.e. the argument can stand on its own. However, it requires the presence of c in order to be accepted whenever a is around. In other words, while $\{a, b\}$ is not conflict–free, $\{a, b, c\}$ is. Concerning admissibility, we receive the following extensions: $\emptyset$, $\emptyset$, $\{a\}$, $\{c\}$, $\{a, c\}$, $\{b, c\}$ and $\{a, b, c\}$.

A similar situation occurs in our SETAF in which c prevents the acceptance of c' and thus can stop the group attack on b carried out by $\{a, c'\}$, as seen in Figure 71b. The framework $SF_1 = (\{a, b, c, c'\}, \{(\{a, c'\}, b), (\{c\}, c')\})$ gives us the following conflict–free extensions: $\emptyset$, $\{a\}$, $\{c\}$, $\{c'\}$, $\{b\}$, $\{a, b\}$, $\{a, c\}$, $\{a, c'\}$, $\{b, c\}$, $\{b, c'\}$ and $\{a, b, c\}$.

We can observe that if we remove the auxiliary argument c' from the extensions (with the exception of $\{a, b\}$), we retrieve the conflict–free extensions of EF_1. However, if we limit ourselves to admissible sets – $\emptyset$, $\{a\}$, $\{c\}$, $\{a, c\}$, $\{b, c\}$ and $\{a, b, c\}$ – we obtain the desired result.



(a) EF_1 (b) Corresponding SF_1

Figure 71: Sample EAF and its corresponding SETAF

Our translation is now as follows. Please note that for similar reasons as in the EAF–AF case, we need to limit ourselves to the bounded hierarchical frameworks:

Translation 40. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF. Its corresponding SETAF is $SF^{EF} = (A', R')$, where:*

- *$A' = A \cup X'$, where $X' = \{x' \mid x \in A, \exists y \in R$ s.t. $(x, y) \in D\}$, and*
- *$R' = \{(\{x\}, x') \mid x' \in X'\} \cup \{(datt'(x, y) \cup \{x\}, y) \mid (x, y) \in R\}$, where $datt'(x, y) = \{c' \mid (c, (x, y)) \in D\}$.*

Although due to the restrictions on the source frameworks not every type of a SETAF can be produced, for now it suffices to know that the target framework will always be minimal. This is simply a result of the fact that each attack set will contain a unique, unprimed argument representing the direct attacker.

Theorem 8.10. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF and $SF^{EF} = (A', R')$ its corresponding SETAF obtained through Translation 40. SF^{EF} is minimal.*

Let us now focus on the semantics. As already observed in Example 95, the conflict–free semantics will be preserved only one–way:

Theorem 8.11. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF and $SF^{EF} = (A', R')$ its corresponding SETAF obtained through Translation 40. If $E \subseteq A$ is a σ–extension of $bh - EF$, where $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$, then $E' = E \cup \{x' \mid x' \in X', x \in E^+\}$ is a σ–extension of SF^{EF}. If $E' \subseteq A'$ is a σ'–extension of SF^{EF}, where $\sigma' \in \{$admissible, complete, preferred, grounded, stable$\}$, then $E' \cap A$ is a σ'–extension of $bh - EF$. This does not necessarily hold for conflict–free semantics.*

We can now put these results into our system:

Redefinition of Translation 40: *Let BH^{EAF} be the collection of all bounded hierarchical EAFs based on domain $\mathcal{U}$ and Min^{SETAF} the collection of all minimal SETAFs based on the domain $\mathcal{U} \cup \mathcal{U}'$. The translation $Tr^{EAF}_{SETAF} : BH^{EAF} \rightarrow Min^{SETAF}$ is defined as $Tr^{EAF}_{SETAF}((A, R, D)) = (A', R')$ for a framework $(A, R, D) \in BH^{EAF}$, where $A' = A \cup X'$ with $X' = \{x' \mid x \in A, \exists y \in R$ s.t. $(x, y) \in D\}$, and $R' = \{(\{x\}, x') \mid x' \in X'\} \cup \{(datt'(x, y) \cup \{x\}, y) \mid (x, y) \in R\}$ s.t. $datt'(x, y) = \{c' \mid (c, (x, y)) \in D\}$.*

Redefinition of Theorem 8.11: *Let $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the removal casting functions for σ defined as $SC^{X}_{\sigma}(E) = E \cap A$, where $X = (A, R, D) \in BH^{EAF}$ is a bounded hierarchical EAF and $E \in \sigma(Tr^{EAF}_{SETAF}(X))$. The translation Tr^{EAF}_{SETAF} is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under the conflict–free semantics and the defined casting function. It is semantics bijective under complete, grounded, preferred and stable semantics.*

Analysis of Translation 40: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and union casting functions, the translation Tr^{EAF}_{SETAF} is:

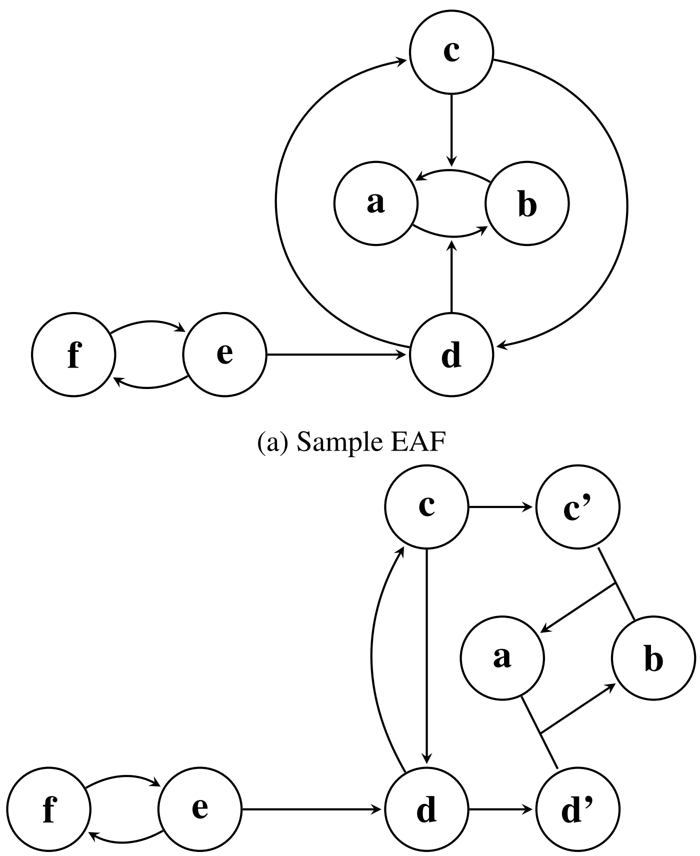
- source–subclass, target–subclass and injective
- weakly argument domain altering, argument introducing, induced attack relation introducing
- generic and weakly semantics domain altering
- structural

The translation Tr^{EAF}_{SETAF} is not modular. It is faithful under complete, preferred, grounded and stable semantics and the defined removal casting functions. We classify Tr^{EAF}_{SETAF} as a basic under the listed semantics and casting functions.

Explanation. Our translation is designed for bounded hierarchical EAFs only, thus it is source–subclass. This restriction also makes it impossible to produce certain SETAFs. However, we can observe that the subclass of minimal framework is not the most accurate description possible. For example, let us consider the framework $SF_1 = (\{a, b, c\}, \{(\{a, b\}, c), (\{c\}, a), (\{c\}, b))$. First of all, a and b carry out a group attack on c. We can observe that one of them has to be a primed argument, representing the rejection of a defense attacker, and as such needs to be attacked by the original argument – in this case, it can only be c. Thus, c has to be its own defense attacker, which violates the bounded hierarchical restrictions on the source EAFs.

For every produced SETAF, it is rather straightforward to recreate the EAF creating it. The source arguments can be retrieved easily by filtering out the primed elements. Moreover, the primed arguments can only be attacked by their originals. Thus, they clearly point to the defense attacker of the conflict represented by the single not primed argument appearing in the group attack. Therefore, it is not possible that two different EAFs will produce the same SETAF, and our translation is injective.

Our translation alters the argument and semantics domains, though only weakly. Moreover, we introduce the auxiliary arguments and their related conflicts. The amount of handled semantics classifies our approach as generic. However, it is not modular. Let us consider two frameworks $EF_1 = (\{a, b\}, \{(a, b)\}, \emptyset)$ and $EF_2 = (\{a, b, c\}, \{(a, b)\}, \{(c, (a, b))\})$. Their union is simply EF_2 and its corresponding SF_2 is $(\{a, b, c, c'\}, \{(\{c\}, c'), (\{a, c'\}, b)\})$, as seen in Example 95. However, the SETAF related to EF_1 is $SF_1 = (\{a, b\}, \{(\{a\}, b)\})$. We can observe that $SF_1 \cup SF_2$ is not the same as SF_2 – the set of attacks is different. Therefore, our approach is not modular. ■



(a) Sample EAF

(b) Associated SETAF

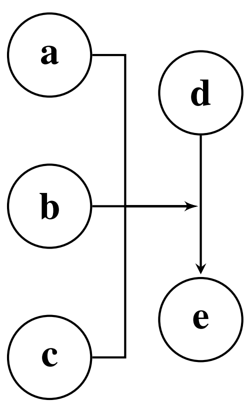
Figure 72: Sample EAF and its associated SETAF

Example 96. Let us come back to the bounded hierarchical EAF $EF = (\{a, b, c, d, e, f\}$, $\{(a, b)$, (b, a), (c, d), (d, c), (e, d), (e, f), $(f, e)\}$, $\{(c, (b, a))$, $(d, (a, b))\})$ previously described in Example 7. We can see it depicted in Figure 72a. The associated SETAF, visible in Figure 72b, is $SF^{EF} = (\{a, b, c, d, e, f, c', d'\}$, $\{(\{a, d'\}, b)$, $(\{b, c'\}, a)$, $(\{c\}, d)$, $(\{d\}, c)$, $(\{e\}, d)$, $(\{e\}, f)$, $(\{f\}, e)$, $(\{c\}, c')$, $(\{d\}, d')\})$. The admissible sets of SF^{EF} are $\emptyset$, $\{c\}$, $\{e\}$, $\{f\}$, $\{a, c\}$, $\{c, d'\}$, $\{c, e\}$, $\{c, f\}$, $\{d, f\}$, $\{e, d'\}$, $\{a, c, e\}$, $\{a, c, f\}$, $\{a, e, d'\}$, $\{a, c, d'\}$, $\{b, d, f\}$, $\{c, e, d'\}$, $\{c, f, d'\}$, $\{d, f, c'\}$, $\{a, c, f, d'\}$, $\{a, c, e, d'\}$ and

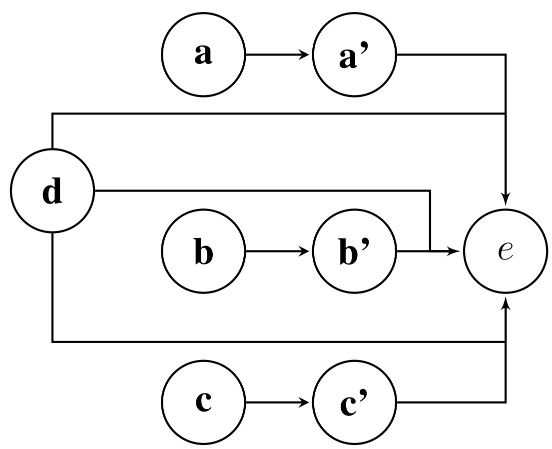
$\{b, d, f, c'\}$. When we remove the primed arguments, we obtain all and only extensions of EF, though please note that the relation is not one–to–one (e.g. the set $\{c\}$ can be obtained both from $\{c\}$ and $\{c, d'\}$). The complete extensions of SF^{EF} are $\emptyset$, $\{f\}$, $\{a, c, d'\}$, $\{a, c, e, d'\}$, $\{a, c, f, d'\}$ and $\{b, d, f, c'\}$. We can again retrieve all the desired complete extensions of EF. It is worth observing that this time, the relation is one–to–one. $\emptyset$, just like in EF, is the grounded extension of SF^{EF}. The sets $\{a, c, e, d'\}$, $\{a, c, f, d'\}$ and $\{b, d, f, c'\}$ are both stable and preferred, which is in correspondence with the answers obtained from EF.

8.3.1 EAFC as SETAF

In the EAFs, defense attacks are carried out by single arguments only and thus every direct attack would be assigned a unique group attack in the result of the EAF–SETAF translation. In EAFCs, we do not have this benefit, although the construction follows similar principles as in the EAF case. Previously, an argument could not be accepted if its attacker were present and all of the defense attackers absent. Now, an argument cannot be accepted if its attacker is present and at least one argument out of each relevant defense attacking set is absent. An example of this behavior can be observed in Figure 73; we can observe that the group defense attack $(\{a, b, c\}, (d, e))$ is transformed into three group attacks – $(\{a', d\}, e)$, $(\{b', d\}, e)$ and $(\{c', d\}, e)$.



(a) Sample EAFC



(b) Related SETAF

Figure 73: Sample EAFC and a related SETAF

Let us now introduce the translation; please note that once again, we need to restrict ourselves to bounded hierarchical target frameworks.

Translation 41. *Let $bh - EFC = (A, R, D)$ be a bounded hierarchical EAFC. Its corresponding SETAF is $SF^{EFC} = (A', R')$, where:*

- *$A' = A \cup X'$, where $X' = \{x' \mid \exists y \in R, B \subseteq A$ s.t. $x \in B$ and $(B, y) \in D\}$, and*

- $R' = \{(\{x\}, x') \mid x' \in X'\} \cup \{(G \cup \{x\}, y) \mid (x, y) \in R, G \subseteq \bigcup gdatt'(x, y)$ *s.t.* $\forall C \in gdatt'(x, y), G \cap C \neq \emptyset\}$, *where* $gdatt'(x, y) = \{\{c'_1, ...c'_n\} \mid (\{c_1, ..., c_n\}, (x, y)) \in D\}$.

It is important to notice that according to this translation, the SETAF depicted in Figure 73 would be the minimal form of the framework associated with the presented EAF, not the framework itself. The way G sets are constructed allows certain redundant (though still correct) sets of attackers to show up in the target SETAF. Consequently, a minimal EAFC can be turned into a non–minimal SETAF. We can thus decide to limit ourselves to minimal G's in the construction of R'. However, doing so can make the target SETAFs minimal, even if the source EAFCs are not. Therefore, we leave the decision on the creation of group attacks to the reader, and perform the analysis of the current version.

The proof of Theorem 8.12 can be easily adapted in order to show that the semantics behave similarly in the EAFC–SETAF case as in the EAF–SETAF one:

Theorem 8.12. *Let* $bh - EFC = (A, R, D)$ *be a bounded hierarchical EAFC and* $SF^{EFC} = (A', R')$ *its corresponding SETAF obtained through Translation 41. If* $E \subseteq A$ *is a* σ*–extension of* $bh - EFC$*, where* $\sigma \in \{$*conflict–free, admissible, complete, preferred, grounded, stable*$\}$*, then* $E' = E \cup \{x' \mid x' \in X', x \in E^+\}$ *is a* σ*–extension of* SF^{EFC}*. If* $E' \subseteq A'$ *is a* σ'*–extension of* SF^{EFC}*, where* $\sigma' \in \{$*admissible, complete, preferred, grounded, stable*$\}$*, then* $E' \cap A$ *is a* σ'*–extension of* $bh - EFC$*. This does not necessarily hold for conflict–free semantics.*

Redefinition of Translation 41: *Let* BH^{EAFC} *be the collection of all bounded hierarchical EAFCs based on domain* $\mathcal{U}$ *and* Fr^{SETAF} *the collection of all SETAFs based on the domain* $\mathcal{U} \cup \mathcal{U}'$*. The translation* $Tr^{EAFC}_{SETAF} : BH^{EAFC} \rightarrow Fr^{SETAF}$ *is defined as* $Tr^{EAFC}_{SETAF}((A, R, D)) = (A', R')$ *for a framework* $(A, R, D) \in BH^{EAFC}$*, where* $A' = A \cup X'$ *with* $X' = \{x' \mid \exists y \in R, B \subseteq A$ *s.t.* $x \in B$ *and* $(B, y) \in D\}$*, and* $R' = \{(\{x\}, x') \mid x' \in X'\} \cup \{(G \cup \{x\}, y) \mid (x, y) \in R, G \subseteq \bigcup gdatt'(x, y)$ *s.t.* $\forall C \in gdatt'(x, y), G \cap C \neq \emptyset\}$*, where* $gdatt'(x, y) = \{\{c'_1, ...c'_n\} \mid (\{c_1, ..., c_n\}, (x, y)) \in D\}$*.*

Redefinition of Theorem 8.12: *Let* $\sigma \in \{$*admissible, preferred, complete, grounded, stable*$\}$ *be a semantics and* SC^{Tr}_{σ} *the removal casting functions for* σ *defined as* $SC^{X}_{\sigma}(E) = E \cap A$*, where* $X = (A, R, D) \in BH^{EAFC}$ *is a bounded hierarchical EAFC and* $E \in \sigma(Tr^{EAFC}_{SETAF}(X))$*. The translation* Tr^{EAFC}_{SETAF} *is strong under* $(\sigma, SC^{Tr}_{\sigma})$*. It is* $\subseteq$*–weak under the conflict–free semantics and the defined casting function. It is semantics bijective under complete, grounded, preferred and stable semantics.*

A lot of the properties carry over from the EAF–SETAF translations. The only difference is that our approach is now overlapping due to the reasons explained before:

Analysis of Translation 41: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and union casting functions, the translation Tr^{EAF}_{SETAF} is:

- source–subclass, target–subclass and overlapping
- weakly argument domain altering, argument introducing, induced attack relation introducing

- generic and weakly semantics domain altering
- structural

The translation Tr^{EAFC}_{SETAF} is not modular. It is faithful under the complete, preferred, grounded and stable semantics and the defined removal casting functions. We classify Tr^{EAFC}_{SETAF} as a basic under the listed semantics and casting functions.

8.3.2 Improvements

The EAF(C)–SETAF translations are faithful for semantics that are also complete. Unfortunately, they are both source–subclass and lack modularity. In order to obtain a faithful and modular translation, we can simply chain the EAF(C)–AF and AF–SETAF approaches. However, our aim was to show how the group attack can handle defense attacks without forcing the presence of arguments representing conflicts, rather than to create another faithful and modular translation. Therefore, we will limit ourselves to the discussion on whether full and exact translations are possible from EAF(C)s to SETAFs.

Unfortunately, the results echo our conclusions for AFs. SETAF complete extensions also form a complete semilattice and the stable extensions need to be incomparable (see Theorems 2.23 and 2.24). Moreover, even though the signatures for SETAF semantics are not yet known, it can be easily shown that if a given set of arguments is conflict–free, then so is any of its subsets. Consequently, we can repeat the analysis done in Section 8.2.2, which brings us to the following results:

Theorem 8.13. *Let Fr^{EAF} be the collection of all EAFs on a domain $\mathcal{U}^{EAF}$ and Fr^{SETAF} the collection of all SETAFs on a domain $\mathcal{U}^{SETAF}$. There does not exist a full translation from Fr^{EAF} to Fr^{SETAF} that is exact under conflict–free, complete and stable semantics and their identity casting functions.*

Theorem 8.14. *Let Fr^{EAFC} be the collection of all EAFCs on a domain $\mathcal{U}^{EAFC}$ and Fr^{SETAF} the collection of all AFs on a domain $\mathcal{U}^{SETAF}$. There does not exist a full translation from Fr^{EAFC} to Fr^{SETAF} that is exact under conflict–free, complete and stable semantics and their identity casting functions.*

8.4 EAF as AFRA

Structurally speaking, the AFRA framework permits deeper recursion than EAFs do. Consequently, if $EF = (A, R, D)$ is an EAF, then $FR = (A, R \cup D)$ is an AFRA. However, as observed in [9] and in Section 7.2, the two frameworks have a different approach towards the semantics:

Example 97. Let us recall the framework $EF_1 = (\{a, b, c, d\}, \{(a, b), (b, a), (c, d), (d, c)\}, \{(a, (d, c)), (b, (c, d)), (c, (b, a)), (d, (a, b))\})$ described in Example 5 and [62]. For convenience, we depict it again in Figure 74. According to Definition 2.42, the set $\{a, b, c, d\}$ is not conflict–free in EF_1. However, both $\{a, b, c, d\}$

and $\{a, b, c, d, (a, (d, c)), (b, (c, d)), (c, (b, a)), (d, (a, b))\}$ are conflict–free in $FR_1 = (\{a, b, c, d\}, \{(a, b), (b, a), (c, d), (d, c), (a, (d, c)), (b, (c, d)), (c, (b, a)), (d, (a, b))\})$. The latter extension is even admissible, preferred and stable in FR_1, while it clearly has no corresponding extension in EF_1.

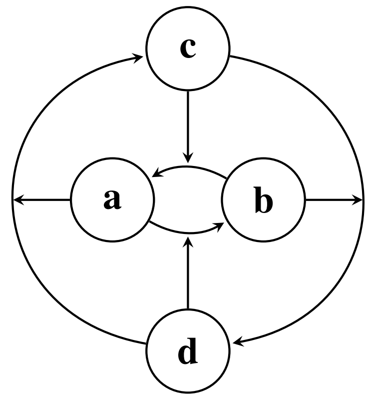


Figure 74: Sample EAF

The issue with conflict–freeness can be considered negligible, particularly since the EAFC generalization of EAFs (see Section 2.1.4.3) relaxes the definition of this semantics. It is also the main reason for the lack of correspondence w.r.t. admissible, preferred and stable semantics. However, the complete and grounded semantics behave differently even when we put conflict–freeness aside.

Example 98 (Taken from [9]). Let us consider the framework $EF_2 = (\{a, b, c\}, \{(b, a), (c, b)\}, \{(b, (c, b))\})$ previously analyzed in Example 94. The sets $\{a, c\}$ and $\{b, c\}$ are complete extensions of EF_2. The set $\{a, c\}$ is also the grounded extension. However, the complete extensions of the corresponding AFRA $FR_2 = (\{a, b, c\}, \{(b, a), (c, b), (b, (c, b))\})$ are $\{c\}$, $\{b, c, (b, a), (b, (c, b))\}$ and $\{a, c, (c, b)\}$, with $\{c\}$ being the grounded one.

The conclusion is that for any EAF semantics, we can find a framework s.t. the AFRA extensions will not correspond to the original sets. The design of the frameworks is very different, even if, structurally speaking, they are not far apart. In this section we will complete the analysis done in [9] and analyze an EAF–AFRA translation which, like in the AF case, its restricted to bounded hierarchical EAFs. In this case the monotonicity of the characteristic operator is retrieved and the stratification of the framework prevents issues such as self–reinstatement.

Translation 42. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF. Its corresponding AFRA is $FR^{EF} = (A, R \cup D)$.*

We can observe that not every type of AFRA can be created with this translation. The produced framework only has none to single recursion of attacks, i.e. attacks can be carried out only on arguments or on attacks targeted at arguments. Consequently, FR^{EF} belongs to the subclass $Rec_0^{AFRA} \cup Rec_1^{AFRA}$. Please note this is not the most accurate description. Due to the fact that the source EAFs are bounded hierarchical, the resulting AFRAs can also be separated into certain levels. Nevertheless, we will not focus on analyzing stratified AFRAs. Let us now continue with the analysis of the semantics. Unfortunately, conflict–freeness and admissibility are preserved only one–way:

Theorem 8.15. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF and $FR^{EF} = (A, R \cup D)$ its corresponding AFRA obtained through Translation 42. If $E \subseteq A$ is a σ–extension of $bh - EF$, where $\sigma \in \{$conflict–free, admissible, complete, grounded, preferred, stable$\}$, then there is a σ–extension $E' \subseteq (A \cup R \cup D)$ of FR^{EF} s.t. $E = E' \cap A$. If $E' \subseteq A \cup R \cup D$ is a σ'–extension of FR^{EF}, where $\sigma' \in \{$complete, grounded, preferred, stable$\}$, then $E = E' \cap A$ is a σ'–extension of $bh - EF$. This does not necessarily hold for conflict–free and admissible semantics.*

We can now redefine these results and put them into our system:
Redefinition of Translation 42: *Let BH^{EAF} be the collection of all bounded hierarchical EAFs and $Rec_0^{AFRA} \cup Rec_1^{AFRA}$ the collection of zero or single–recursion AFRAs, both based on domain $\mathcal{U}$. The translation $Tr_{AFRA}^{EAF} : BH^{EAF} \to (Rec_0^{AFRA} \cup Rec_1^{AFRA})$ is defined as $Tr_{AFRA}^{EAF}((A, R, D)) = (A, R \cup D)$ for a framework $(A, R, D) \in BH^{EAF}$.*
Redefinition of Theorem 8.15: *Let $\sigma \in \{$complete, preferred, grounded, stable$\}$ be a semantics and SC_σ^{Tr} the removal casting functions for σ defined as $SC_\sigma^X(E) = E \cap A$, where $X = (A, R, D) \in BH^{EAF}$ is a bounded hierarchical EAF and $E \in \sigma(Tr_{AFRA}^{EAF}(X))$. The translation Tr_{AFRA}^{EAF} is strong under (σ, SC_σ^{Tr}). It is $\subseteq$–weak under the conflict–free and admissible semantics and the defined casting function. It is semantics bijective under complete, grounded, preferred and stable semantics.*

The presented translation is quite straightforward and thus we will omit further explanations.
Analysis of Translation 42: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and union casting functions, the translation Tr_{AFRA}^{EAF} is:

- source–subclass, target–subclass and injective
- argument domain and structure preserving
- generic and semantics domain altering
- structural and modular

The translation Tr_{AFRA}^{EAF} is faithful under the complete, preferred, grounded and stable semantics and the defined removal casting functions. We classify Tr_{AFRA}^{EAF} as a basic translation under the listed semantics and casting functions.

Please note that AFRAs do not permit conflicts from sets of arguments. Consequently, the most reasonable way to translate an EAFC into an AFRA is to use the EAF bypass. Please note than unlike in the EAFC–AF translation, simple chaining is sufficient.

8.4.1 Improvements

It is easy to see that due to the nature of the AFRA semantics, an exact translation from EAFs to AFRAs is not possible, even when we assume that the source framework is bounded hierarchical. AFRA extensions will consist of arguments only if there are no conflicts in the structure to start with, which is possible only for very particular EAFs or unique status semantics. Similar results hold for EAFCs.

Theorem 8.16. *Let Fr^{EAF} be the collection of all EAFs on a domain $\mathcal{U}^{EAF}$ and Fr^{AFRA} the collection of all AFRAs on a domain $\mathcal{U}^{AFRA}$. There is no full translation from Fr^{EAF} to Fr^{AFRA} that is exact under conflict–free, admissible, complete, preferred and stable semantics and their identity casting functions. Let BH^{EAF} be the collection of all bounded hierarchical EAFs on a domain $\mathcal{U}^{EAF}$. There is no full translation from BH^{EAF} to Fr^{AFRA} that is exact under conflict–free, admissible, complete, preferred and stable semantics and their identity casting functions.*

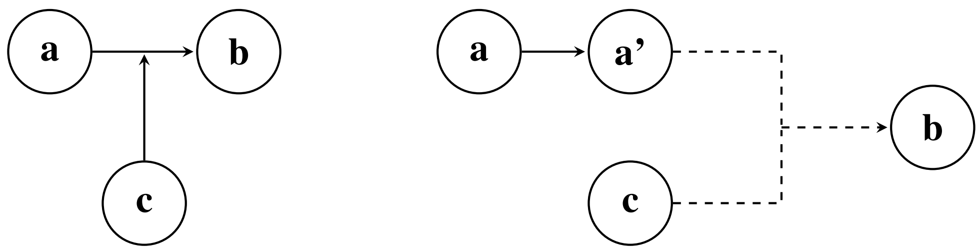
Theorem 8.17. *Let Fr^{EAFC} be the collection of all EAFCs on a domain $\mathcal{U}^{EAFC}$ and Fr^{AFRA} the collection of all AFRAs on a domain $\mathcal{U}^{AFRA}$. There is no full translation from Fr^{EAFC} to Fr^{AFRA} that is exact under conflict–free, admissible, complete, preferred and stable semantics and their identity casting functions. Let BH^{EAFC} be the collection of all bounded hierarchical EAFCs on a domain $\mathcal{U}^{EAFC}$. There is no full translation from BH^{EAFC} to Fr^{AFRA} that is exact under conflict–free, admissible, complete, preferred and stable semantics and their identity casting functions.*

The only possible improvement we can consider relates to the strength of our translation under the admissible semantics. A strong relation can be achieved by adjusting the semantics casting function in the same way as done in Section 5.2.1. Nevertheless, it is not sufficient for faithfulness, and we are not convinced this property can be in fact achieved.

8.5 EAF as AFN

There are two ways we can transform an EAF into an AFN. The first one follows the same principles as the AFRA–AFN approach, where conflicts become new arguments and are connected to their sources through support. However, another approach can be seen as more interesting, particularly when we try to compare the defense attacks to different forms of support (more on this in Section 8.6.1).

Example 99. Let us assume a simple framework $EF_1 = (\{a, b, c\}, \{(a, b)\}, \{(c, (a, b))\})$ visible in Figure 75a. The set $\{b\}$ is a conflict–free extension, i.e. the argument can stand on its own. However, it requires the presence of c in order to be accepted whenever a is around. In other words, while $\{a, b\}$ is not conflict–free, $\{a, b, c\}$ is. The same situation occurs in an AFN in which c is set as a supporter of b and a indirectly attacks b through another supporting argument, as seen in Figure 75b. The framework



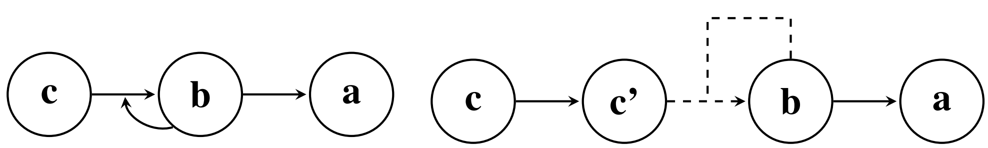
(a) EF_1 (b) Corresponding FN_1

Figure 75: Sample EAF and its corresponding AFN

$FN_1 = (\{a, a', b, c\}, \{(a, a')\}, \{(\{a', c\}, b)\})$ gives us the following strongly coherent extensions: $\emptyset$, $\{a\}$, $\{a'\}$, $\{c\}$, $\{a, c\}$, $\{a', c\}$, $\{a', b\}$, $\{a', b, c\}$, $\{a, b, c\}$. We can observe that if we remove the auxiliary argument a', we retrieve the conflict–free extensions of EF_1.

Therefore, as observed in the example, the defense attack can be seen as a particular form of necessary group support, when accompanied by an additional argument (in our case, a'). This auxiliary argument can be read similarly as in the EAF–AF translation – either as *a is rejected*, or, perhaps more accurately, as *the attack from a is not in force*. We can observe that in a certain sense, the presented construction is dual to the EAF–SETAF approach. Previously, it was the positive defense attack relation that was simulated with combination of conflicts, and in our case it is the negative direct attack relation that is simulated with a mix of conflict and support. Unfortunately, not every EAF can undergo such a translation – the resulting AFN can contain support cycles and produce only some of the extensions of the original structure. Therefore, we will need to restrict ourselves to bounded hierarchical EAFs also in this case.

Example 100. Let us come to the framework $EF_2 = (\{a, b, c\}, \{(b, a), (c, b)\}, \{(b, (c, b))\})$ depicted in Figure 76a and analyzed in Examples 94 and 98. The corresponding AFN $FN_2 = (\{a, b, c, c'\}, \{(c, c'), (b, c)\}, \{(\{b, c'\}, b)\})$ is visible in Figure 76b. The sets $\{a, c\}$ and $\{b, c\}$ are complete extensions of EF_2, with $\{a, c\}$ being the grounded one. Both of the complete extensions are also preferred and stable. However, FN_2 has only one complete extension, namely $\{a, c\}$, which is also the grounded, preferred and stable extension. Argument b has only one powerful sequence (c', b) that is attacked by c. Unlike in EAFs, b cannot reinstate itself in AFN semantics, and not all of the source extensions are retrieved.



(a) EF_2 (b) Corresponding FN_2

Figure 76: EAF with a self–reinstating argument and its corresponding AFN

We can now formally introduce our EAF–AFN translation. Given an EAF (A, R, D), with $R_D \subseteq R$ we will denote the set of those attacks, for which there exists a defense attack, i.e. $R_D = \{(x, y) \mid (x, y) \in R, \exists c \in A \text{ s.t. } (c,(x, y)) \in D\}$. The set of arguments is extended with primed version of arguments that carry out such attacks in R_D. Moreover, the original and primed arguments are then connected with conflict. The attacks in R_D are then removed from the framework and transformed into support along with the conflicts in D:

Translation 43. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF. Its corresponding AFN is $FN^{EF} = (A', R', N')$, where:*

- $A' = A \cup X'$*, where* $X' = \{x' \mid x \in A, \exists y \in A \text{ s.t. } (x, y) \in R_D\}$,
- $R' = (R \setminus R_D) \cup \{(x, x') \mid x' \in X'\}$*, and*
- $N' = \{(datt(x, y) \cup \{x'\}, y) \mid (x, y) \in R_D\}$*, where* $datt(x, y) = \{c \mid (c,(x, y)) \in D\}$.

We can observe that the produced AFN exhibits a number of desirable properties. In particular, it meets all of the introduced normal form requirements, thus producing a well–structured AFN. However, a given support set can contain more than one argument. Therefore, the target AFNs will not be support binary and thus not elementary.

Theorem 8.18. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF and $FN^{EF} = (A', R', N')$ its corresponding AFN obtained through Translation 43. FN^{EF} is (strongly) consistent, minimal, weakly, relation and strongly valid.*

Let us now prove how semantics behave after the translation. Due to the fact that a defense attack argument does not always need to be accepted along with the arguments carrying it out, conflict–freeness is preserved only one–way.

Theorem 8.19. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF and $FN^{EF} = (A', R', N')$ its corresponding AFN obtained through Translation 43. If $E \subseteq A$ is a conflict–free extension of $bh - EF$, then $E' = E \cup \{x' \mid x' \in X, x \in A \setminus E\}$ is strongly coherent in FN^{EF}. If $E \subseteq A$ is a σ–extension of $bh - EF$, where $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$, then $E' = E \cup \{x' \mid x' \in X, x \in E^+\}$ is a σ–extension of FN^{EF}. If $E' \subseteq A'$ is a σ'–extension of FN^{EF}, where $\sigma' \in \{$admissible, complete, preferred, grounded, stable$\}$, then $E' \cap A$ is a σ'–extension of $bh - EF$. This does not necessarily hold for conflict–free semantics.*

We can now redefine and analyze our results:
Redefinition of Translation 43: *Let BH^{EAF} be the collection of all bounded hierarchical EAFs on domain $\mathcal{U}$ and WSt^{AFN} the collection of all well–structured AFNs on domain $\mathcal{U} \cup \mathcal{U}'$. The translation $Tr^{EAF}_{AFN} : BH^{EAF} \rightarrow WSt^{AFN}$ is defined as $Tr^{EAF}_{AFN}((A, R, D)) = (A', R', N')$ for a framework $(A, R, D) \in BH^{EAF}$, where:*

- $A' = A \cup X'$, *where* $X' = \{x' \mid x \in A, \exists y \in A$ *s.t.* $(x,y) \in R_D\}$ *and* $R_D = \{(x,y) \mid (x,y) \in R, \exists c \in A$ *s.t.* $(c,(x,y)) \in D\}$,
- $R' = (R \setminus R_D) \cup \{(x,x') \mid x' \in X'\}$, *and*
- $N' = \{(datt(x,y) \cup \{x'\}, y) \mid (x,y) \in R_D\}$, *where* $datt(x,y) = \{c \mid (c,(x,y)) \in D\}$.

Redefinition of Theorem 8.19: *Let* $\sigma \in \{$*admissible, preferred, complete, grounded, stable*$\}$ *be a semantics and* SC_σ^{Tr} *the removal casting functions for* σ *defined as* $SC_\sigma^X(E) = E \cap A$, *where* $X = (A,R,D) \in BH^{EAF}$ *and* $E \in \sigma(Tr_{AFN}^{EAF}(X))$. *The translation* Tr_{AFN}^{EAF} *is strong under* (σ, SC_σ^{Tr}). *It is* $\subseteq$*–weak under conflict–free semantics and removal casting functions. For complete, preferred, grounded and stable semantics, the translation is semantics bijective.*

Analysis of Translation 43: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and removal casting functions, the translation Tr_{AFN}^{EAF} is:

- source–subclass, target–subclass and injective
- weakly argument domain altering, argument introducing, induced attack introducing, support introducing
- generic and weakly semantics domain altering
- semi–structural

Translation Tr_{AFN}^{EAF} is not modular. Under the complete, preferred, grounded and stable semantics and removal casting functions, Tr_{AFN}^{EAF} is faithful. Translation Tr_{AFN}^{EAF} is classified as basic under the listed semantics and casting functions.

Explanation. It is easy to see that our translation is both source and target–subclass. However, it is also injective. Although the new set of arguments does not necessarily uniquely define the source EAF, the attack set retrieves part of the original one. Then, the support sets precisely correspond to the defense attacks and the primed argument shows which element was the direct attacker. Therefore, we can reconstruct the original structure.

We can observe that the translation Tr_{AFN}^{EAF} performs a number of modifications of the source EAF. Both the argument and semantics domain are altered. The primed arguments are introduced and along with them the relevant attacks. The translation can make a direct attacker become an indirect one (i.e. one that attacks the support of an argument). However, the conflict is preserved. The defense attacks are also represented in the framework, though more as supporters. Nevertheless, since the translation is injective, we do not lose data. Finally, the translation is support introducing - for example, the newly introduced arguments need to be connected to the targets.

Due to the amount of the handled semantics (in a strong manner), we classify the translation as generic. Furthermore, we also consider it a semi–structural approach. We use the indirect attacks in AFNs and the fact that an argument does not have to be rejected as long as there exists a suitable coherent set for it, in order to simulate the behavior of conflicts that can themselves be attacked.

Our translation is unfortunately not modular. Let us consider two frameworks $EF_1 = (\{a, b\}, \{(a, b)\}, \emptyset)$ and $EF_2 = (\{a, b, c\}, \{(a, b)\}, \{(c, (a, b))\})$. Their union is simply EF_2 and its corresponding FN_2 is $(\{a, a', b, c\}, \{(a, a')\}, \{(\{a', c\}, b)\})$, as seen in Example 99. However, the AFN related to EF_1 is $FN_1 = (\{a, b\}, \{(a, b)\}, \emptyset)$ (there is no defense attack, so no support relation is introduced). We can observe that $FN_1 \cup FN_2$ is not the same as FN_2 – the set of attacks is different. Therefore, our approach is not modular. ■

The other EAF–AFN translation can be seen as a chain of EAF–AFRA and AFRA–AFN approaches. It introduces both the direct and defense attacks as new arguments and connects them to their sources via support. Although normally we would not focus on the analysis of a chained translation, this one has an interesting property that none of the translations building it do. Namely, both EAF–AFRA and AFRA–AFN transformations have problems with the admissible semantics – one created too many extensions, the other – not enough. However, chaining them nullifies this effect.

Translation 44. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF. Its corresponding AFN is $FN^{EF} = (A', R', N')$, where:*

- $A' = A \cup R \cup D$,
- $R' = \{((a, b), b) \mid (a, b) \in R, b \in A\} \cup \{((c, (a, b)), (a, b)) \mid (c, (a, b)) \in D, (a, b) \in R\}$*, and*
- $N' = \{(\{a\}, (a, b)) \mid (a, b) \in R, a \in A\} \cup \{(\{c\}, (c, (a, b))) \mid (c, (a, b)) \in D, c \in A\}$.

We can observe that the resulting AFN is, structurally speaking, the same as in the AFRA–AFN case (see Theorem 7.4). The only difference between the approaches is the depth of the attacks represented by the attack arguments, which is not relevant for normal forms.

Theorem 8.20. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF and $FN^{EF} = (A', R', N')$ its corresponding AFN obtained through Translation 44. Then FN^{EF} is (strongly) consistent, minimal, weakly, relation and strongly valid.*

The constructed AFNs are moreover support binary and of support depth 1. This puts our framework in the class of well–structured and elementary AFNs. We can now move on to the semantics. As noted before, the admissible semantics will be preserved by the translation:

Theorem 8.21. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF and $FN^{EF} = (A', R', N')$ its corresponding AFN obtained through Translation 44. If a set $E \subseteq A$ is a σ–extension of $bh - EF$, where $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$, then there is a σ–extension $E' \subseteq A'$ of FN^{EF} s.t. $E' \cap A = E$. If a set $E' \subseteq A'$ is a σ'–extension of FN^{EF}, where $\sigma' \in \{$admissible, complete, preferred, grounded, stable$\}$, then $E = E' \cap A$ is a σ'–extension of $bh - EF$.*

We can now put this translation into the system and analyze its properties. In order to differentiate between this approach and the previous one, we will prefix it with $c-$ standing for chained.

Redefinition of Translation 44: *Let BH^{EAF} be the collection of all bounded hierarchical EAFs on domain $\mathcal{U}$ and $WSt^{AFN} \cap SEle_1^{AFN}$ the collection of all well–structured and elementary AFNs of support depth 1 on domain $\bigcup_{i=1}^{3} \mathcal{U}^i$, where $\mathcal{U}^1 = \mathcal{U}$ and $\mathcal{U}^i = \mathcal{U} \times \mathcal{U}^{i-1}$ for $i > 1$. The translation $c\text{-}Tr_{AFN}^{EAF} : BH^{EAF} \rightarrow WSt^{AFN} \cap SEle_1^{AFN}$ is defined as $c\text{-}Tr_{AFN}^{EAF}((A, R, D)) = (A', R', N')$ for a framework $(A, R, D) \in BH^{EAF}$, where:*

- $A' = A \cup R \cup D$,
- $R' = \{((a, b), b) \mid (a, b) \in R, b \in A\} \cup \{((c, (a, b)), (a, b)) \mid (c, (a, b)) \in D, (a, b) \in R\}$, *and*
- $N' = \{(\{a\}, (a, b)) \mid (a, b) \in R, a \in A\} \cup \{(\{c\}, (c, (a, b))) \mid (c, (a, b)) \in D, c \in A\}$.

Redefinition of Theorem 8.21: *Let $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC_{σ}^{Tr} the removal casting functions for σ defined as $SC_{\sigma}^{X}(E) = E \cap A$, where $X = (A, R, D) \in BH^{EAF}$ and $E \in \sigma(c\text{-}Tr_{AFN}^{EAF}(X))$. The translation $c\text{-}Tr_{AFN}^{EAF}$ is strong under $(\sigma, SC_{\sigma}^{Tr})$. It is $\subseteq$–weak under conflict–free semantics and removal casting functions. For complete, preferred, grounded and stable semantics, the translation is semantics bijective.*

The properties of our translation are simply a result of properties of EAF–AFRA and AFRA–AFN and thus we will omit the explanations.

Analysis of Translation 44: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and removal casting functions, the translation $c\text{-}Tr_{AFN}^{EAF}$ is:

- source–subclass, target–subclass and injective
- argument domain altering, argument introducing, attack relation preserving, support relation introducing
- generic and semantics domain altering
- semi–structural and modular

Under the complete, preferred, grounded and stable semantics and removal casting functions, $c\text{-}Tr^{EAF}_{AFN}$ is faithful. Translation $c\text{-}Tr^{EAF}_{AFN}$ is classified as basic under the listed semantics and casting functions.

Another reason why we decided to introduce both of these approaches is their relation to the EAF–AF translations, both to the one we analyzed [64], and to the one we briefly recalled [18] (see Figure 78). When we translated EAFs to AFs, defense was used e.g. to tie a defense attack argument to the argument actually carrying out the attack. In the defender AFN–SETAF translation, which will be described in Section 10.2.2, defense was used to tie a supported argument to its supporter in a similar fashion. Consequently, we can decide to modify the EAF–AF approach by reverting the defender transformation. There are now two ways to do this; firstly, we can remove the primed arguments and introduced support links between conflicts and their sources, as seen in Figure 77 for the [64]. However, it is also the primed arguments we can decide to keep – both primed arguments and those that carry out a defense attacks are in fact the defenders of the target of the conflict in question. Consequently, it is them we can transform into group supporters, as performed in Figure 78 on the [18] approach.

We can now observe that reverting defense to support in fact gave us both of our EAF–AFN translations, which is an interesting reminder that a given translation can be obtained by starting from various different perspectives and that while usually it is the support that is replaced by attacks, the other way around is also possible.

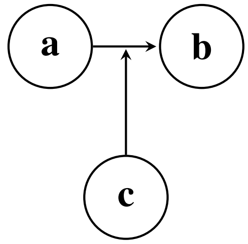
8.5.1 EAFC as AFN

In order to translate EAFCs to AFNs, we can quite straightforwardly adapt the EAF–AFN translations. The changes done to Translation 43 are similar to the ones done to the EAF–SETAF approach in order to make it suitable for EAFCs. We now need to ensure that every set supporting a given argument contains at at least one argument from every defense attacking set (see Figure 79). Due to the way support works in AFNs, this ensures that an argument can be accepted only if a relevant primed argument or a whole defense attacking set is present:

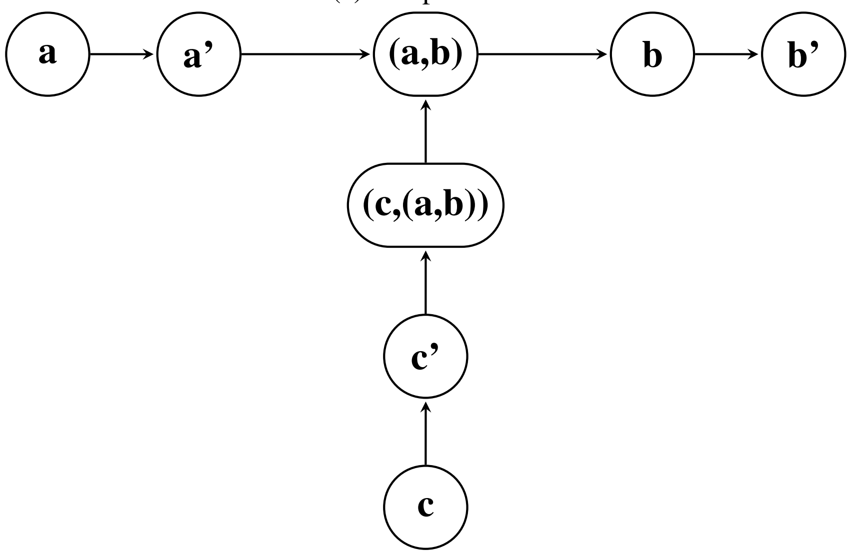
Translation 45. *Let $bh - EFC = (A, R, D)$ be a bounded hierarchical EAFC. Its corresponding AFN is $FN^{EFC} = (A', R', N')$, where:*

- $A' = A \cup X'$, *where* $X' = \{x' \mid x \in A, \exists y \in A$ *s.t.* $(x, y) \in R_D\}$,
- $R' = (R \setminus R_D) \cup \{(x, x') \mid x' \in X'\}$, *and*
- $N' = \{G \cup \{x'\}, y) \mid (x, y) \in R_D, G \subseteq \bigcup gdatt(x, y)$ *s.t.* $\forall C \in gdatt(x, y), G \cap C \neq \emptyset\}$, *where* $gdatt(x, y) = \{C \mid (C, (x, y)) \in D\}$.

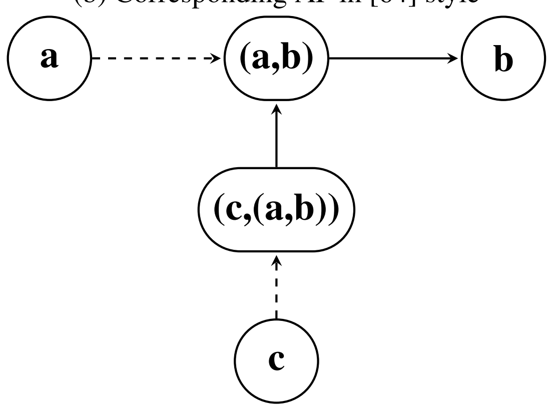
Please note that the current construction of the supporting sets introduces certain redundant relations in the target AFN. Thus, as explained in Section 8.3.1, the choice on whether to restrict ourselves to minimal sets is left to the reader. Nevertheless, aside from minimality, the results presented in Theorem 8.18 hold in EAFCs as well:



(a) Sample EAF



(b) Corresponding AF in [64] style

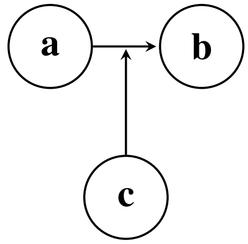


(c) AFN version of the meta–level AF with support replacing defense

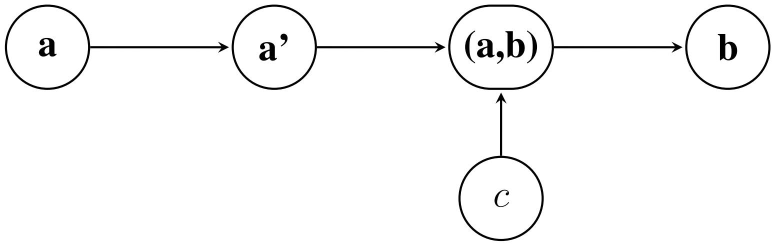
Figure 77: Sample EAF, its [64] meta–level AF and AFN with defense replacement

Theorem 8.22. *Let $bh - EFC = (A, R, D)$ be a bounded hierarchical EAFC and $FN^{EFC} = (A', R', N')$ its corresponding AFN obtained through Translation 45. FN^{EFC} is strongly consistent, weakly, relation and strongly valid.*

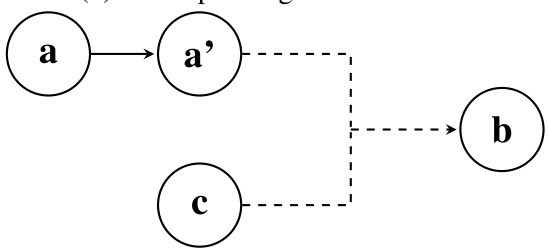
Let us now focus on the semantics. We can easily adapt the proof of Theorem 8.19 in



(a) Sample EAF

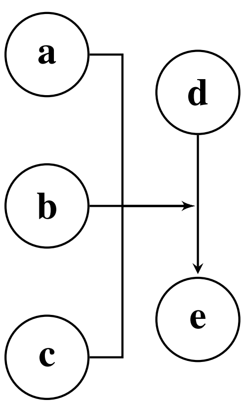


(b) Corresponding meta–level AF

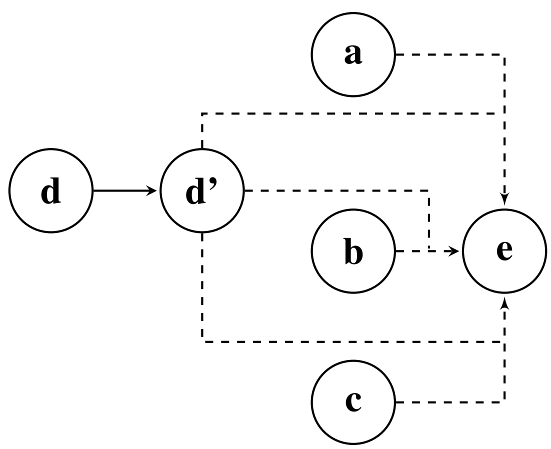


(c) AFN version of the meta–level AF with support replacing defense

Figure 78: Sample EAF and its [18] meta–level AF and AFN with support replacement



(a) Sample EAFC



(b) Related AFN

Figure 79: Sample EAFC and related AFN

order to show that the following holds:

Theorem 8.23. *Let $bh - EFC = (A, R, D)$ be a bounded hierarchical EAFC and $FN^{EF} = (A', R', N')$ its corresponding AFN obtained through Translation 45. If $E \subseteq A$ is a conflict–free extension of $bh - EFC$, then $E' = E \cup \{x' \mid x' \in X, x \in A \setminus E\}$ is strongly coherent in FN^{EFC}. If $E \subseteq A$ is a σ–extension of $bh - EFC$, where $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$, then $E' = E \cup \{x' \mid x' \in X, x \in E^+\}$ is a σ–extension of FN^{EFC}. If $E' \subseteq A'$ is a σ'–extension of FN^{EFC}, where $\sigma' \in \{$admissible, complete, preferred, grounded, stable$\}$, then $E' \cap A$ is a σ'–extension of $bh - EFC$. This does not necessarily hold for conflict–free semantics.*

The translation is entered into our system similarly to Translation 43.

Redefinition of Translation 45: *Let BH^{EAFC} be the collection of all bounded hierarchical EAFCs on domain $\mathcal{U}$ and WSt^{AFN} the collection of all well–structured AFNs on domain $\mathcal{U} \cup \mathcal{U}'$. The translation $Tr^{EAFC}_{AFN} : BH^{EAFC} \rightarrow WSt^{AFN}$ is defined as $Tr^{EAFC}_{AFN}((A, R, D)) = (A', R', N')$ for a framework $(A, R, D) \in BH^{EAFC}$, where:*

- *$A' = A \cup X'$, where $X' = \{x' \mid x \in A, \exists y \in A$ s.t. $(x, y) \in R_D\}$,*
- *$R' = (R \setminus R_D) \cup \{(x, x') \mid x' \in X'\}$, and*
- *$N' = \{G \cup \{x'\}, y) \mid (x, y) \in R_D, G \subseteq \bigcup gdatt(x, y)$ s.t. $\forall C \in gdatt(x, y), G \cap C \neq \emptyset\}$, where $gdatt(x, y) = \{C \mid (C, (x, y)) \in D\}$.*

Redefinition of Theorem 8.23: *Let $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the removal casting functions for σ defined as $SC^X_{\sigma}(E) = E \cap A$, where $X = (A, R, D) \in BH^{EAFC}$ and $E \in \sigma(Tr^{EAFC}_{AFN}(X))$. The translation Tr^{EAFC}_{AFN} is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under conflict–free semantics and removal casting functions. For complete, preferred, grounded and stable semantics, the translation is semantics bijective.*

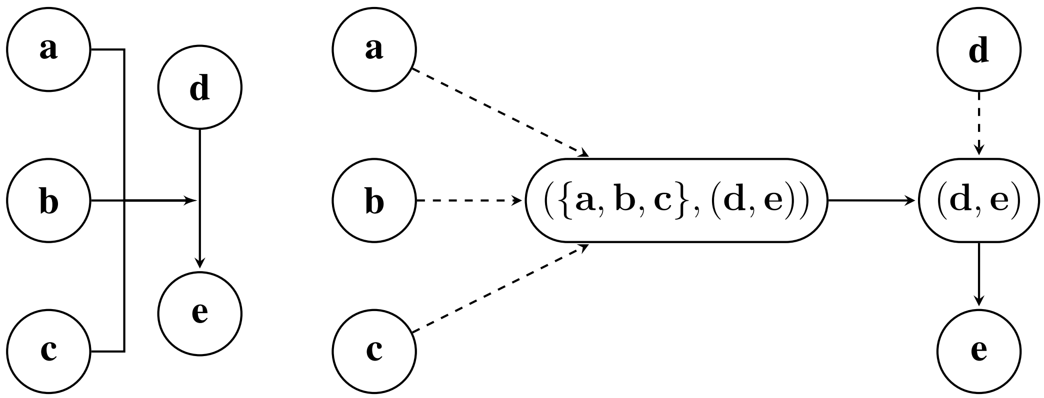
The same analysis as for Translation 43 holds; the only difference is that now, our approach is overlapping, similarly as in Translation 41.

Analysis of Translation 45: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and removal casting functions, the translation Tr^{EAFC}_{AFN} is:

- source–subclass, target–subclass and overlapping
- weakly argument domain altering, argument introducing, induced attack introducing, support introducing
- generic and weakly semantics domain altering
- semi–structural

Translation Tr^{EAFC}_{AFN} is not modular. Under the complete, preferred, grounded and stable semantics and removal casting functions, Tr^{EAFC}_{AFN} is faithful. Translation Tr^{EAFC}_{AFN} is classified as basic under the listed semantics and casting functions.

We can now adapt the other EAF–AFN transformation. The only required modification of Translation 44 is ensuring that all of the arguments taking part in a defense attack support the relevant attack argument, as visible in Figure 80. Similar change, though attack related, was performed e.g. in the EAFC–AF Translation 39.



(a) Sample EAFC

(b) Related AFN

Figure 80: Sample EAFC and related AFN

The translation is now as follows:

Translation 46. *Let $bh - EFC = (A, R, D)$ be a bounded hierarchical EAFC. Its corresponding AFN is $FN^{EFC} = (A', R', N')$, where:*

- $A' = A \cup R \cup D$,
- $R' = \{((a,b),b) \mid (a,b) \in R, b \in A\} \cup \{((C,(a,b)),(a,b)) \mid (C,(a,b)) \in D, (a,b) \in R\}$, *and*
- $N' = \{(\{a\},(a,b)) \mid (a,b) \in R, a \in A\} \cup \{(\{c\},(C,(a,b))) \mid (C,(a,b)) \in D, c \in C\}$.

We can observe that even though our AFN is no longer singular, it satisfies the same normal forms as in the EAF–AFN case (see Theorem 8.20). Moreover, the proof Theorem 8.21 and the analysis of Translation 44 can also be easily adapted to the EAFC–AFN case. Thus, we will omit further explanations.

Theorem 8.24. *Let $bh - EFC = (A, R, D)$ be a bounded hierarchical EAFC and $FN^{EFC} = (A', R', N')$ its corresponding AFN obtained through Translation 44. FN^{EFC} is strongly consistent, minimal, weakly, relation and strongly valid.*

Theorem 8.25. *Let $bh - EFC = (A, R, D)$ be a bounded hierarchical EAFC and $FN^{EFC} = (A', R', N')$ its corresponding AFN obtained through Translation 46. If a set $E \subseteq A$ is a σ–extension of $bh - EFC$, where $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$, then there is a σ–extension $E' \subseteq A'$ of FN^{EFC} s.t. $E' \cap A = E$. If a set $E' \subseteq A'$ is a σ'–extension of FN^{EFC}, where $\sigma' \in \{$admissible, complete, preferred, grounded, stable$\}$, then $E = E' \cap A$ is a σ'–extension of $bh - EFC$.*

Redefinition of Translation 46: *Let BH^{EAFC} be the collection of all bounded hierarchical EAFCs on domain $\mathcal{U}$ and $WSt^{AFN} \cap SEle_1^{AFN}$ the collection of all well–structured and elementary AFNs of support depth 1 on domain $\bigcup_{i=1}^{3} \mathcal{U}^i$, where $\mathcal{U}^1 = \mathcal{U}$ and $\mathcal{U}^i = \mathcal{U} \times \mathcal{U}^{i-1}$ for $i > 1$. The translation c-$Tr_{AFN}^{EAFC} : BH^{EAFC} \rightarrow WSt^{AFN} \cap SEle_1^{AFN}$ is defined as c-$Tr_{AFN}^{EAFC}((A, R, D)) = (A', R', N')$ for a framework $(A, R, D) \in BH^{EAFC}$, where:*

- $A' = A \cup R \cup D$,
- $R' = \{((a, b), b) \mid (a, b) \in R, b \in A\} \cup \{((c, (a, b)), (a, b)) \mid (c, (a, b)) \in D, (a, b) \in R\}$, *and*
- $N' = \{(\{a\}, (a, b)) \mid (a, b) \in R, a \in A\} \cup \{(\{c\}, (c, (a, b))) \mid (c, (a, b)) \in D, c \in A\}$.

Redefinition of Theorem 8.25: *Let $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC_σ^{Tr} the removal casting functions for σ defined as $SC_\sigma^X(E) = E \cap A$, where $X = (A, R, D) \in BH^{EAFC}$ and $E \in \sigma(c\text{-}Tr_{AFN}^{EAFC}(X))$. The translation c-$Tr_{AFN}^{EAFC}$ is strong under (σ, SC_σ^{Tr}). It is $\subseteq$–weak under the conflict–free semantics and removal casting functions. For complete, preferred, grounded and stable semantics, the translation is semantics bijective.*

Analysis of Translation 46: Under the conflict–free, admissible, preferred, complete, grounded and stable semantics and removal casting functions, the translation $c\text{-}Tr_{AFN}^{EAFC}$ is:

- source–subclass, target–subclass and injective
- argument domain altering, argument introducing, attack relation preserving, support relation introducing
- generic and semantics domain altering
- semi–structural and modular

Under the complete, preferred, grounded and stable semantics and removal casting functions, $c\text{-}Tr_{AFN}^{EAFC}$ is faithful. Translation $c\text{-}Tr_{AFN}^{EAFC}$ is classified as basic under the listed semantics and casting functions.

8.5.2 Improvements

Just like in the EAF(C)–AF case, our translations are faithful for semantics that are at least complete. Moreover, one of them is modular. Unfortunately, even though AFN semantics signatures have not yet been analyzed, we can already say that for some of the semantics, a full and exact EAF(C)–AFN translation is not possible.

We can observe that conflict–free semantics for AFNs are defined in the same way as for AFs and focus solely on the attack relation. Consequently, the conflict–free extensions of an AFN (A, R, N), are the same as of a Dung's framework (A, R). This means that the analysis done in Section 8.2.2 can be repeated here in order to show that no full and exact translation from EAFs to AFNs is possible under the conflict–free semantics.

According to Theorem 2.95, the AFN grounded extension is the least complete one. This means that there cannot exist an AFN that would realize under the complete semantics a collection of extensions that does not contain the least element. As seen in Example 90, the set of complete extensions of a given EAF does not have to have the least element. Hence, a full and exact (under the complete semantics) EAF(C)–AFN translation is also not possible. Due to the fact that EAF stable extension can be comparable w.r.t. $\subseteq$, and AFN ones cannot, the same negative result holds for the stable semantics (see Example 9).

Theorem 8.26. *Let Fr^{EAF} be the collection of all EAFs on a domain $\mathcal{U}^{EAF}$ and Fr^{AFN} the collection of all AFNs on a domain $\mathcal{U}^{AFN}$. There does not exist a full translation from Fr^{EAF} to Fr^{AFN} that is exact under conflict–free, complete and stable semantics and their identity casting functions.*

Theorem 8.27. *Let Fr^{EAFC} be the collection of all EAFCs on a domain $\mathcal{U}^{EAF}$ and Fr^{AFN} the collection of all AFNs on a domain $\mathcal{U}^{AFN}$. There does not exist a full translation from Fr^{EAFC} to Fr^{AFN} that is exact under conflict–free, complete and stable semantics and their identity casting functions.*

For the same reasons as in Section 8.2.2, the question concerning the admissible and preferred semantics and the exactness analysis for bounded hierarchical frameworks are left for future work.

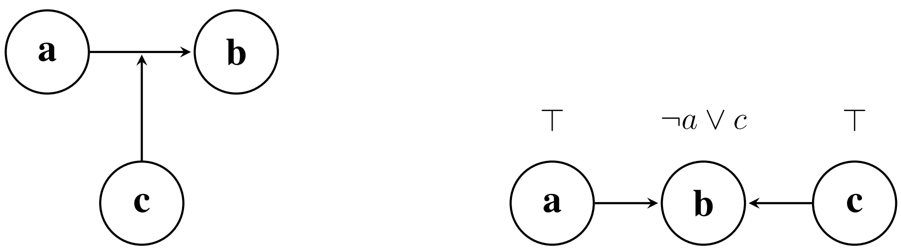
8.6 EAF and EAFC as ADF

We can now proceed with translating EAFs and EAFCs to ADFs. To the best of our knowledge, this is also the only case in which we are not limited to bounded hierarchical frameworks only, even though now we have to face the issue of inconsistency (see Section 4.4.3). We will first describe how defense attacks can be handled by what we informally refer to as the overpowering support. It is a type of a positive relation between arguments that can override incoming conflicts, but itself is not always necessary for acceptance. We will then introduce the EAF translations, first limited for frameworks meeting the EAF–EAFC translation requirements, and then differing between the consistent and general structures. Finally, we introduce the EAFC translations, that are not limited to the frameworks that are bounded hierarchical or without symmetric attacks anymore.

8.6.1 Defense Attack as a Form of Support

In this section we will focus on explaining how the defense attacks can be seen as a form of support and show that certain issues described in Section 2.1.4.2 can be, in fact, seen as attempts at handling support cycles.

Just like defense, defense attack is a type of a positive, indirect relation towards the "defended" argument. The difference is that while in the first case it is also a negative relation towards the argument carrying out the attack, in the latter the attacker and the defense attacker might be unrelated and accepted together in an extension. However, not all interpretations of support can describe this behavior directly. In the abstract case, the presence of the supporter is not required for the acceptance of an argument, which correctly grasps the fact that a defense attacker does not always accompany the defended argument. Unfortunately, if a (direct) attacker appears, so has to an appropriate defense attacker, and thus abstract support is insufficient. The necessary, and thus evidential and deductive supports, have the opposite problem. This is one of the reasons we had to use auxiliary arguments in the EAF–AFN translations. However, the "if attacker is accepted, then accept defense attacker" reasoning can be quite easily expressed with the acceptance conditions in ADFs, as observed in Figure 81. However, most importantly, ADFs possess the family of ca–semantics, which exhibits the same behaviors as EAF semantics and that made exact translations to other frameworks impossible.

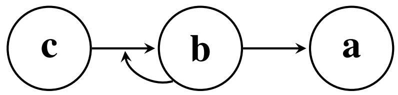


(a) Sample EAF (b) Corresponding ADF

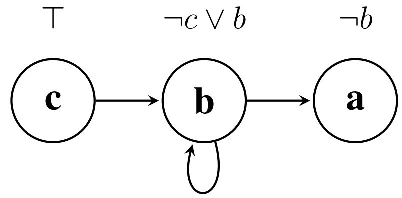
Figure 81: Sample EAF and its corresponding ADF

Example 101. Let us come back to the framework $EF = (\{a, b, c\}, \{(b, a), (c, b)\}, \{(b, (c, b))\})$ depicted in Figure 82a and previously analyzed in Examples 94 and 100. Its corresponding ADF is $D = (\{a, b, c\}, \{C_a = \neg b, C_b = \neg c \vee b, C_c = \top)$ visible in Figure 82b. The sets $\{a, c\}$ and $\{b, c\}$ are complete extensions of EF, with $\{a, c\}$ being the grounded one. Both of the complete extensions are also preferred and stable. The fact that $\{b, c\}$ is an extension shows us that self–reinstatement is sufficient for acceptance. However, as also $\{a, c\}$ is admitted, it also points to the fact that it is not sufficient for defense. This is perfectly grasped by the ca_2–semantics of ADFs. In the D case, $\{a, c\}$ and $\{b, c\}$ are ca_2–complete and ca_2–preferred. Moreover, $\{a, c\}$ is the acyclic grounded extension. It is worth nothing that the stable extensions of EF are not stable extensions of D – the set $\{b, c\}$ is not acyclic. They are however models, which play the same role in ca_2–semantics as stability does in the aa–family.

The fact that defense attacks form what we have previously described as optional or hidden cycles (see Section 4.3.3) can explain some of the issues we have described in Section 2.1.4.2. The symmetric attack restriction of conflict–freeness can be seen as an attempt to deal with defense attack cycles. Furthermore, it is also cycles that made the stable extensions comparable. With the following examples we close this section and move on to the translations.



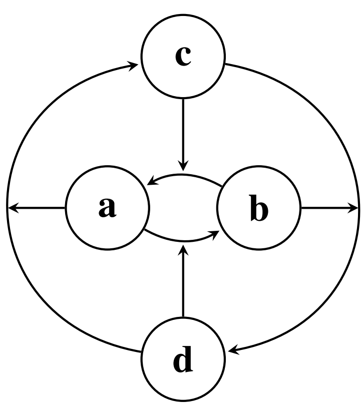
(a) Sample EAF



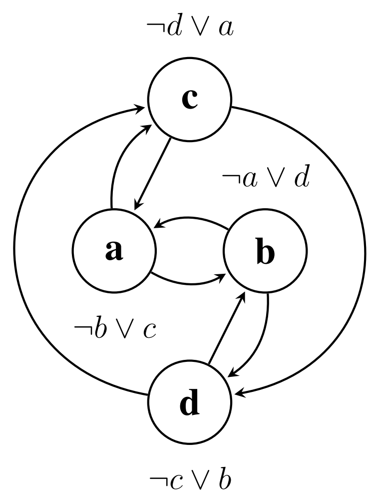
(b) Corresponding ADF

Figure 82: Sample EAF and its corresponding ADF

Example 102. Let us come back to Example 5 and let $(\{a, b, c, d\}, \{(a, b), (b, a), (c, d), (d, c)\}, \{(a, (d, c)), (b, (c, d)), (c, (b, a)), (d, (a, b))\})$ be the EAF depicted in Figure 83a. This framework motivated the conflict–freeness definition that treated symmetric attacks as a special case. The set $\{a, b, c, d\}$ was not considered conflict–free, even though there were no defeats in it. We can observe that this extension activates all the "support" cycles in the framework, which perhaps can be more easily seen in the associated ADF in Figure 83b. b cannot be accepted without d due to the attack from a and d cannot be accepted without b due to the attack from c. Similarly, presence of a in a sense forces and requires the presence of c. Therefore, the restrictive definition of conflict–freeness can be seen as an attempt to limit this behavior.



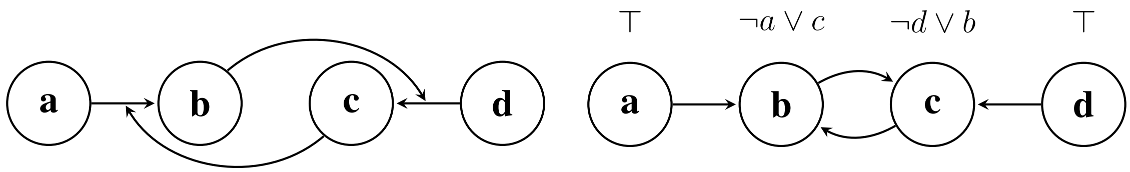
(a) Sample EAF



(b) Corresponding ADF

Figure 83: Sample EAF and corresponding ADF

Example 103. Let us come back to the Example 9 and the EAF $(\{a, b, c, d\}, \{(a, b), (d, c)\}, \{(b, (d, c)), (c, (a, b))\})$ depicted in Figure 84a. Both $\{a, d\}$ and $\{a, b, c, d\}$ are its stable extensions. If we look at the associated ADF depicted in Figure 84b, the optional cycle between b and c might be more visible. It is the reason why $\{a, b, c, d\}$, in which the cycle becomes active, would not be considered e.g. ADF or AFN stable. However, both of the extensions are ADF models.



(a) Sample EAF

(b) Corresponding ADF

Figure 84: Sample EAF and corresponding ADF

8.6.2 EAF as ADF

Just as it will be required in translations from bipolar frameworks to ADFs, in the EAF–ADF approach we need to distinguish between the consistent and not necessarily consistent EAFs. Not doing so can produce unintended extensions, as explained in Sections 2.3.9 and 4.4.3. What we will do is provide one translation specialized at consistent frameworks and the other not making such restrictions. Unfortunately, while right now we can handle frameworks that are not bounded hierarchical, we have a problem arising from the original definition of conflict–freeness. As we have explained in the previous section, the distinction between symmetric attacks could have been viewed as an attempt to deal with support cycles in the extensions. In this sense, the ca_2 semantics – due to the fact that the cycles are accepted in extensions – can produce more extensions than desired. If we look back at Example 102, we can observe that the set $\{a, b, c, d\}$ was not conflict–free in the source, while it was in the associated ADF. However, the set is not pd–acyclic conflict–free. Nevertheless, since this cycle detection in EAFs is not thorough (see Section 2.1.4.2), the (inside) acyclic semantics of ADFs would not be applicable in a variety of other cases. However, the ca_2 family fits perfectly if we assume that no symmetric attacks are present. Therefore, even though we do not claim that a full EAF translation is impossible, we limit ourselves to their subclass.

8.6.2.1 Consistent EAF

The way we transform EAFs into ADFs is quite straightforward and could have already been witnessed in the previous section. For a given attack by b on a, we create a disjunction with a negated literal $\neg b$ representing an attacker and positive literals standing for the defense attacks on the (b, a) conflict. A conjunction of all such formulas gives us the condition for a.

Translation 47. *Let $EF = (A, R, D)$ be an EAF s.t. it is bounded hierarchical, or (strongly) consistent and without symmetric attacks. For a conflict $(b, a) \in R$, let $D_{b,a} = \{c \mid (c, (b, a)) \in D\}$ denote the arguments defense attacking (b, a). The ADF associated with EF is $D^{EFC} = (A, L, C)$, where $L = \{(a, b) \mid (a, b) \in R$ or $\exists x$ s.t. $(a, (x, b)) \in D\}$ and $C = \{C_a \mid a \in A\}$ is the set of acceptance conditions, where every condition C_a is created in the following way:*

- *functional form:*
  - *for a subset of parents $B \subseteq par(a)$, if there exists $x \in B$ s.t. $(x, a) \in R$ and $\nexists b \in B$ s.t. $(b, (x, a)) \in D$, then $C_a(B) = out$; otherwise, $C_a(B) = in$*
- *propositional form:*
  - *let $b \in A$ be an argument s.t. $(b, a) \in R$. The attack formula corresponding to b is $att_a^b = \neg b \vee \bigvee D_{b,a}$. If $D_{b,a}$ is empty, then it is simply $\neg b$.*
  - *the acceptance condition is the conjunction of all such att_a^b parts: $C_a = \bigwedge_{b \in A, (b,a) \in R} att_a^b$. In case a is not attacked at all, it is simply $\top$.*

When we looked at Dung–style frameworks, an acceptance condition was either $\top$ or a conjunction of negated literals. In case of SETAFs, it was $\top$ or conjunction of clauses consisting of negations of arguments. In EAFs, we will have either $\top$ or conjunctions of clauses s.t. there is exactly negated argument in the clause (there are no restrictions on positive ones). Furthermore, we can say that this negation is unique, i.e. our translation cannot produce an expression of the form $(\neg a \vee b) \wedge (\neg a \vee c)$. We will refer to conditions of this type as EAF–style, though please note that due to varying intuitions concerning EAF structure, we will not make any restrictions concerning conflicts induced by the preferences on symmetric attacks. This means that we will refer to an ADF as EAF–style whenever all of the conditions are follow this style and each condition is based only on the incoming relations and not on the conditions of other arguments. Furthermore, the fact that we either assume consistent frameworks or introduce bypass arguments (more on this in the next section) means that no argument appearing as a negative literal in one clause can appear as a positive one in another. Therefore, the class of EAF–style ADFs is somewhat bigger than the class of EAF–produced ADFs.

In this translation we could have observed that no consistency restrictions were put on the bounded hierarchical EAFs. This is because the hierarchical EAFs are already (strongly) consistent (see Lemma 4.67).

Theorem 8.28. *Let $EF = (A, R, D)$ be an EAF s.t. it is bounded hierarchical, or (strongly) consistent and without symmetric attacks. Let $D^{EF} = (A, L, C)$ be its corresponding ADF obtained through Translation 47. D^{EF} is a BADF. It is also in redundancy–free, cleansed, and weakly valid form. It is not necessarily an AADF$^+$ and does not have to be in relation or strongly valid form. If EF is bounded hierarchical, then D^{EF} is an AADF$^+$ and is in relation and strongly valid forms.*

Let us now proceed with the semantics. First of all, we can observe that the conflict–free sets and the (partially acyclic) discarded sets coincide between the source EAF and the target ADF:

Theorem 8.29. *Let EF be an EAF s.t. it is bounded hierarchical, or (strongly) consistent and without symmetric attacks. Let $D^{EF} = (A, L, C)$ be its corresponding ADF obtained through Translation 47. A set of arguments $E \subseteq A$ is a conflict–free extension of EF iff it is conflict–free in D^{EF}.*

Lemma 8.30. *Let EF be an EAF s.t. it is bounded hierarchical, or (strongly) consistent and without symmetric attacks. Let $D^{EF} = (A, C)$ be its corresponding ADF obtained through Translation 47. Let E be a conflict–free extension of EF (and thus of D^{EF}). The discarded set of E in EF coincides with the partially acyclic discarded set of E in D^{EF}.*

Moreover, we can also show that the decisiveness in ADFs and defense in EAFs coincide:

Lemma 8.31. *Let EF be an EAF s.t. it is bounded hierarchical, or (strongly) consistent and without symmetric attacks. Let $D^{EF} = (A, C)$ be its corresponding ADF obtained through Translation 47. A conflict–free set of arguments E defends an argument $a \in A$ in EF iff a is decisively in w.r.t. the partially acyclic range v_E^p of E in D^{EF}.*

Using these partial results, we can show that the admissible, complete and preferred extensions of EAFs that meet our requirements coincide with their ca$_2$–counterparts in ADFs. We can also use Lemma 2.159 to show the correspondence between the stable and model extensions of the two frameworks. Finally, following Definition 2.142 and the approach from [64], we can show the relation between the grounded and acyclic grounded extensions of EAFs and ADFs by starting with the empty set and iteratively expanding it by the arguments defended by it (decisively in w.r.t. its range). This brings us to the following theorem:

Theorem 8.32. *Let EF be an EAF s.t. it is bounded hierarchical, or (strongly) consistent and without symmetric attacks. Let $D^{EF} = (A, C)$ be its corresponding ADF obtained through Translation 47. A set of arguments $E \subseteq A$ is a conflict–free extension of EF iff it is conflict–free in D^{EF}. E is a stable extension of EF iff it is a model of D^{EF}. E is a grounded extension of EF iff it is the acyclic grounded extension of D^{EF}. E is a σ–extension of EF, where $\sigma \in \{$admissible, complete, preferred$\}$, iff it is a ca$_2$–σ–extension of D^{EF}.*

We can now put the results into our system and analyze the properties of our translation. We will prefix it with $con-$ in order to avoid confusion with the general EAF-ADF transformation. Please note that although we will use EAF^{ADF} as the translation codomain, it is not the most accurate description of the EAF–produced ADFs. Due to difference in the naming of the semantics between EAFs and ADFs, we will need to pair them based on similarity (see Definition 3.2).

Redefinition of Translation 47: *Let $BH^{EAF} \cup (Cons^{EAF} \cap NSym^{EAF})$ be the collection of all EAFs that are bounded hierarchical, or (strongly) consistent and without symmetric attacks, and let EAF^{ADF} the collection of all EAF–style ADFs, both on domain $\mathcal{U}$. The translation con-$Tr^{EAF}_{ADF} : (BH^{EAF} \cup (Cons^{EAF} \cap NSym^{EAF})) \rightarrow EAF^{ADF}$ is defined as con-$Tr^{EAF}_{ADF}((A, R, D)) = (A, L, C)$ for a framework $(A, R, D) \in (BH^{EAF} \cup (Cons^{EAF} \cap NSym^{EAF}))$, where $L = \{(a, b) \mid (a, b) \in R$ or $\exists x$ s.t. $(a, (x, b)) \in D\}$ and $C = \{C_a \mid a \in A\}$ is the set of acceptance conditions, where every condition C_a is created in the following way:*

- *functional form: for every subset of parents $B \subseteq par(a)$, if there exists $x \in B$ s.t. $(x, a) \in R$ and $\nexists b \in B$ s.t. $(b, (x, a)) \in D$, then $C_a(B) = out$; otherwise, $C_a(B) = in$*
- *propositional form: if there is no $b \in A$ s.t. $(b, a) \in R$, $C_a = \top$; otherwise, $C_a = \bigwedge_{b \in A, (b,a) \in R} att^b_a$, where $att^b_a = \neg b \vee \bigvee D_{b,a}$ if $D_{b,a} \neq \emptyset$ and $att^b_a = \neg b$ if $D_{b,a} = \emptyset$.*

Redefinition of Theorem 8.32: *Let $\sigma^{EAF} \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$ be an EAF semantics and $\sigma^{ADF} \in \{$conflict–free, ca$_2$–admissible, ca$_2$–preferred, c$_2$–complete, acyclic grounded, model$\}$ be a similar ADF semantics. Let SC^{Tr}_{σ} the identity casting functions for σ. The translation con-Tr^{EAF}_{ADF} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*

Analysis of Translation 47: Under the conflict–free, (ca$_2$–) admissible, (ca$_2$–) preferred, (ca$_2$–) complete, (acyclic) grounded and (model) stable semantics and identity casting functions, the translation *con-*Tr^{EAF}_{ADF} is:

- source–subclass, target–subclass and injective
- argument domain preserving, argument preserving and relation introducing
- generic, semantics domain preserving and exact
- structural

The translation is neither $\oplus$ nor $\otimes$–modular. Translation *con-*Tr^{EAF}_{ADF} is classified as basic under the listed semantics and casting functions.

Explanation. We have limited ourselves to bounded hierarchical or (strongly) consistent EAFs without symmetric attacks and therefore our translation is source–subclass. Not every ADF can be produced – for example, a framework consisting of a single falsum argument is out of the question (see Theorem 8.28). Therefore, the approach is also target–subclass. Let us now analyze whether our translation is injective. We will now assume it is not. The set of arguments in the source and target framework is the same. Consequently, it can only be the case that two (strongly) consistent EAFs $EF_1 = (A, R_1, D_1)$

and $EF_2 = (A, R_2, D_2)$ differ by an attack or a defense attack, but the constructed acceptance conditions are the same. If the conditions constructed for $a \in A$ in both frameworks is the same, then so are the sets of parents, and thus the sets of arguments connected via direct or defense attack to a in EF_1 and EF_2. We can observe that due to the fact that EF_1 and EF_2 are consistent, no argument can defense attack a direct attack it carries out. Therefore, from the translation it follows that $(b, a) \in R_1$ iff $C_a(\{b\} \cap par(a)) = out$, and $(b, a) \in R_2$ iff $C_a(\{b\} \cap par(a)) = out$. This means that $R_1 = R_2$. Furthermore, since no defense attacker can be a direct attacker, the arguments defense attacking the (b, a) conflict can be found easily by verifying for which two–element subsets of $par(a)$ containing b the condition of a turns to in. In other words, the defense attacks can be retrieved easily as well from the conditions and $D_1 = D_2$. Therefore, EF_1 is in fact not different from EF_2, and thus $con\text{-}Tr^{EAF}_{ADF}$ is injective.

The fact that the argument and semantics domain are preserved can be easily observed from the translation. Due to the amount of handled semantics (in a strong manner), we classify $con\text{-}Tr^{EAF}_{ADF}$ as generic. The fact it is exact can be easily seen from the redefinition of Theorem 8.32. We choose not to categorize our translation as structure preserving due to the fact that previously indirect connections between defense attackers and direct attacked arguments change from indirect to direct, even though no argument is added.

Unfortunately, we have a problem with modularity since the defense attack parts need to be connected differently than the direct attack ones. Consider two frameworks $EF_1 = (\{a, b, c\}, \{(a, c)\}, \{(b, (a, c))\})$ and $EF_2 = (\{a, c, d\}, \{(a, c)\}, \{(d, (a, c))\})$. Their union is $EF_3 = EF_1 \cup EF_2 = (\{a, b, c, d\}, \{(a, c)\}, \{(b, (a, c)), (d, (a, c))\})$. The condition for c in the EFC_1 framework is $\neg a \vee b$, while in EFC_2 it is $\neg a \vee d$. Their conjunction is equivalent to $\neg a \vee (b \wedge d)$, which is clearly different from the EF_3 condition $\neg a \vee b \vee d$. Thus, the translation is not $\otimes$–modular. However, it cannot be $\oplus$–modular either. We can look at two AF–style EAFs – $(\{a, b\}, \{(a, b)\}, \emptyset)$ and $(\{b, c\}, \{(c, b)\}, \emptyset)$. The condition of b is respectively $\neg a$ and $\neg c$ in the translated versions of these frameworks. In the translation of their union we will obtain $\neg a \wedge \neg c$, while following the $\oplus$ approach would give us $\neg a \vee \neg c$. Consequently, no straightforward modularity is available, and we would need an approach that could distinguish between the attacks and defense attack parts of the acceptance condition. ■

In Theorem 8.28 we could have observed that the bounded hierarchical EAFs produce AADF$^+$ frameworks. This means that our semantics classification collapses (see Theorem 2.172) and we do not need to restrict ourselves to ca$_2$–family of semantics. This gives us the following results:

Theorem 8.33. *Let $bh - EF$ be a bounded hierarchical EAF and $D^{EF} = (A, C)$ be its corresponding ADF obtained through Translation 47. A set of arguments $E \subseteq A$ is a conflict–free extensions of $bh - EF$ iff it is (pd–acyclic) conflict–free in D^{EF}. $E \subseteq A$ is a stable extensions of $bh - EF$ iff it is (stable) model of D^{EF}. $E \subseteq A$ is a grounded extensions of $bh - EF$ iff it is (acyclic) grounded in D^{EF}. $E \subseteq A$ is a σ extensions of*

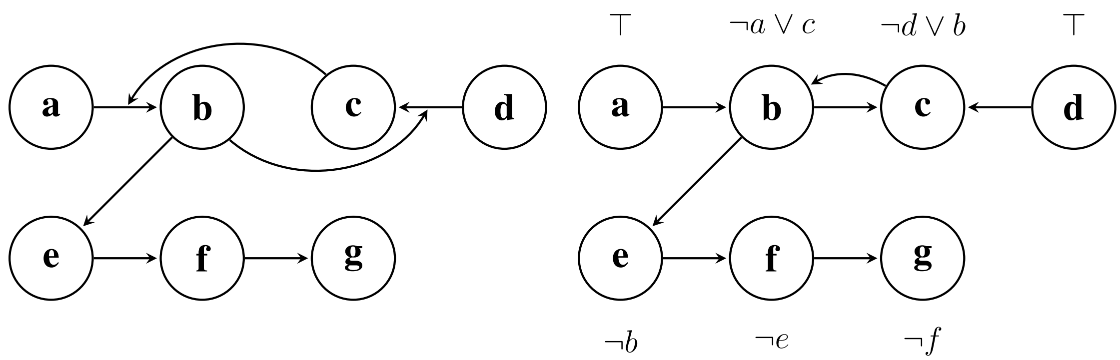
$bh - EF$, where where $\sigma \in \{$admissible, preferred, complete$\}$ iff it is an xy–σ–extension of D^{EF} for $x, y \in \{a, c\}$.

Redefinition of Theorem 8.33: *Let $\sigma^{EAF} \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$ be an EAF semantics and $\sigma^{ADF} \in \{$conflict–free, pd–acyclic conflict–free, xy–admissible, xy–complete, xy–preferred, grounded, acyclic grounded, model, stable$\}$ for $x, y \in \{a, c\}$ be a similar ADF semantics. Let SC_{σ}^{Tr} the identity casting functions for σ. The translation con-Tr_{ADF}^{EAF} is strong and semantics bijective under $(\sigma, SC_{\sigma}^{Tr})$ for EAFs in BH^{EAF}.*

Example 104. [62] Let us consider the EAF $EF = (\{a, b, c, d, e, f, g\}, \{(a, b), (d, c), (b, e), (e, f), (f, g)\}, \{(b, (d, c)), (c, (a, b))\})$ previously described in Example 90 and now depicted in Figure 85a. Its associated ADF, visible in Figure 85b, is $D^{EF} = (\{a, b, c, d, e, f, g\}, \{C_a = \top, C_b = \neg a \vee c, C_c = \neg d \vee b, C_d = \top, C_e = \neg b, C_f = \neg e, C_g = \neg f)$.

The minimal decisively in interpretation v_a and v_d for a and d are simply empty. For b, we can construct the interpretation $v_b^1 = \{a : \mathbf{f}\}$ and $v_b^2 = \{c : \mathbf{t}\}$. Similarly, for c we obtain $v_c^1 = \{d : \mathbf{f}\}$ and $v_c^2 = \{b : \mathbf{t}\}$. For the remaining arguments e, f and g, the interpretation are $v_e = \{b : \mathbf{f}\}$, $v_f = \{e : \mathbf{f}\}$ and $v_g = \{f : \mathbf{f}\}$. Therefore, the minimal partially acyclic evaluations of D^{EF} are $ev_a = ((a), \emptyset, \emptyset)$, $ev_d = ((d), \emptyset, \emptyset)$, $ev_e = ((e), \emptyset, \{b\})$, $ev_f = ((f), \emptyset, \{e\})$, $ev_g = ((g), \emptyset, \{f\})$ for arguments a, d, e, f, g. For arguments b and c, the situation is somewhat more complicated and we obtain the evaluations $ev_b^1 = ((b), \emptyset, \{a\})$, $ev_b^2 = ((c, b), \emptyset, \{d\})$, $ev_c^1 = ((c), \emptyset, \{d\})$, $ev_c^2 = ((b, c), \emptyset, \{a\})$ and $ev_b^3 = ev_c^3 = (\emptyset, \{b, c\}, \emptyset)$. With the exception of ev_b^3 (ev_c^3), all of the listed evaluations are in fact acyclic.

Our D^{EF} has a number of conflict–free extensions and we will focus only on some of them. In particular, we are concerned with $E_1 = \emptyset$, $E_2 = \{a\}$, $E_3 = \{d\}$, $E_4 = \{a, d\}$, $E_5 = \{b, c\}$, $E_6 = \{a, b, c\}$, $E_7 = \{b, c, d\}$, $E_8 = \{a, d, e\}$, $E_9 = \{b, c, f\}$, $E_{10} = \{a, b, c, f\}$, $E_{11} = \{b, c, d, f\}$, $E_{12} = \{a, d, e, g\}$, $E_{13} = \{a, b, c, d\}$ and $E_{14} = \{a, b, c, d, f\}$. Their partially acyclic discarded sets are $E_1^{p+} = E_2^{p+} = E_3^{p+} = \emptyset$, $E_4^{p+} = \{b, c\}$, $E_5^{p+} = E_6^{p+} = E_7^{p+} = E_{13}^{p+} = \{e\}$, $E_8^{p+} = E_{12}^{p+} = \{b, c, f\}$ and $E_9^{p+} = E_{10}^{p+} = E_{11}^{p+} = E_{14}^{p+} = \{e, g\}$. We can verify that all of the listed extensions are indeed ca$_2$–admissible, i.e. every argument in a given set is decisively in w.r.t. the partially acyclic range of this set. However, not all of these extensions are ca$_2$–complete. Due to the fact that both a and d possess empty decisively in interpretations (i.e. they are "initial" arguments), none of the sets E_1, E_2, E_3, E_5, E_6, E_7, E_9, E_10 and E_11 can be ca$_2$–complete. We can observe that e is decisively in w.r.t. the partially acyclic range of E_4. Thus, this set is not ca$_2$–complete either. However, when we accept e, we can also accept g, and thus E_8 also does not meet our requirements. Finally, also E_{13} cannot be complete due to the fact that f is decisively in w.r.t. its partially acyclic range. This leaves us with two ca$_2$–complete extensions, namely $E_{12} = \{a, d, e, g\}$ and $E_{14} = \{a, b, c, d, f\}$. We can observe they are in fact incomparable and do not follow the typical pattern of complete extensions, as was already noted in Theorem 2.158. Both of these sets are also ca$_2$–preferred and model extensions of D^{EF}. By iterating from the empty set and following Definition 2.142 (i.e. we go through extensions E_1, E_4, E_8 and finally E_{12}), we obtain the acyclic grounded extension of D^{EF} $E_{12} = \{a, d, e, g\}$. We can observe that our answers exactly correspond to the sets produced by EF.



(a) A sample EAF (b) Associated ADF

Figure 85: A sample EAF and its associated ADF

8.6.2.2 General EAF

The way we handle EAFs that do not necessarily have to be consistent is very similar to the bypass translation for the consistency forms of AFNs and EASs (Translations 13 and 14). For those arguments that cause the inconsistency (i.e. attack a given argument and at the same time defense attack a conflict directed at it) we introduce bypass arguments that take over the support relation generated from the defense attack, as proposed in Section 4.4.3. The remaining part of this approach is now very similar to the one employed for consistent EAFs. In order to make our approach somewhat more readable, we will introduce the notions of inconsistency origins and replacement functions, similarly as we did for AFNs and EASs in Section 4.4.

Definition 8.34. Let $EF = (A, R, D)$ be an EAF and $a \in A$ an argument. The **inconsistency origin** of a is defined as $O^a = \{b \in A \mid \exists c \in A \text{ s.t. } (b, (c, a)) \in D \text{ and } (b, a) \in R\}$.

We can observe that if a is strongly consistent, then $O^a = \emptyset$. By the abuse of notation, we will write O^E to denote the collection of all inconsistency origins of the arguments in $E \subseteq A$.

Definition 8.35. Let $EF = (A, R, D)$ be an EAF and $(b, a) \in A$ a conflict. Let $D_{b,a} = \{c \mid (c, (b, a)) \in D\}$ be the collection of all arguments defense attacking (b, a). Given a set of arguments $S \subseteq A$, the replacing arguments $P^b = \{e^b \mid e \in S\}$, the **replacement function** for defense attacks is defined as:

$$rep(S, P^b, D_{b,a}) = \begin{cases} D_{b,a} & \text{if } D_{b,a} \cap S = \emptyset \\ D'_{b,a} & \text{if } D_{b,a} \cap S \neq \emptyset, \text{ where } D'_{b,a} = (D_{b,a} \setminus S) \cup \{e^b \mid e \in D_{b,a} \cap S\} \end{cases}$$

Let us now introduce a more general version of Translation 47 which can be applied to frameworks that are not necessarily strongly consistent:

Translation 48. *Let $EF = (A, R, D)$ be a bounded hierarchical EAF or an EAF without symmetric attacks. Let $E \subseteq A$ be the set of arguments in EAF that are not strongly consistent and $A^b = \{a^b \mid a \in O^E\}$ the set of bypass arguments for the elements causing the inconsistencies. For an attack $(b, a) \in R$, let $D_{b,a} = \{c \mid (c, (b, a)) \in D\}$ be the collection of all arguments defense attacking (b, a) and $D'_{b,a} = rep(O^a, A^b, D_{b,a})$ the modification of $D_{b,a}$ replacing occurrences of inconsistency origins by their bypasses.*

The ADF corresponding to EF is defined as $D^{EF} = (A', L, C)$, where $A' = A \cup A^b$, $L = \{(a, b) \mid (a, b) \in R$, or $\exists (c, b) \in R, a \in D'_{c,b}$, or $b = a^b\}$ and $C = \{C_a \mid a \in A'\}$ is the set of acceptance conditions, where every condition C_a is created in the following way:

- *gunctional form:*
  - *if $a \in A$, then for every subset of parents $F \subseteq par(a)$, if $x \in F$ s.t. $(x, a) \in R$ and $\nexists f \subseteq F$ s.t. $f \in D'_{x,a}$, then $C_a(F) = out$; otherwise, $C_a(F) = in$*
  - *if $a^b \in A^b$, then $C_{a^b}(\emptyset) = out$ and $C_{a^b}(\{a\}) = in$ – argument a^b has only one parent, which is its original argument, and its presence is required for the acceptance of a^b.*
- *propositional form:*
  - *if $a \in A$:*
    - *let $b \in A$ be an argument s.t. $(b, a) \in R$ and $D'_{b,a}$ the modified defense attacker collection for this attack. The attack formula corresponding to b is $att^b_a = \neg b \vee \bigvee D'_{b,a}$. If $D'_{b,a}$ is empty, then it is simply $\neg b$.*
    - *the acceptance condition is the conjunction of all such att^b_a parts: $C_a = \bigwedge_{b \in A, (b,a) \in R} att^b_a$. In case a is not attacked at all, it is simply $\top$.*
  - *if $a^b \in A^b$, then $C_{a^b} = a$, i.e. it contains only a positive occurrence of the original argument represented by a^b.*

We can observe that despite the additional arguments, the produced ADFs exhibit all the normal forms they did for consistent EAFs:

Theorem 8.36. *Let $EF = (A, R, D)$ be a bounded hierarchical EAF or an EAF without symmetric attacks and $D^{EF} = (A', L, C)$ its corresponding ADF obtained through Translation 48. D^{EF} is a BADF. It is also in redundancy–free, cleansed, and weakly valid* *form. However, it is not necessarily an AADF$^+$ and does not have to be in relation or strongly valid form. If EF is a bounded hierarchical EAF, then D^{EF} is an AADF$^+$ and is in relation and strongly valid forms.*

Although now we need to account for the auxiliary arguments in our extensions, the proof is only a minor adaptation of the one for consistent frameworks:

Theorem 8.37. *Let $EF = (A, R, D)$ be a bounded hierarchical EAF or an EAF without symmetric attacks and $D^{EF} = (A', L, C)$ its corresponding ADF obtained through Translation 48. Let E^b denote the (possibly empty) set of bypass arguments generated by E in A'.*

If a set of arguments $E \subseteq A$ is a conflict–free extension of EF then $E' = E \cup E^b$ is conflict–free in D^{EF}. If $E' \subseteq A'$ is conflict–free in D^{EF}, then $E = E' \cap A$ is conflict–free in EF.

If a set of arguments $E \subseteq A$ is a stable extension of EF then $E' = E \cup E^b$ is a model of D^{EF}. If $E' \subseteq A'$ is a model of D^{EF}, then $E = E' \cap A$ is stable in EF.

If a set of arguments $E \subseteq A$ is the grounded extension of EF then $E' = E \cup E^b$ is the acyclic grounded extension of D^{EF}. If $E' \subseteq A'$ is the acyclic grounded extension of D^{EF}, then $E = E' \cap A$ is grounded in EF.

If $E \subseteq A$ is a σ–extension of EF, where $\sigma \in \{$admissible, complete, preferred$\}$, then $E' = E \cup E^b$ is a ca_2–σ–extension of D^{EF}. If $E' \subseteq A'$ ca_2–σ–extension of D^{EF}, then $E = E' \cap A$ is a σ–extension of EF.

We can now put the translation into our system and analyze it. Please note we will reuse the notation introduced in the original version. Furthermore, as already noted before, even though EAF^{ADF} is not the most accurate description of EAF–produced ADFs, we will still use it as our target domain.

Redefinition of Translation 48: *Let $BH^{EAF} \cup NSym^{EAF}$ be the collection of all EAFs that are bounded hierarchical or without symmetric attacks on the domain $\mathcal{U}$ and EAF^{ADF} the collection of all EAF–style ADFs on domain $\mathcal{U} \cup \mathcal{U}^b$. The translation $Tr^{EAF}_{ADF} : (BH^{EAF} \cup NSym^{EAF}) \rightarrow EAF^{ADF}$ is defined as $Tr^{EAF}_{ADF}((A, R, D)) = (A', L, C)$ for a framework $(A, R, D) \in (BH^{EAF} \cup NSym^{EAF})$, where $A' = A \cup A^b$ for $A^b = \{a^b \mid a \in O^E\}$ and E being the set of arguments in EAF that are not strongly consistent, $L = \{(a, b) \mid (a, b) \in R$, or $\exists (c, b) \in R, a \in D'_{c,b}$, or $b = a^b\}$, and $C = \{C_a \mid a \in A'\}$ is the set of acceptance conditions, where every condition C_a is created in the following way:*

- *functional form:*
  - *if $a \in A$, then for every subset of parents $F \subseteq par(a)$, if $x \in F$ s.t. $(x, a) \in R$ and $\nexists f \subseteq F$ s.t. $f \in D'_{x,a}$, then $C_a(F) = out$; otherwise, $C_a(F) = in$*
  - *if $a^b \in A^b$, then $C_{a^b}(\emptyset) = out$ and $C_{a^b}(\{a\}) = in$*
- *propositional form:*

– *if $a \in A$, then $C_a = \top$ if there is no $b \in A$ s.t. $(b, a) \in R$; otherwise, $C_a = \bigwedge_{b \in A, (b,a) \in R} att_a^b$, where $att_a^b = \neg b \vee \bigvee D'_{b,a}$ if $D'_{b,a} \neq \emptyset$ and $att_a^b = \neg b$ if $D'_{b,a} = \emptyset$.*

– *if $a^b \in A^b$, then $C_{a^b} = a$.*

Redefinition of Theorem 8.37: *Let $\sigma^{EAF} \in$ {conflict–free, admissible, preferred, complete, grounded, stable} be an EAF semantics and $\sigma^{ADF} \in$ {conflict–free, ca$_2$–admissible, ca$_2$–preferred, c$_2$–complete, acyclic grounded, model} be a similar ADF semantics. Let SC_σ^{Tr} be the removal casting functions for σ defined as $SC_\sigma^X(E) = E \cap A$, where $X = (A, R, D) \in Fr^{EAF}$ is an EAF and $E \in \sigma(Tr_{ADF}^{EAF}(X))$. The translation Tr_{ADF}^{EAF} is strong under (σ, SC_σ^{Tr}). It is semantics bijective under the complete, preferred, grounded and stable semantics and the removal casting functions.*

Analysis of Translation 48: Under the conflict–free, (ca$_2$–) admissible, (ca$_2$–) preferred, (ca$_2$–) complete, (acyclic) grounded and (model) stable semantics and removal casting functions, the translation Tr_{ADF}^{EAF} is:

- source–subclass, target–subclass and injective
- weakly argument domain altering, argument and relation introducing
- generic and weakly semantics domain altering
- semi–structural

The translation is neither $\oplus$ nor $\otimes$–modular. Under the complete, preferred, grounded and stable semantics and removal casting functions, translation Tr_{ADF}^{EAF} is faithful. Translation Tr_{ADF}^{EAF} is classified as basic under the listed semantics and casting functions.

Explanation. Although we can operate on more types of EAFs, the translation is still source and target–subclass. It is also injective; similar explanations as in the analysis of Translation 47 hold. The only difference is that "searching" for defense attackers can produce bypass arguments, but they are assigned uniquely to their original arguments and thus the structure of the source EAF can still be retrieved.

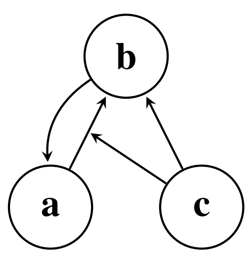
We can observe that in our translation we change both the argument and semantics domain. We also introduce new arguments and the relations from/to them. Furthermore, while defense attackers and the direct attacked arguments were not directly related in EAFs, they can be in the associated ADFs. Faithfulness of our translation under the listed semantics is a result of Theorem 8.37. We choose to classify our approach as semi–structural, as we had exploited the way inconsistencies in the source framework affect the target framework in our translation. The rest of the explanations follows similarly as in the Translation 47. ■

Example 105. Let $EF = (\{a, b, c\}, \{(a, b), (b, a), (c, b)\}, \{(c, (a, b))\})$ be the EAF depicted in Figure 86a. The admissible extensions of this framework are $\emptyset$, $\{c\}$ and $\{a, c\}$.

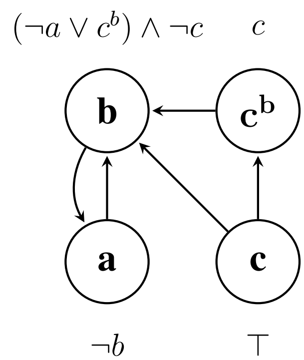
Due to the fact that b is attacked by the initial argument c, it cannot appear in any extension. There is also no reinstatement set for the (a, b) defeat w.r.t. any set of arguments, which means that $\{a\}$ is not an admissible extension either. We only have one complete extension $\{a, c\}$, which is at the same time preferred, stable, and grounded.

We can observe that EF is not consistent framework, i.e. c both attacks b and defense attacks a conflict directed at b. We therefore introduce a bypass argument for c that will take over the defense attack. The ADF associated with our EAF is thus $D^{EF} = (\{a, b, c, c^b\}, \{C_a = \neg b, C_b = (\neg a \vee c^b) \wedge \neg c, C_c = \top, C_{c^b} = c\})$. The minimal decisively in interpretations for our arguments are $v_a = \{b : \mathbf{f}\}$, $v_b^1 = \{a : \mathbf{f}, c : \mathbf{f}\}$, $v_b^2 = \{c^b : \mathbf{t}, c : \mathbf{f}\}$, $v_c = \emptyset$ and $v_{c^b} = \{c : \mathbf{t}\}$. Therefore, the minimal partially acyclic evaluations associated with our arguments are $ev_a = ((a), \emptyset, \{b\})$, $ev_b^1 = ((b), \emptyset, \{a, c\})$, $ev_b^2 = ((c, c^b, b), \emptyset, \{c\})$, $ev_c = ((c), \emptyset, \emptyset)$ and $ev_{c^b} = ((c, c^b), \emptyset, \emptyset)$. Hence, the ca$_2$–admissible extensions of D^{EF} are $\emptyset$, $\{c\}$, $\{c, c^b\}$, $\{a, c\}$ and $\{a, c, c^b\}$. We can observe that if we remove c^b, we retrieve all and only extensions of EF. Please note that a single set in EF can be obtained from more than one set in D^{EF}. The only ca$_2$–complete extension of D^{EF} is $\{a, c, c^b\}$. It is also the single ca$_2$–preferred, acyclic grounded and model extension, which is the desired result. Additionally, we can see that there is a one–to–one correspondence between the answers of D^{EF} and EF.

Let us for a moment assume that we did not detect the inconsistency of EF. The produced ADF would have been $(\{a, b, c\}, \{C_a = \neg b, C_b = (\neg a \vee c) \wedge \neg c, C_c = \top\})$. The condition for C_b would have been equivalent to simply $\neg a \wedge \neg c$. We can observe that in this case, b would be contained in any type of a discarded set of $\{a\}$, thus making the set an admissible extension of any type of our ADF. Since $\{a\}$ is not admissible in EF, such an answer would be quite undesirable.



(a) Sample EAF



(b) Corresponding ADF

Figure 86: Sample not consistent EAF and corresponding ADF

8.6.3 EAFC as ADF

Due to the fact that EAFC semantics are now defined based purely on defeats and do not treat symmetric attacks as a special case, we do not need to exclude frameworks including such conflicts from the translation. Unfortunately, we still need to make a distinction between (strongly) consistent and not consistent EAFCs.

8.6.3.1 Consistent EAFC

The translation from EAFCs to ADFs is very similar to the EAF–ADF one. In order to account for the group defense attack, a single positive literal representing the source of the attack is now a conjunction of positive literals:

Translation 49. *Let $EFC = (A, R, D)$ be a strongly consistent EAFC. For a conflict (b, a), let $D_{b,a} = \{E \mid (E, (b, a)) \in D\}$ denote the sets of arguments defense attacking (b, a). The ADF corresponding to EFC is $D^{EFC} = (A, L, C)$, where $L = \{(a, b) \mid (a, b) \in R$ or $\exists c \in A, E \subseteq A$ s.t. $a \in E$ and $(E, (c, b)) \in D\}$ and $C = \{C_a \mid a \in A\}$ is the set of acceptance conditions, where every condition C_a is created in the following way:*

- *functional form:*
  - *for a subset of parents $B \subseteq par(a)$, if there exists $x \in B$ s.t. $(x, a) \in R$ and $\nexists B' \subseteq B$ s.t. $(B', (x, a)) \in D$, then $C_a(B) = out$; otherwise, $C_a(B) = in$*
- *propositional form:*
  - *let $b \in A$ be an argument s.t. $(b, a) \in R$ and $D_{b,a} = \{B_1, ..., B_m\}$ the collection of all sets defense attacking (b, a). The attack formula corresponding to b is $att_a^b = \neg b \vee (\bigwedge B_1 \vee ... \bigwedge B_m)$. If $D_{b,a}$ is empty, then it is simply $\neg b$.*
  - *the acceptance condition is the conjunction of all such att_a^b parts: $C_a = \bigwedge_{b \in A, (b,a) \in R} att_a^b$. In case a is not attacked at all, it is simply $\top$.*

In the previous sections we have discussed the EAF–style conditions, that were either $\top$ or conjunctions of clauses s.t. there is at most one negated argument in the clause. Moreover, each negation was unique among the clauses. EAFC conditions are similar, with the exception on the uniqueness. Bringing e.g. an attack formula $\neg a \vee (b \wedge c) \vee (d \wedge e)$ to CNF will give us $(\neg a \vee b \vee d) \wedge (\neg a \vee b \vee e) \wedge (\neg a \vee c \vee d) \wedge (\neg a \vee c \vee e)$. We will refer to such conditions as EAFC–style. Please note that not all EAFC–style ADFs can be EAFC–produced. Although we do not have restrictions on e.g. defense attacks the way we did in EAFs, the fact that we either assume consistent frameworks or introduce bypass arguments (more on this in Section 8.6.3.2) means that again, no argument appearing as a negative literal in one clause can appear as a positive one in another.

Let us now move on to the main subclasses and normal forms of the EAFC–produced ADFs.

Theorem 8.38. *Let $EFC = (A, R, D)$ be a strongly consistent EAFC and $D^{EFC} = (A, C)$ its corresponding ADF obtained through Translation 49. D^{EF} is a BADF. It is also in cleansed and weakly valid form. It is not necessarily an AADF$^+$ and does not have to be in redundancy–free, relation or strongly valid form. if EFC is minimal, then D^{EF} is redundancy–free. If EFC is bounded hierarchical, then D^{EFC} is an AADF$^+$, and if it is additionally minimal, then D^{EFC} is in relation and strongly valid forms.*

The results we have presented for the consistent EAFs that met our constraints concerning attack symmetry hold also for EAFCs. This means we can draw connections between conflict–freeness, discarded sets and defense/decisiveness in EAFCs and ADFs. Fortunately, in this case we do not need to limit ourselves to frameworks that are bounded hierarchical or do not have symmetric attacks; the design of conflict–free semantics for EAFCs is much closer to ADFs than it was in the case of EAFs.

Theorem 8.39. *Let EFC be a strongly consistent EAFC and $D^{EFC} = (A, L, C)$ its corresponding ADF obtained through Translation 49. A set of arguments $E \subseteq A$ is a conflict–free extension of EFC iff it is a conflict–free extension of D^{EFC}.*

Lemma 8.40. *Let EFC be a strongly consistent EAFC and $D^{EFC} = (A, L, C)$ its corresponding ADF obtained through Translation 49. Let $E \subseteq A$ be a conflict–free extension of EFC (and thus of D^{EFC}). The discarded set of E in EFC coincides with the partially acyclic discarded set of E in D^{EFC}.*

Lemma 8.41. *Let EFC be a strongly consistent EAFC and $D^{EFC} = (A, L, C)$ its corresponding ADF obtained through Translation 49. A conflict–free set of arguments $E \subseteq A$ defends an argument $a \in A$ in EFC iff a is decisively in w.r.t. the partially acyclic range v_E^p of E in D^{EFC}.*

Again, using these partial results, we can show that the admissible, complete and preferred extensions of EAFCs coincide with their ca$_2$–counterparts in ADFs. The same holds for stable and model extensions due to Lemma 2.159 and the iterative method of building grounded and acyclic grounded extensions from Definition 2.142 and [64]. This brings us to the following result:

Theorem 8.42. *Let EFC be a strongly consistent EAFC and $D^{EFC} = (A, L, C)$ its corresponding ADF obtained through Translation 49. A set of arguments $E \subseteq A$ is a conflict–free extension of EFC iff it is conflict–free in D^{EFC}. E is a stable extension of EFC iff it is a model of D^{EFC}. E is a grounded extension of EFC iff it is the acyclic grounded extension of D^{EFC}. Finally, E is a σ–extension of EFC, where $\sigma \in \{$admissible, complete, preferred$\}$, iff it is a ca$_2$–σ–extension of D^{EFC}.*

We can now put the obtain translation and theorem into our system and analyze them. Due to the fact that we are operating on consistent EAFCs, we will prefix the translation with $con-$.

Redefinition of Translation 49: *Let $SCons^{EAFC}$ be the collection of all strongly consistent EAFCs and $EAFC^{ADF}$ the collection of all EAFC–style ADFs, both on domain $\mathcal{U}$. The translation con-Tr^{EAFC}_{ADF} : $SCons^{EAFC} \rightarrow Fr^{ADF}$ is defined as con-$Tr^{EAFC}_{ADF}((A, R, D)) = (A, L, C)$ for a framework $(A, R, D) \in SCons^{EAFC}$, where $L = \{(a, b) \mid (a, b) \in R$ or $\exists c \in A, E \subseteq A$ s.t. $a \in E$ and $(E, (c, b)) \in D\}$ and $C = \{C_a \mid a \in A\}$ is the set of acceptance conditions, where every condition C_a is created in the following way:*

- *functional form: for every subset of parents $B \subseteq par(a)$, if there exists $x \in B$ s.t. $(x, a) \in R$ and $\nexists B' \subseteq B$ s.t. $(B, (x, a)) \in D$, then $C_a(B) = out$; otherwise, $C_a(B) = in$*
- *propositional form: if a is not attacked, then $C_a = \top$; otherwise, $C_a = \bigwedge_{b \in A, (b,a) \in R} att^b_a$, where, given that $D_{b,a} = \{B_1, ..., B_m\}$ is the collection of all sets defense attacking the conflict (b, a), $att^b_a = \neg b \vee (\bigwedge B_1 \vee ... \bigwedge B_m)$ when $D_{b,a} \neq \emptyset$ and $att^b_a = \neg b$ if $D_{b,a} = \emptyset$.*

Redefinition of Theorem 8.42: *Let $\sigma^{EAFC} \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$ be an EAFC semantics and $\sigma^{ADF} \in \{$conflict–free, ca$_2$–admissible, ca$_2$–preferred, c$_2$–complete, acyclic grounded, model$\}$ be a similar ADF semantics. Let SC^{Tr}_{σ} the identity casting functions for σ. The translation con-Tr^{EAFC}_{ADF} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*

Analysis of Translation 49: Under the conflict–free, (ca$_2$–) admissible, (ca$_2$–) preferred, (ca$_2$–) complete, (acyclic) grounded and (model) stable semantics and identity casting functions, the translation *con-*Tr^{EAFC}_{ADF} is:

- source–subclass, target–subclass and overlapping
- argument domain preserving, argument preserving and relation introducing
- generic, semantics domain preserving and exact
- structural

The translation is neither $\oplus$ nor $\otimes$–modular. Translation *con-*Tr^{EAFC}_{ADF} is classified as basic under the listed semantics and casting functions.

Explanation. Most of our explanations are very similar as in the EAF–ADF consistent translation (Translation 47). The only difference is that now we lose the injective property. Unfortunately, similarly as in the SETAF–ADF approach (Translation 31), the EAFC–ADF translation is overlapping, even though assuming minimality on the source framework can address this behavior. We can consider two EAFCs with an attack from a to b that is defense attacked by sets $\{c, d\}$, $\{d, e\}$ in the first case and $\{c, d\}$, $\{d, e\}$, and $\{c, d, e\}$ in the other. The functional conditions created for b will be the same in both cases. Even though the (propositional) acceptance conditions can be syntactically different, they will be equivalent as well. ■

In Theorem 8.38 we could have observed that the bounded hierarchical EAFs produce AADF$^+$ frameworks. This means that our semantics classification collapses (see Theorem 2.172) and we can shift the Theorem 8.33 to the EAFC setting, which permits us to use any subfamily of ADF semantics:

Theorem 8.43. *Let $bh-EFC$ be a bounded hierarchical EAFC and $D^{EFC} = (A, L, C)$ be its corresponding ADF obtained through Translation 49. A set of arguments $E \subseteq A$ is a conflict–free extensions of $bh-EFC$ iff it is (pd–acyclic) conflict–free in D^{EFC}. $E \subseteq A$ is a stable extensions of $bh-EFC$ iff it is (stable) model of D^{EFC}. $E \subseteq A$ is a grounded extensions of $bh-EFC$ iff it is (acyclic) grounded in D^{EFC}. $E \subseteq A$ is a σ extensions of $bh-EFC$, where where $\sigma \in \{$admissible, preferred, complete$\}$ iff it is an xy–σ–extension of D^{EFC} for $x, y \in \{a, c\}$.*

Redefinition of Theorem 8.43: *Let $\sigma^{EAFC} \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$ be an EAFC semantics and $\sigma^{ADF} \in \{$conflict–free, pd–acyclic conflict–free, xy–admissible, xy–complete, xy–preferred, grounded, acyclic grounded, model, stable$\}$ for $x, y \in \{a, c\}$ be a similar ADF semantics. Let SC^{Tr}_{σ} the identity casting functions for σ. The translation con-Tr^{EAFC}_{ADF} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$ for EAFCs in BH^{EAFC}.*

Example 106. Let us consider the EAFC $EFC = (\{a, b, c, d, e, f\}, \{(c, f), (d, e), (f, a)\}, \{(\{f\}, (d, e)), (\{a, b, c\}, (d, e)), (\{e\}, (c, f))\})$ from Example 11 and depicted in Figure 87a. Its associated ADF is $D^{EFC} = (\{a, b, c, d, e, f\}, \{C_a = \neg f, C_b = \top, C_c = \top, C_d = \top, C_e = \neg d \vee (a \wedge b \wedge c) \vee f, C_f = \neg c \vee e\}$, visible in Figure 87b.

The minimal decisively in interpretations for our arguments are $v_a = \{f : \mathbf{f}\}$, $v_b = \emptyset$, $v_c = \emptyset$, $v_d = \emptyset$, $v_e^1 = \{d : \mathbf{f}\}$, $v_e^2 = \{a : \mathbf{t}, b : \mathbf{t}, c : \mathbf{t}\}$, $v_e^3 = \{f : \mathbf{t}\}$, $v_f^1 = \{c : \mathbf{f}\}$ and $v_f^2 = \{e : \mathbf{t}\}$. From them we can produce a number of (minimal) partially acyclic evaluations. The single minimal evaluation for a is $((a), \emptyset, \{f\})$. For b, c and d we simply create $((b), \emptyset, \emptyset)$, $((c), \emptyset, \emptyset)$ and $((d), \emptyset, \emptyset)$. For e, we have in total four meaningful evaluations; $((e), \emptyset, \{d\})$, $((a, b, c, e), \emptyset, \{f\})$, $((f, e), \emptyset, \{c\})$ and $(\emptyset, \{e, f\}, \emptyset)$. Finally, for f we obtain $((f), \emptyset, \{c\})$, $((e, f), \emptyset, \{d\})$, $((a, b, c, e, f), \emptyset, \{f\})$ and again, $(\emptyset, \{e, f\}, \emptyset)$.

The framework D^{EFC} has a number of ca$_2$–admissible extensions. We can observe that $\emptyset$ and any combination of arguments b, c and d are admissible in D^{EFC}. Argument a will never appear in an admissible extension due to the fact that there is no conflict–free set of D^{EFC} that would have f in its partially acyclic discarded set. The evaluation for f responsible for this is $((a, b, c, e, f), \emptyset, \{f\})$. Neither $\{e\}$ nor $\{f\}$ are themselves ca$_2$–admissible. Their partially acyclic discarded sets do not include d and c respectively. However, the set $\{e, f\}$ already becomes ca$_2$ admissible thanks to the evaluation $(\emptyset, \{e, f\}, \emptyset)$. This also means we can extended this set with any combination of arguments b, c and d and still obtain a ca$_2$–admissible extension. We therefore retrieve all and only sets produced by EFC (for details on the extensions of EF please consult Example 11). The ca$_2$–complete extensions of D^{EFC} are $\{b, c, d\}$ and $\{b, c, d, e, f\}$. It is easy to verify that the first set is acyclic grounded and the second is ca$_2$–preferred. The set $\{b, c, d, e, f\}$ is also the only model of

D^{EFC}, as w.r.t. $\{b, c, d\}$, the condition of a is satisfied. These answers correspond exactly to the extensions of EFC.

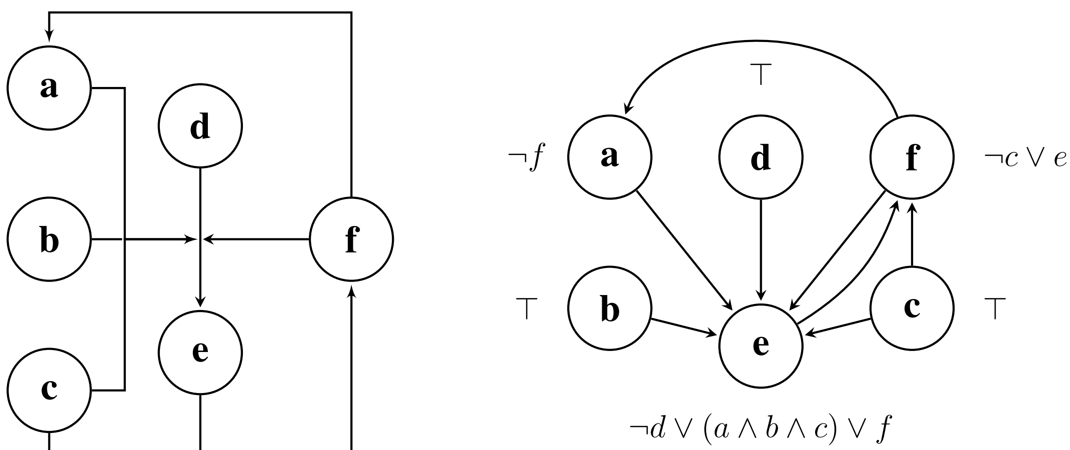


Figure 87: Sample EAFC and its associated ADF

8.6.3.2 General EAFC

The way we handle the inconsistencies in the source EAFC is similar to the EAF case. Following the approach used e.g. in Translation 13 and sketched for EAFs in Section 4.4.3, we introduce bypass arguments for the arguments appearing in defense attacks that cause the inconsistencies. The bypasses then replace the occurrences of the original arguments and are tied to them by support. Just like in the previous cases, we introduce some auxiliary notions to improve the readability of the translation:

Definition 8.44. Let $EFC = (A, R, D)$ be an EACF and $a \in A$ an argument. The **inconsistency origin** of a is defined as $O^a = \{b \in A \mid \exists c \in A, B \subseteq A$ s.t. $b \in B, (B, (c, a)) \in D$ and $(b, a) \in R\}$.

If a is strongly consistent, then $O^a = \emptyset$. By the abuse of notation, we will write O^E to denote the collection of all inconsistency origins of the arguments in $E \subseteq A$.

Definition 8.45. Let $EFC = (A, R, D)$ be an EAFC and $(b, a) \in A$ a conflict. Let $D_{b,a} = \{E \mid (E, (b, a)) \in D\}$ be the collection of all sets of arguments defense attacking (b, a). Given a set of arguments $S \subseteq A$, the replacing arguments $P^b = \{e^b \mid e \in S\}$, the **replacement function** for defense attacks is defined as:

$$rep(S, P^b, D_{b,a}) = \begin{cases} D_{b,a} & \text{if } \forall B \in D_{b,a}, B \cap S = \emptyset \\ D'_{b,a} & \text{if } \exists B \in D_{b,a} \text{ s.t. } \cap S \neq \emptyset, \text{ where } D'_{b,a} = \{(B \setminus S) \cup \{e^b \mid \\ & e \in B \cap S\} \mid B \in D_{b,a}\} \end{cases}$$

Translation 50. *Let $EFC = (A, R, D)$ be an EAFC, $E \subseteq A$ the set of arguments in EAFC that are not strongly consistent and $A^b = \{a^b \mid a \in O^E\}$ the set of bypass arguments for the elements causing the inconsistencies. For an attack $(b, a) \in R$, let $D_{b,a} = \{B \mid (B, (b, a)) \in D\}$ be the collection of all sets of arguments defense attacking (b, a) and $D'_{b,a} = rep(O^a, A^b, D_{b,a})$ the modification of $D_{b,a}$ replacing occurrences of inconsistency origins by their bypasses.*

The ADF corresponding to EFC is defined as $D^{EFC} = (A', L, C)$, where $A' = A \cup A^b$, $L = \{(a, b) \mid (a, b) \in R$, or $\exists (c, b) \in R, G \in D'_{c,b}$ s.t. $a \in G$, or $b = a^b\}$ and $C = \{C_a \mid a \in A'\}$ is the set of acceptance conditions, where every condition C_a is created in the following way:

- *functional form:*
  - *if $a \in A$, then for every subset of parents $F \subseteq par(a)$, if there exists $x \in F$ s.t. $(x, a) \in R$ and $\nexists F' \subseteq F$ s.t. $F' \in D'_{x,a}$, then $C_a(F) = out$; otherwise, $C_a(F) = in$*
  - *if $a^b \in A^b$, then $C_{a^b}(\emptyset) = out$ and $C_{a^b}(\{a\}) = in$ – argument a^b has only one parent, which is its original argument, and its presence is required for the acceptance of a^b.*
- *propositional form:*
  - *if $a\ inA$:*
    - *let $b \in A$ be an argument s.t. $(b, a) \in R$ and $D'_{b,a} = \{B'_1, ..., B'_m\}$ the modified defense attacker collection for this attack. The attack formula corresponding to b is $att^b_a = \neg b \vee (\bigwedge B'_1 \vee ... \bigwedge B'_m)$. If $D'_{b,a}$ is empty, then it is simply $\neg b$.*
    - *the acceptance condition is the conjunction of all such att^b_a parts: $C_a = \bigwedge_{b \in A, (b,a) \in R} att^b_a$. In case a is not attacked at all, it is simply $\top$.*
  - *if $a^b \in A^b$, then $C_{a^b} = a$, i.e. it contains only a positive occurrence of the original argument represented by a^b.*

We can observe that the EAFCs that are not necessarily consistent will produce ADFs belonging to the same classes and norms as the consistent ones due to the use of bypass arguments. The proof is easily adapted from Theorems 8.36 and 8.38 and will thus be omitted here.

Theorem 8.46. *Let $EFC = (A, R, D)$ be an EAFC and $D^{EFC} = (A', L, C)$ its corresponding ADF obtained through Translation 50. D^{EF} is a BADF. It is also in cleansed and weakly valid form. It is not necessarily an AADF$^+$ and does not have to be in redundancy–free, relation or strongly valid form. if EFC is minimal, then D^{EF} is redundancy–free. If EFC is bounded hierarchical, then D^{EFC} is an AADF$^+$, and if it is additionally minimal, then D^{EFC} is in relation and strongly valid forms.*

Although now we need to account for the auxiliary arguments in our extensions, the proof is only a minor adaptation of the one for consistent frameworks:

Theorem 8.47. *Let $EFC = (A, R, D)$ be an EAFC and $D^{EFC} = (A', L, C)$ its corresponding ADF obtained through Translation 50. Let E^b denote the (possibly empty) set of bypass arguments generated by E in A'.*

If a set of arguments $E \subseteq A$ is a conflict–free extension of EFC then $E' = E \cup E^b$ is conflict–free in D^{EFC}. If $E' \subseteq A'$ is conflict–free in D^{EF}, then $E = E' \cap A$ is conflict–free in EFC.

If a set of arguments $E \subseteq A$ is a stable extension of EFC then $E' = E \cup E^b$ is a model of D^{EFC}. If $E' \subseteq A'$ is a model of D^{EF}, then $E = E' \cap A$ is stable in EFC.

If a set of arguments $E \subseteq A$ is the grounded extension of EFC then $E' = E \cup E^b$ is the acyclic grounded extension of D^{EFC}. If $E' \subseteq A'$ is the acyclic grounded extension of D^{EF}, then $E = E' \cap A$ is grounded in EFC.

If $E \subseteq A$ is a σ–extension of EFC, where $\sigma \in \{$admissible, complete, preferred$\}$, then $E' = E \cup E^b$ is a ca_2–σ–extension of D^{EFC}. If $E' \subseteq A'$ ca_2–σ–extension of D^{EFC}, then $E = E' \cap A$ is a σ–extension of EFC.

We can now put these results into our system and analyze the properties of our translation:

Redefinition of Translation 50: *Let Fr^{EAFC} be the collection of all EAFCs on domain $\mathcal{U}$ and $EAFC^{ADF}$ the collection of all EAFC–style ADFs on domain $\mathcal{U} \cup \mathcal{U}^b$. The translation $Tr^{EAFC}_{ADF} : Fr^{EAFC} \to Fr^{ADF}$ is defined as $Tr^{EAFC}_{ADF}((A, R, D)) = (A', L, C)$ for a framework $(A, R, D) \in Fr^{EAF}$, where $A' = A \cup A^b$ for $A^b = \{a^b \mid a \in O^E\}$ and $E \subseteq A$ being the set of arguments that are not strongly consistent, $L = \{(a, b) \mid (a, b) \in R$, or $\exists (c, b) \in R, G \in D'_{c,b}$ s.t. $a \in G$, or $b = a^b\}$ and $C = \{C_a \mid a \in A'\}$ is the set of acceptance conditions, where every condition C_a is created in the following way:*

- *functional form:*
  - *if $a \in A$, then for every subset of parents $F \subseteq par(a)$, if there exists $x \in F$ s.t. $(x, a) \in R$ and $\nexists F' \subseteq F$ s.t. $F' \in D'_{x,a}$, then $C_a(F) = out$; otherwise, $C_a(F) = in$*
  - *if $a^b \in A^b$, then $C_{a^b}(\emptyset) = out$ and $C_{a^b}(\{a\}) = in$.*
- *propositional form:*
  - *if $a\ inA$, then $C_a = \top$ if there is no $b \in A$ s.t. $(b, a) \in R$; otherwise, $C_a = \bigwedge_{b \in A, (b,a) \in R} att^b_a$, where, given that $D'_{b,a} = \{B'_1, ..., B'_m\} = rep(O^a, A^b, D_{b,a}$ the modified defense attacker collection for the attack (b, a), $att^b_a = \neg b \vee (\bigwedge B'_1 \vee ... \bigwedge B'_m)$ if $D'_{b,a} \neq \emptyset$ and $att^b_a = \neg b$ if $D'_{b,a} = \emptyset$.*
  - *if $a^b \in A^b$, then $C_{a^b} = a$.*

Redefinition of Theorem 8.47: *Let $\sigma^{EAFC} \in$ {conflict–free, admissible, preferred, complete, grounded, stable} be an EAFC semantics and $\sigma^{ADF} \in$ {conflict–free, ca$_2$–admissible, ca$_2$–preferred, c$_2$–complete, acyclic grounded, model} be a similar ADF semantics. Let SC^{Tr}_{σ} be the removal casting functions for σ defined as $SC^{X}_{\sigma}(E) = E \cap A$, where $X = (A, R, D) \in Fr^{EAFC}$ is an EAF and $E \in \sigma(Tr^{EAFC}_{ADF}(X))$. The translation Tr^{EAFC}_{ADF} is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is semantics bijective under the complete, preferred, grounded and stable semantics and the removal casting functions.*

Analysis of Translation 50: Under the conflict–free, (ca$_2$–) admissible, (ca$_2$–) preferred, (ca$_2$–) complete, (acyclic) grounded and (model) stable semantics and removal casting functions, the translation Tr^{EAFC}_{ADF} is:

- full, target–subclass and overlapping
- weakly argument domain altering, argument and relation introducing
- generic and weakly semantics domain altering
- semi–structural

The translation is neither $\oplus$ nor $\otimes$–modular. Under the complete, preferred, grounded and stable semantics and removal casting functions, Tr^{EAFC}_{ADF} is faithful. Translation Tr^{EAFC}_{ADF} is classified as basic under the listed semantics and casting functions.

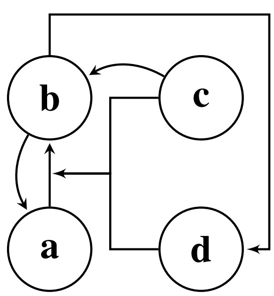
Explanation. We can now operate on all EAFCs and thus our translation is full. It is also easy to observe that we change both the argument and semantics domain. We also introduce new arguments and the relations from/to them. Furthermore, while defense attackers and the direct attacked arguments were not directly related in EAFCs, they can be in the associated ADFs. We choose to classify our approach as semi–structural, as we had exploited the way inconsistencies in the source framework affect the target framework in our translation. The rest of the explanations follows similarly as in the translation for consistent EAFCs. ■

Example 107. Let us consider a modification of the framework from Example 105. The analysis will be similar, but it will depict how bypass argument introduction occurs in group defense attack. Let $EFC = (\{a, b, c, d\}, \{(a, b), (b, a), (c, b), (b, d)\}, \{(\{c, d\}, (a, b))\})$ be the EAFC depicted in Figure 88a. The admissible extensions of this framework are $\emptyset$, $\{c\}$, $\{a, c\}$, $\{c, d\}$ and $\{a, c, d\}$. The set $\{a, c, d\}$ is at the same time the only complete, preferred, stable, and grounded extension.

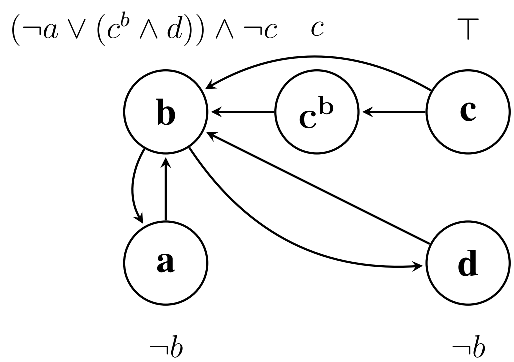
EFC is not a consistent framework, i.e. c both attacks b and participates in a defense attack on a conflict directed at b. The ADF associated with our EAFC is thus $D^{EFC} = (\{a, b, c, d, c^b\}, \{C_a = \neg b, C_b = (\neg a \vee (c^b \wedge d)) \wedge \neg c, C_c = \top, C_{c^b} = c, C_d = \neg b\})$ (see Figure 88b). The minimal decisively in interpretations for our arguments are $v_a = \{b : \mathbf{f}\}$, $v_b^1 = \{a : \mathbf{f}, c : \mathbf{f}\}$, $v_b^2 = \{c^b : \mathbf{t}, d : \mathbf{t}, c : \mathbf{f}\}$, $v_c = \emptyset$, $v_{c^b} = \{c : \mathbf{t}\}$ and $v_d = \{b : \mathbf{f}\}$.

Therefore, the meaningful partially acyclic evaluations associated with our arguments are $ev_a = ((a), \emptyset, \{b\})$, $ev_b^1 = ((b), \emptyset, \{a, c\})$, $ev_b^2 = ((d, c, c^b, b), \emptyset, \{c, b\})$, $ev_c = ((c), \emptyset, \emptyset)$, $ev_{c^b} = ((c, c^b), \emptyset, \emptyset)$ and $ev_d = ((d), \emptyset, \{b\})$. Hence, the ca$_2$–admissible extensions of D^{EF} are $\emptyset$, $\{c\}$, $\{c, c^b\}$, $\{a, c\}$, $\{a, c, c^b\}$, $\{c, d\}$, $\{c, d, c^b\}$, $\{a, c, d\}$ and $\{a, c, d, c^b\}$. We can observe that if we remove c^b, we retrieve all and only extensions of EFC. Please note that a single set in EFC can be obtained from more than one set in D^{EFC}. The only ca$_2$–complete extension of D^{EFC} is $\{a, c, d, c^b\}$. It is also the single ca$_2$–preferred, acyclic grounded and model extension, which is the desired result. Additionally, we can see that there is a one–to–one correspondence between the answers of D^{EFC} and EFC.

Let us for a moment assume that we did not detect the inconsistency of EFC. The produced ADF would have been $(\{a, b, c, d\}, \{C_a = \neg b, C_b = (\neg a \vee (c \wedge d)) \wedge \neg c, C_c = \top, C_d = \neg b\})$. The condition for C_b would have been equivalent to simply $\neg a \wedge \neg c$, similarly as in Example 105. This means that $\{a\}$ would have been a ca$_2$–admissible extension of D^{EFC}, despite the fact that this set is not admissible in EFC.



(a) Sample EAF



(b) Corresponding ADF

Figure 88: Sample not consistent EAFC and corresponding ADF

8.6.4 Improvements

In this section we have considered a number of translations from EAF(C)s to ADFs. We could have observed that they allowed us to go beyond the bounded hierarchical subclass, to which all of the remaining approaches were limited. Although the most general translations we have obtained were faithful, creating exact approaches is indeed possible if we replicate the self–attacker consistency form (see Section 4.4.2) rather than the pure bypass one, which was used so far. The reason why we did not apply it to the EAF–ADF translation lies in the issues mentioned in Section 2.1.4.2; the differing intuitions on the conflict–free semantics would still force us to limit ourselves to a subclass of the extended frameworks on which there is no difference between the EAF and EAFC semantics. However, in the EAFC–ADF case, the translation would be both full and exact. We will therefore include it in our future analysis, but before we do so we would like to address the difficulties this approach has with the stable semantics (see Section 4.4.2).

8.7 EAFs and EAFCs as Other Frameworks

In this section we have neglected only two frameworks; BAFs and EASs. Although BAFs have been a most interesting target for an AFRA translation, an attack in EAFs needs to be accompanied by its source. Consequently, if we were to repeat the AFRA–BAF construction, the d–family of BAF would be somewhat weak for our purposes. Unfortunately, the other families are not fully defined. Moreover, due to the lack of group relations in BAFs, the approach presented in the EAF–AFN Translation 43 is not that easy to adapt to this framework. Therefore, for now we propose to use e.g. the Dung's framework as a bypass for an EAF–BAF translation. Concerning the evidential systems, EASs can handle EAFs in a manner similar to AFNs. The only issue lies in reorganizing the support sets and adding the support from the evidence argument. We refer the reader to Section 10.5 focusing on the AFN–EAS translation, which contains the details on how such modifications can be performed.

8.8 Summary

The results of our translations are summarized in Tables 10 and 11. We can observe that we have failed to find a translation from EAFs to other frameworks that would not be source–subclass, though in the case of ADFs this issue was caused by differing intuition on the conflict–free semantics, not the unusual structure of the complete semantics. Out of the possible approaches, the conversions to EAFCs and ADFs appear to have the widest input range. These are also the only two frameworks for which we have managed to construct an exact approach. Nevertheless, for now we are not able to say whether the exact translations from bounded hierarchical EAFs to structures other than EAFCs and ADFs are possible or not. The situation looks somewhat better for EAFCs; although the only exact translation is a source–subclass one, there exists a full faithful one. Moreover, we believe it can be improved by mimicking self–attacker consistency normal form rather than normal bypass one in the translation. Nevertheless, in both cases, the target structures are the ADFs. The remaining results for EAFCs resemble the ones obtained for EAFs.

Table 10: Translations from EAFs to other frameworks

Properties		AF	EAFC	SETAF	AFRA	AFN		ADF	
Translation		38	36	40	42	43	44	47	48
Strength	cf	$\subseteq$–weak	exact	$\subseteq$–weak	$\subseteq$–weak	$\subseteq$–weak	$\subseteq$–weak	exact	strong
	adm	strong	exact	strong	$\subseteq$–weak	strong	strong	exact	strong
	comp	faithful	exact	faithful	faithful	faithful	faithful	exact	faithful
	pref	faithful	exact	faithful	faithful	faithful	faithful	exact	faithful
	grd	faithful	exact	faithful	faithful	faithful	faithful	exact	faithful
	stb	faithful	exact	faithful	faithful	faithful	faithful	exact	faithful
Functional		source–subclass	source–subclass	source–subclass	source–subclass	source–subclass	source–subclass	source–subclass	source–subclass
		target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass
		injective	injective	injective	injective	injective	injective	injective	injective
Syntactical		argument domain altering	argument domain preserving	weakly argument domain altering	argument domain preserving	weakly argument domain altering	argument domain altering	argument domain preserving	weakly argument domain altering
		argument introducing	structure preserving	argument introducing	structure preserving	argument introducing	argument introducing	relation introducing	argument introducing
		induced attack introducing		induced attack introducing		induced attack introducing	support introducing		relation introducing
						support introducing			
Semantical		generic	generic	generic	generic	generic	generic	generic	generic
		semantics domain altering	semantics domain preserving	weakly semantics domain altering	semantics domain altering	weakly semantics domain altering	semantics domain altering	semantics domain preserving	weakly semantics domain altering
Computational		semi–structural	structural	structural	structural	semi–structural	semi–structural	structural	semi–structural
		modular	modular		modular		modular		

Table 11: Translations from EAFCs to other frameworks

Properties		AF	EAF	SETAF	AFN		ADF	
Translation		39	37	41	45	46	49	50
Strength	cf	$\subseteq$–weak	$\subseteq$–weak	$\subseteq$–weak	$\subseteq$–weak	$\subseteq$–weak	exact	strong
	adm	strong	strong	strong	strong	strong	exact	strong
	comp	faithful	faithful	faithful	faithful	faithful	exact	faithful
	pref	faithful	faithful	faithful	faithful	faithful	exact	faithful
	grd	faithful	faithful	faithful	faithful	faithful	exact	faithful
	stb	faithful	faithful	faithful	faithful	faithful	exact	faithful
Functional		source–subclass	source–subclass	source–subclass	source–subclass	source–subclass	source–subclass	full
		target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass
		injective	injective	overlapping	overlapping	injective	overlapping	overlapping
Syntactical		argument domain altering	argument domain altering	weakly argument domain altering	weakly argument domain altering	argument domain altering	argument domain preserving	weakly argument domain altering
		argument introducing	argument introducing	argument introducing	argument introducing	argument introducing	relation introducing	argument introducing
		induced attack introducing	induced attack introducing	induced attack introducing	induced attack introducing	support introducing		relation introducing
					support introducing			
Semantical		generic	generic	generic	generic	generic	generic	generic
		semantics domain altering	semantics domain altering	weakly semantics domain altering	weakly semantics domain altering	semantics domain altering	semantics domain preserving	weakly semantics domain altering
Computational		semi–structural	semi–structural	structural	semi–structural	semi–structural	structural	semi–structural
		modular	modular			modular		

9 Translating BAFs

In this section we will discuss how to translate BAFs to other frameworks. However, please note that unlike the previous sections, this one will be more focused on discussion than giving translations themselves. Moreover, we will also omit the usual "Improvements" section. The main reason for that is the freedom that BAFs give us in the choice of indirect attacks and their usage in conflict–freeness and defense, which makes their semantics significantly different from the approaches in other structures. Consequently, most of the analysis can be done only for certain BAF subclasses, where a fixed set of indirect attacks is assumed. Moreover, although e.g. necessary support can be modeled within BAFs, the resulting extensions can be different from the ones in AFNs. One of the reasons for this situation is the presence of support cycles, which are ignored in one framework, and treated specially in the other. Thus, BAFs focus on the analysis of a given relation, not in creating a translation that would retrieve the extensions of the framework in which the relation was first proposed.

9.1 BAF as AF

The translations from BAFs to AFs have two purposes. Initially, the authors were interested in obtaining an AF that would retrieve the extensions of the source BAF [29]. Later [30, 31], the translation–based family of semantics was proposed, i.e. the extensions of the target structure would be taken as the desired ones, somewhat independently of their relation to the d–/s–/c–/i– families of semantics. Moreover, all of the proposals were created for a given set of indirect attacks, chosen for modeling the deductive and necessary supports. In what follows we will recall the existing results and propose more general versions when possible.

9.1.1 Attack Propagation Translation

Although the attack propagation translation is only the second approach for BAF–AF conversion, we recall it as first due to the fact that apart from the "standard" semantics explained in Section 2.2.1, BAFs have translation–based semantics as well. In [30] it was proposed that by adding indirect attacks appropriate for a given interpretation of support, we can drop the support relation altogether and focus only on the resulting AF. The extensions of the produced structure were then taken as the extensions of the source one, without further conditions:

Translation 51. *Let $BF = (A, R, S)$ be a BAF specialized for deductive support and $R' = \{R^{sup}, R^{med}_{R^{sup}}\} \subseteq R^{ind}$ the collection of supported and super–mediated attacks in BF. The associated attack propagation AF for deductive support is $apd - F^{BF} = (A, R \cup \bigcup R')$.*

Translation 52. *Let $BF = (A, R, S)$ be a BAF specialized for necessary support and $R' = \{R^{sec}, R^{ext}\} \subseteq R^{ind}$ the collection of secondary and extended attacks in BF [21]. The associated attack propagation AF for necessary support is $apn - F^{BF} = (A, R \cup \bigcup R')$.*

[21] As already mentioned in Section 2.2.1, please observe that what we understand as extended attack here corresponds to a particular case of the original version. This change was motivated by the fact that the other cases are already covered by other existing conflicts.

As already stated in [30], the deductive and necessary supports are tightly related. Not only their meaning is dual, but we can observe that if we invert the support relation, the supported attack becomes extended attack, and secondary turns to mediated. Unfortunately, the super mediated attack does not have its associated dual conflict. For these reasons it was also proposed to model the necessary support through dual, to permit the propagation of extended attacks. Nevertheless, the work in [31] comes back to the secondary–extended approach again, and thus we refer the readers to [30] for further details on the previous method.

We can observe that by adding indirect conflicts to the structure, we not only treat them as attacks that would break conflict–freeness, but also as ones sufficient for defense. In terms of our new BAF semantics classification, this means that we are using the same parametrization for both notions. Consequently, we can present a more general attack propagation BAF–AF translation, and observe that the resulting extensions are the same as in the d–family of semantics.

Translation 53. *Let $BF = (A, R, S)$ be a BAF and $R' \subseteq R^{ind}$ a collection of indirect attacks in BF. The attack propagation AF associated with BF w.r.t. R' is $F^{BF} = (A, R \cup \bigcup R')$.*

Theorem 9.1. *Let $BF = (A, R, S)$ be a BAF, $R' \subseteq R^{ind}$ a collection of indirect attacks in BF and $F^{BF} = (A, R \cup \bigcup R')$ its associated attack propagation AF obtained through Translation 53. $E \subseteq A$ is a +conflict–free extension of BF w.r.t. R' iff it is conflict–free in F^{BF}. E is a d–σ extension of BF w.r.t. (R', R'), where $\sigma \in \{$admissible, complete, preferred$\}$ iff it is a σ–extension of F^{BF}. E is stable in BF w.r.t. R' iff it is stable in F^{BF}. E is d–grounded w.r.t. R' in BF iff it is grounded in F^{BF}.*

This behavior of the semantics can be trivially proved simply by observing that the definition of a given semantics in F^{BF} becomes identical to the ones in BF. However, what is more interesting, is the fact that if we assume certain indirect conflicts, it is not just +conflict–freeness, but also closure that can hold. For example, if our choice was to use the secondary attacks, defending an argument meant that the arguments supporting it would also be defended. Similarly, the use of supported and super mediated attacks leads to the fact that the defense of an argument implied defense of the arguments it supported. This means that the d–complete extensions could also be closed and inverse closed under support and exhibit properties we would expect from the c– and i–complete semantics, if they existed (see [30, 31] for more details). It is also possible that for particular combinations of conflicts, further properties such as safety can be enforced. Nevertheless, for our purposes closure is sufficient:

Theorem 9.2. *Let $BF = (A, R, S)$ be a deductive BAF and $R' = \{R^{sup}, R^{med}_{R^{sup}}\} \subseteq R^{ind}$ the collection of supported and super–mediated attacks in BF. Let $apd - F^{BF} = (A, R \cup \bigcup R')$ be the associated attack propagation AF obtained through Translation 53 and $E \subseteq A$ a complete extension of $apd - F^{BF}$. Then, E is closed under S in BF.*

Theorem 9.3. *Let $BF = (A, R, S)$ be a BAF and $R' = \{R^{sec}\} \subseteq R^{ind}$ the collection of secondary attacks in BF. Let $apd - F^{BF} = (A, R \cup \bigcup R')$ be the associated attack propagation AF obtained through Translation 53 and $E \subseteq A$ a complete extension of $apd - F^{BF}$. Then, E is inverse closed under S in BF.*

Please note we are not entirely sure whether the inverse closure property holds if we assume both secondary and extended attacks. In the proof of Theorem 9.2 we have depended on the fact that if an admissible extension defends a given argument, then it defends the arguments it supports. This is not necessarily true if we take both secondary and extended attacks. Let us look at the BAF presented in Figure 89 and assume that a_1 is contained in some admissible extension. We can observe that the extended attacker a_3 of a_2 is neither an extended nor secondary attacker of a_1. However, the supporter a_4 of a_3 secondary attacks a_1. Let us assume that the admissible extension thus contains a_6 to defend a_1. However, a_6 does not defend a_2 against a_3. We can continue the analysis each time choosing the appropriate extended attacker. Although we are only working with finite frameworks here and are bound to reach the "end" of a framework and resolve the situation in a way that our admissible extension would defend a_2, for the time being we choose to limit ourselves to the provided result.

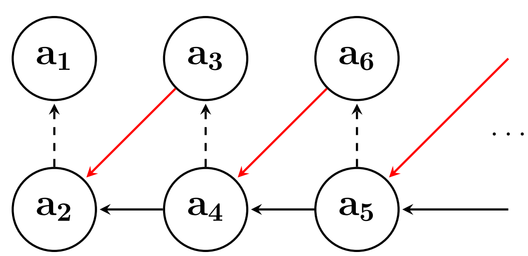


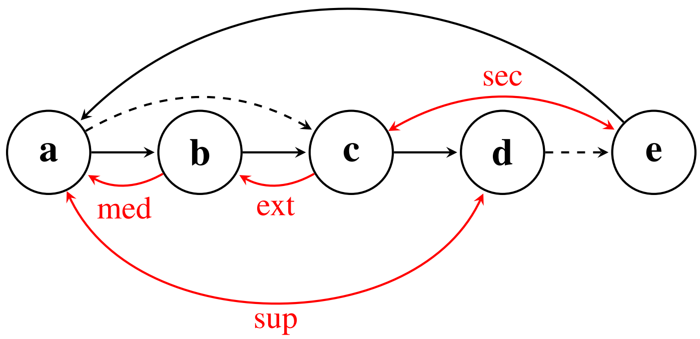
Figure 89: Sample BAF with extended attacks

We can also observe that if our indirect attacks do not lead to desired closure, it can also be enforced by using defense against auxiliary arguments. In this case, we can modify the translation in a manner similar to Translation 58. We turn a supporter of an argument into its sole defender against an auxiliary attacker and thus enforce its presence in an admissible extension. The new attacker is self–conflicting in order to prevent it from coming up in an extension. Thus, we merge the attack propagation and defender approaches. However, please note that as we are using defense, the desired closure is obtained only when we consider extensions that are at least admissible:

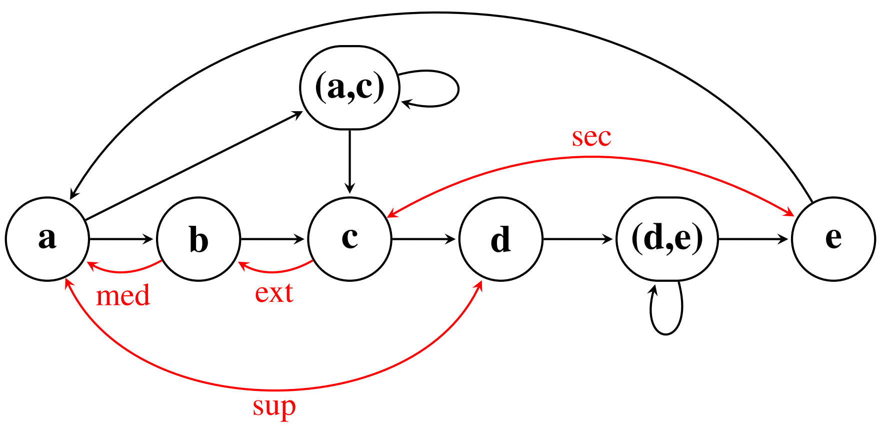
Translation 54. *Let $BF = (A, R, S)$ be a BAF and $R' \subseteq R^{ind}$ a collection of indirect attacks in BF. The associated inverse closure attack propagation–defender AF w.r.t. R' is $iclo - F^{BF} = (A', R'')$, where $A' = A \cup S$ and $R'' = R \cup \bigcup R' \cup \{(x, x) \mid x \in S\} \cup \{(b, (b, a)), ((b, a), a) \mid a, b \in A, (b, a) \in S\}$.*

Theorem 9.4. *Let $BF = (A, R, S)$ be a BAF, $R' \subseteq R^{ind}$ a collection of indirect attacks in BF and $iclo - F^{BF} = (A', R'')$ its associated inverse closure attack propagation–defender*

AF w.r.t. R' *obtained through Translation 54. A set of arguments* $E \subseteq A$ *is i–admissible (i–preferred) in* BF *w.r.t.* (R', R') *iff it is admissible (preferred) in* $iclo - F^{BF}$.



(a) Sample BAF with possible indirect attacks



(b) Possible associated AF

Figure 90: Sample BAF with possible indirect attacks and possible associated AF

Example 108. Let us consider a simple BAF $BF = (\{a, b, c, d, e\}, \{(a, b), (b, c), (c, d), (e, a)\}, \{(a, c), (d, e)\})$. The supported attack, secondary, extended, mediated and super–mediated attacks in this framework are respectively $R^{sup} = \{(a, d), (d, a)\}$, $R^{sec} = \{(e, c), (c, e)\}$, $R^{ext} = \{(c, b)\}$, $R^{med} = \{(b, a)\}$ and $R^{med}_{R^{sup}} = \{(b, a)\}$. The framework, along with its indirect conflicts, is visible in Figure 90a.

The associated AF targeted at inverse closure is $iclo - F^{BF} = (\{a, b, c, d, e, (a, c), (d, e)\}, X \cup \{(a, b), (b, c), (c, d), (e, a), ((a, c), (a, c)), ((d, e), (d, e)), (a, (a, c)), ((a, c), c), (d, (d, e)), ((d, e), e)\})$, where X is a set of indirect attacks that we will be changing now. The framework, along with all possible types of indirect conflicts that we might add, is visible in Figure 90b.

Let us first consider $X = \emptyset$. The +conflict–free sets of BF w.r.t. only direct attacks are $\emptyset$, $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$, $\{a, c\}$, $\{a, d\}$, $\{b, d\}$, $\{b, e\}$, $\{c, e\}$, $\{d, e\}$ and $\{b, d, e\}$. Out of this, the d–admissible extensions are $\emptyset$, $\{e\}$, $\{b, e\}$ and $\{b, d, e\}$. The i–admissible sets are $\emptyset$ and $\{b, d, e\}$; the latter is also i–preferred. The conflict–free extensions in $iclo - F^{BF}$ w.r.t. X are the same as the +conflict–free ones w.r.t. X of BF. Since (a, c) and (d, e) are self–attackers, they will not appear in any sets. The admissible extensions of our AF are now $\emptyset$ and $\{b, d, e\}$. Argument a is not defended by any conflict–free set and b requires the presence of e, Since a is not defended, then neither is c due to the (a, c) attack. Finally, d requires the presence of b (note that (a, c) is a self–attacker) and e cannot be accepted without d due to the (d, e) attack. Therefore, our answers coincide with the i–admissible and i–preferred extensions of BF.

Let $X = R^{sup} \cup R^{med}_{R^{sup}}$ consist of supported and super–mediated attacks. The +conflict–free extensions are the same as in the case of direct attacks, with the exception of $\{a, d\}$. The d–admissible extensions are now $\emptyset$, $\{b\}$, $\{e\}$, $\{b, d\}$, $\{b, e\}$, $\{b, d, e\}$. From them, $\emptyset$, $\{b\}$, $\{b, d\}$ and $\{b, d, e\}$ are i–admissible. $\{b, d, e\}$ is the only i–preferred extension. Let us now consider the admissible sets of $iclo - F^{BF}$ w.r.t. X. The arguments a and c are still not defended by any conflict–free sets; b can defend itself, d requires the presence of b and e of d. Hence, our extensions are $\emptyset$, $\{b\}$, $\{b, d\}$ and $\{b, d, e\}$, which coincides with the i–admissible sets of BF.

Finally, let $X = R^{sec} \cup R^{ext}$ consist of secondary and extended attacks. From the previously listed +conflict–free sets w.r.t. direct attacks, we need to exclude the set $\{c, e\}$. The new d–admissible extension are $\emptyset$, $\{c\}$, $\{e\}$, $\{a, c\}$, $\{b, e\}$, $\{d, e\}$ and $\{b, d, e\}$. From this, $\emptyset$, $\{a, c\}$, $\{d, e\}$ and $\{b, d, e\}$ are i–admissible. This gives us two i–preferred extensions $\{a, c\}$ and $\{b, d, e\}$. We can now shift to our AF. We can observe that c protects itself against b and defends a, which in turn defends c from (a, c). Consequently, $\{a, c\}$ is one of our admissible extensions. Similarly, e attacks c and d attacks (d, e), making $\{d, e\}$ admissible as well. Finally, b and e attack a and c, which we can use to show that $\{b, d, e\}$ is another admissible set. This, along with $\emptyset$, gives us all of the extensions of $iclo - F^{BF}$. We can see that these are the answers we expected.

We can observe that the stable semantics are not necessarily preserved for the same reasons as in the self–attacker consistency form (see Section 4.4.2). The normal closure can be proved in a similar manner:

Translation 55. *Let $BF = (A, R, S)$ be a BAF and $R' \subseteq R^{ind}$ a collection of indirect attacks in BF. The associated closure attack propagation AF w.r.t. R' is $clo - F^{BF} = (A', R'')$, where $A' = A \cup S$ and $R'' = R \cup \bigcup R' \cup \{(x, x) \mid x \in S\} \cup \{(a, (b, a)), ((b, a), b) \mid a, b \in A, (b, a) \in S\}$.*

Theorem 9.5. *Let $BF = (A, R, S)$ be a BAF, $R' \subseteq R^{ind}$ a collection of indirect attacks in BF and $clo - F^{BF} = (A', R'')$ its associated closure attack propagation AF w.r.t. R' obtained through Translation 55. A set of arguments $E \subseteq A$ is c–admissible (c–preferred) in BF w.r.t. (R', R') iff it is admissible (preferred) in $clo - F^{BF}$.*

Please note that our analysis here is by no means exhaustive. Moreover, we have not yet established an abstract construction (i.e. independent of the chosen indirect conflicts) that would enforce safety of our extensions, even though some results for special cases are available [30]. Additionally, due to lack of appropriate i–/c–/s–complete and grounded semantics, our results are still limited. Nevertheless, this task needs to be left for future work, and for now we refer the reader for a more in–depth discussion to the original papers [30,31]. We close this section by putting the available results into our classification system.

Redefinition of Translation 53: *Let Fr^{BAF} be the collection of all BAFs and Fr^{AF} the collection of all AFs, both on domain $\mathcal{U}$. The translation $att\text{-}Tr^{BAF}_{AF} : Fr^{BAF} \to Fr^{AF}$ is defined as $att\text{-}Tr^{BAF}_{AF}((A, R, S)) = (A, R \cup \bigcup R')$ for a framework $(A, R, S) \in Fr^{BAF}$, where $R' \subseteq R^{ind}$ is a chosen set of indirect conflicts in (A, R, S).*

Redefinition of Theorem 9.1: *Let $\sigma^{BAF} \in \{$+conflict–free, d–admissible, d–preferred, d–complete, d–grounded, stable$\}$ be a BAF semantics with identical parametrization [22] and let $\sigma^{AF} \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$ be a similar AF semantics. Let SC^{Tr}_{σ} be the identity casting functions for σ. The translation $att\text{-}Tr^{BAF}_{AF}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*

Analysis of Translation 53: Under the (+) conflict–free, (d–) admissible, (d–) preferred, (d–) complete, (d–) grounded and stable semantics with identical parametrization and identity casting functions, the translation $att\text{-}Tr^{BAF}_{AF}$ is:

- full, surjective and overlapping
- argument domain preserving, attack relation introducing and support relation removing
- generic, semantics domain preserving and exact
- semi–structural

Our approach is not modular. Translation $att\text{-}Tr^{BAF}_{AF}$ is classified as basic–attack propagation hybrid under the listed semantics and casting functions.

Explanation. Any BAF can undergo the translation, and thus our approach is full. Moreover, given any target AF, we can simply add an empty set for the support relation and receive a possible source BAF. Thus, the translation is also surjective. Unfortunately, it is also overlapping. For example, two BAFs with the same set of arguments, empty set of attacks and different set of supports, will be translated into a single AF.

The translation is both argument and semantics domain preserving. It is also attack introducing, as previously indirect conflicts become direct and visible in the structure of the framework. The translation is however support relation removing; arguments can become completely detached and there is no way of telling whether there has been a positive interaction between them or not.

[22] By identical parametrization we understand that we use the same set $R' \subseteq R^{ind}$ or pair (R', R') when applicable.

Due to the amount of handled semantics, even though with identical parametrization, we classify the approach is generic. As indirect attacks are a semantical concept, even though not necessarily a computationally difficult one, the translation cannot be classified as structural. Therefore, we choose to see it as a semi–structural one. Its exactness under the described semantics follows from Theorem 9.1.

Unfortunately, our translation is in no way modular. For any (nonempty) type of parametrization we can find a suitable counterexample. It suffices to separate a given BAF in a way that the attack relations stay in one framework, and support in the other – as a result we will obtain an AF corresponding to the attack–bases subgraph without any conflicts propagated. For example, let us look at two BAFs $BF_1 = (\{a,b\}, \{(a,b)\}, \emptyset)$ and $BF_1 = (\{b,c\}, \emptyset, \{(b,c)\})$ and focus on secondary attack. Our structures are translated to frameworks $(\{a,b\}, \{(a,b)\})$ and $(\{b,c\}, \emptyset)$ respectively. Their union is simply $(\{a,b,c\}, \{(a,b)\})$, while the AF associated with $BF_1 \cup BF_2$ is $(\{a,b,c\}, \{(a,b),(a,c)\})$. Thus, the indirect attack is "lost" and our translation is not modular. ■

Redefinition of Translations 54 and 55: *Let Fr^{BAF} be the collection of all BAFs on domain $\mathcal{U}$ and Fr^{AF} the collection of all AFs on domain $\mathcal{U} \cup (\mathcal{U} \times \mathcal{U})$. The translation $iclo\text{-}Tr_{AF}^{BAF} : Fr^{BAF} \rightarrow Fr^{AF}$ is defined as $iclo\text{-}Tr_{AF}^{BAF}((A,R,S)) = (A', R'')$ for a framework $(A,R,S) \in Fr^{BAF}$, where $A' = A \cup S$, $R'' = R \cup \bigcup R' \cup \{(x,x) \mid x \in S\} \cup \{(b,(b,a)),((b,a),a) \mid a,b \in A, (b,a) \in S\}$ and $R' \subseteq R^{ind}$ is a chosen set of indirect conflicts in (A,R,S).*

The translation $clo\text{-}Tr_{AF}^{BAF} : Fr^{BAF} \rightarrow Fr^{AF}$ is defined as $clo\text{-}Tr_{AF}^{BAF}((A,R,S)) = (A', R'')$ for a framework $(A,R,S) \in Fr^{BAF}$, where $A' = A \cup S$, $R'' = R \cup \bigcup R' \cup \{(x,x) \mid x \in S\} \cup \{(a,(b,a)),((b,a),b) \mid a,b \in A, (b,a) \in S\}$ and $R' \subseteq R^{ind}$ is a chosen set of indirect conflicts in (A,R,S).

Redefinition of Theorems 9.4 and 9.5: *Let $\sigma^{BAF} \in \{$i–admissible, i–preferred$\}$ be a BAF semantics with identical parametrization* [23] *and let $\sigma^{AF} \in \{$admissible, preferred$\}$ be a similar AF semantics. Let SC_{σ}^{Tr} be the identity casting functions for σ. The translation $iclo\text{-}Tr_{AF}^{BAF}$ is strong and semantics bijective under $(\sigma, SC_{\sigma}^{Tr})$.*

Let $\delta^{BAF} \in \{$c–admissible, c–preferred$\}$ be a BAF semantics with identical parametrization and let $\delta^{AF} = \sigma^{AF}$ be a similar AF semantics. Let SC_{δ}^{Tr} be the identity casting functions for δ. The translation $clo\text{-}Tr_{AF}^{BAF}$ is strong and semantics bijective under $(\delta, SC_{\delta}^{Tr})$.

Analysis of Translation Translations 54 and 55: Under the (i–/c–) admissible and (i–/c–) preferred semantics with identical parametrization and identity casting functions, the translations $iclo\text{-}Tr_{AF}^{BAF}$ and $clo\text{-}Tr_{AF}^{BAF}$ are:

- full, target–subclass and overlapping
- argument domain altering, argument introducing and attack relation introducing

[23] By identical parametrization we understand that we use the same set $R' \subseteq R^{ind}$ or pair (R', R') when applicable.

- specialized, semantics domain preserving and exact
- semi–structural

Our approach is not modular. Translations $iclo\text{-}Tr^{BAF}_{AF}$ and $clo\text{-}Tr^{BAF}_{AF}$ are classified as basic–attack propagation–defender hybrids under the listed semantics and casting functions.

Explanation. Similarly as in the previous translation, our approaches are full. Unfortunately, they are no longer surjective due to the domain change. We can easily create a framework containing only $\mathcal{U} \times \mathcal{U}$ type of arguments and observe that it could not have been produced by any BAF. Nevertheless, any types of attack patterns can appear. Similarly as in the previous translation, our approaches is overlapping. For example, we can consider two BAFs $(\{a, b, c\}, \{(a, b)\}, \{(b, c)\})$ and $(\{a, b, c\}, \{(a, b), (a, c)\}, \{(b, c)\})$, where the latter represents the first framework completed with secondary attack. Both of them will be translated into the same inverse closure AF $(\{a, b, c, (b, c)\}, \{(a, b), (a, c), (b, (b, c)), ((b, c), c)\})$. We can create a similar example for the normal closure translation.

The translations are clearly argument domain altering, even though they are semantics domain preserving. They are also attack introducing, as previously indirect conflicts become direct and visible in the structure of the framework. Moreover, additional arguments cause the creation of their respective attacks as well. The translations are no longer support relation removing, as the supports are represented with auxiliary arguments and arguments connected previously by support are now tied with defense. Exactness follows easily from Theorems 9.4 and 9.5.

Our translations do not handle too many semantics; thus, we classify them as specialized. For reasons similar as in Translation 53, we lose modularity and classify our approach as semi–structural. ∎

9.1.2 Coalition Translation

The coalition approach was the first attempt at translating BAFs into AFs. The original version [29] joined arguments into groups meeting certain support and conflict–freeness requirements, and then used these groups as new AF arguments:

Definition 9.6. Deprecated Let $BF = (A, R, S)$ be a BAF and $BF^{att} = (A, R)$ the AF representing the attack subgraph of BF. A set $C \subseteq A$ is a **coalition** of BF iff:

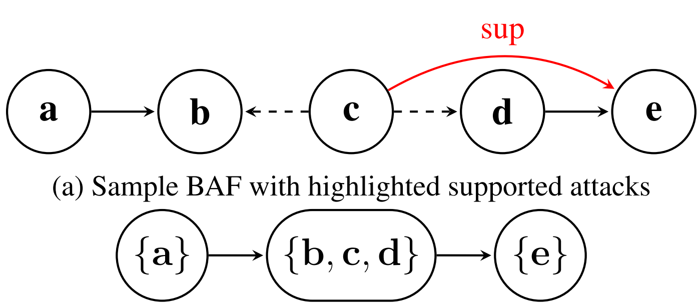
- the support subgraph $(C, S \cap (C \times C))$ induced by C is connected,
- C is conflict–free in BF^{att}, and
- C is maximal under $\subseteq$ among the sets satisfying the previous two conditions.

Translation 56. ***Faulty Translation*** *Let $BF = (A, R, S)$ be a BAF. The associated coalition AF is defined as c-$F^{BF} = (A', R')$, where $A' = \{C \mid C$ is a coalition of $BF\}$ and $R' = \{(C_1, C_2) \mid \exists a_1 \in C_1, a_2 \in C_2$ s.t $a_1 R a_2\}$.*

Unfortunately, as observed in various works [19, 29, 30], this approach does not preserve the behaviour of the semantics. There are two main reasons for this situation. One is the conflict–freeness assumption within the coalitions, which led to the loss of certain arguments in the translation. Another is the lack of a more precise interpretation of abstract support, which in turn caused the creation of coalitions in a way that a direct attack on a single member of it did not translate to an appropriate indirect attacks on the remaining members. This means that the defense in the produced AF did not correspond to defense in the source BAF. Let us look at an example:

Example 109. We can consider a simple AF–style BAF $BF_1 = (\{a, b\}, \{(b, a), (b, b)\}, \emptyset)$. We can observe that $\{a\}$ cannot be an admissible extension of any type in BF_1. Nevertheless, $\{\{a\}\}$ is an admissible set of the associated coalition AF $F^{BF_1} = (\{\{a\}\}, \emptyset)$. This framework will also produce a preferred extension not representing any set that BF_1 can create. Additionally, while BF_1 does not have stable extensions, the associated AF does.

These issues will appear even if we consider BAFs without self–attackers. Let $BF_2 = (\{a, b, c, d, e\}, \{(a, b), (d, e)\}, \{(c, b), (c, d)\})$ be a BAF from [29], now depicted in Figure 91a. The attacks originally considered for this framework were the secondary and supported ones; we can observe that the only indirect conflict in this case is the supported attack from c to e. Neither c nor d are in any way attacked in BF_2. The coalitions for this framework are $C_1 = \{a\}$, $C_2 = \{b, c, d\}$ and $C_3 = \{e\}$. So, the associated coalition AF is $F^{BF_2} = (\{C_1, C_2, C_3\}, \{(C_1, C_2), (C_2, C_3)\})$ (see Figure 91b). The sets $\emptyset$, $\{C_1\}$ and $\{C_1, C_3\}$ are admissible in F^{BF_2}, with $\{C_1, C_3\}$ being the stable, preferred and grounded set. This translates to the set $\{a, e\}$, which, based on the previous remark on c and d, cannot possibly be contained in an admissible, preferred, stable or grounded extension of any type in BF_2 w.r.t. the supported and secondary attacks.



(a) Sample BAF with highlighted supported attacks

(b) Original coalition AF

Figure 91: Sample BAF with associated coalition AF based on Translation 56

Therefore, we turn our heads to the newer versions of the coalition translation, introduced in [30]. Its focus was on translating the deductive support and thus, indirectly, also the necessary one. The new definition of a coalition and the translation are now as follows:

Definition 9.7. Let $BF = (A, R, S)$ be a BAF with S being a set of deductive supports. The **deductive coalition** associated with an argument $a \in A$ is defined as $C(a) = \{a\} \cup \{b \mid a \text{ supports } b\}$.

We can observe that the conflict–freeness restriction from Definition 9.6 has been dropped, and the ones concerning support has been transformed into requiring that every node reachable from a through the support edges appears in $C(a)$.

Translation 57. *Let $BF = (A, R, S)$ be a BAF. The associated coalition AF for deductive support is defined as $cd - F^{BF} = (A', R')$, where $A' = \{C(a) \mid a \in A\}$ and $R' = \{(C_1(a), C_2(b)) \mid \exists a_1 \in C_1(a), a_2 \in C_2(b) \text{ s.t } a_1 R a_2\}$.*

The results concerning the semantics of the resulting AF are given by its connection to the attack propagation AF for deductive support [30]:

Theorem 9.8. ***Deprecated*** *Let $BF = (A, R, S)$ be a deductive BAF and $R' = \{R^{sup}, R^{med}_{R^{sup}}\} \subseteq R^{ind}$ the collection of supported and super–mediated attacks in BF. Let $apd - F^{BF} = (A, R \cup \bigcup R')$ be the associated attack propagation AF and $cd - F^{BF} = (A', R'')$ the associated coalition AF. A set $E = \{a_1, ..., a_n\} \subseteq A$ is a σ–extension of $apd - F^{BF}$, where $\sigma \in \{$conflict–free, admissible, stable$\}$ iff $\{C(a_1), C(a_2), ..., C(a_n)\}$ is a σ–extension of $cd - F^{BF}$.*

Unfortunately, the results are not entirely correct. Due to the fact that the arguments in the coalition AF are simply sets of BAF arguments and carry no information as to who created them (indeed, in the presence of cycles one coalition can be created by multiple arguments), we have to resort to union casting function to relate the source and target extensions. This means that not every conflict–free or admissible extension of the attack propagation approach has a corresponding conflict–free or admissible one in the coalition framework. Thus, this relation is not strong, let alone one–to–one. Although for a given argument $a \in A$ we produce a single coalition, there can be multiple coalitions containing it. This means that to a single conflict–free or admissible extension in the attack propagation approach, an arbitrary number (including zero) of conflict–free or admissible extensions can be related:

Example 110. Let $(\{a, b\}, \emptyset, \{(a, b)\})$ be a BAF for deductive support. The associated attack propagation AF is simply $(\{a, b\}, \emptyset)$ and its conflict–free (admissible) extensions are $\emptyset$, $\{a\}$, $\{b\}$ and $\{a, b\}$. The coalition AF is $(\{\{a, b\}, \{b\}\}, \emptyset)$ and its conflict–free (admissible) extensions are $\emptyset$, $\{\{b\}\}$, $\{\{a, b\}\}$ and $\{\{a, b\}, \{b\}\}$. We can observe that the set $\{a, b\}$ can be retrieved both from $\{\{a, b\}\}$ and $\{\{a, b\}, \{b\}\}$. Thus, the relation between the extensions is not one–to–one. Moreover, the set $\{a\}$ in the attack propagation AF is not represented by any coalition extension.

By analyzing the coalition arguments it can be observed that $\{a, b\}$ was created for a and $\{b\}$ for b. Thus, one can argue that the extension $\{a\}$ corresponds to $\{\{a, b\}\}$ and that a certain relation is in fact preserved. However, this is not a long–term strategy. Let us consider the framework $(\{a, b\}, \emptyset, \{(a, b), (b, a)\})$ containing a support cycle. We obtain a single coalition argument $\{a, b\}$, originating both from a and b. In this case, all of the sets $\{a\}$, $\{b\}$ and $\{a, b\}$ would be admissible in the associated attack propagation AF. Therefore, the "choose any coalition source" approach retrieves the desired extensions. However, only $\{a, b\}$ is complete, and it is the "use all sources" approach that produces it. Hence, obtaining the extensions of the attack–propagation framework from the coalition one by trying to find out what argument created a given coalition needs different approaches for different semantics.

The reason for this situation is quite simple; as we have observed before, the attack propagation approach is related to the d–family of BAF semantics. However, the use of coalition enforces the closure property and is thus related more to the c–family of semantics, and clearly the extensions of the two types are not always the same. Nevertheless, as stated before, some of the indirect attacks can be used to enforce closure. Therefore, although the provided results are not correct for conflict–free and admissible semantics, they are so when we consider approaches enforcing some notions of maximality. Moreover, unlike in the original results, the union of the coalition arguments in an extension will correspond to the actual extension, i.e. $\{a_1, ..., a_n\} = \bigcup_{i=1}^{n} C(a_i)$.

Theorem 9.9. *Let $BF = (A, R, S)$ be a deductive BAF and $R' = \{R^{sup}, R^{med}_{R^{sup}}\} \subseteq R^{ind}$ the collection of supported and super–mediated attacks in BF. Let $apd - F^{BF} = (A, R \cup \bigcup R')$ the associated attack propagation AF and $cd - F^{BF} = (A', R'')$ the associated coalition AF obtained through Translations 53 and 57. If set $E = \{a_1, ..., a_n\} \subseteq A$ is a σ–extension of $apd - F^{BF}$, where $\sigma \in \{$complete, preferred, grounded, stable$\}$, then $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ is a σ–extension of $cd - F^{BF}$. If set $E' \subseteq A'$ is a σ–extension of $cd - F^{BF}$, then $E = \bigcup E'$ is a σ–extension of $apd - F^{BF}$.*

We can now finally tie the extensions of the coalition AF back to BAF, not to another of its translations. Although due to lack of appropriate semantics some of the results will be given for the d–family rather than c–family, it should be noted that complete extensions are in fact closed under support due to Theorem 9.2. Again, please note that the union of the coalition arguments in an extension will correspond to the actual extension, i.e. $\{a_1, ..., a_n\} = \bigcup_{i=1}^{n} C(a_i)$. This will hold even for the admissible semantics due to the fact that we follow the c–family:

Theorem 9.10. *Let $BF = (A, R, S)$ be a deductive BAF, $R' = \{R^{sup}, R^{med}_{R^{sup}}\} \subseteq R^{ind}$ the collection of supported and super–mediated attacks in BF and $cd - F^{BF} = (A', R'')$ the associated coalition AF obtained through Translation 57. The following holds:*

- *if set $E = \{a_1, ..., a_n\} \subseteq A$ is +conflict–free w.r.t. R' and closed under S in BF, then $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ is a conflict–free extension of $cd - F^{BF}$.*

- *if set $E = \{a_1, ..., a_n\} \subseteq A$ is a c–admissible (c–preferred) extension of BF w.r.t. (R', R'), then $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ is an admissible (preferred) extension of $\mathrm{cd} - F^{BF}$.*

- *if set $E = \{a_1, ..., a_n\} \subseteq A$ is a d–complete extension of BF w.r.t. (R', R'), then $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ is a complete extension of $\mathrm{cd} - F^{BF}$.*

- *if set $E = \{a_1, ..., a_n\} \subseteq A$ is a d–grounded (stable) extension of BF w.r.t. R', then $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ is a grounded (stable) extension of $\mathrm{cd} - F^{BF}$.*

- *if set $E' \subseteq A'$ is a conflict–free extension of $\mathrm{cd} - F^{BF}$, then $E = \bigcup E'$ is +conflict–free w.r.t. R' and closed under S in BF.*

- *if set $E' \subseteq A'$ is an admissible (preferred) extension of $\mathrm{cd} - F^{BF}$, then $E = \bigcup E'$ is a c–admissible (c–preferred) extension of BF w.r.t. (R', R').*

- *if set $E' \subseteq A'$ is a complete extension of $\mathrm{cd} - F^{BF}$, then $E = \bigcup E'$ is a d–complete extension of BF w.r.t. (R', R').*

- *if set $E' \subseteq A'$ is a grounded (stable) extension of $\mathrm{cd} - F^{BF}$, then $E = \bigcup E'$ is a d–grounded (stable) extension of BF w.r.t. R'.*

Please note that the coalition translation for necessary support would be the same as for deductive; it is only the used definition of a coalition that would change. Instead of building it on closure, we would have to use the inverse version. However, we will omit further analysis and proceed with analyzing the existing results. Please note that we are not entirely sure on how to perceive the changes done to the original relations:

Redefinition of Translation 57: *Let Fr^{BAF} be the collection of all BAFs on domain $\mathcal{U}$ and Fr^{AF} the collection of all AFs on domain $2^{\mathcal{U}}$. The translation cd-$Tr_{AF}^{BAF} : Fr^{BAF} \rightarrow Fr^{AF}$ is defined as cd-$Tr_{AF}^{BAF}((A, R, S)) = (A', R')$ for a framework $(A, R, S) \in Fr^{BAF}$, $A' = \{C(a) \mid a \in A\}$ and $R' = \{(C_1(a), C_2(b)) \mid \exists a_1 \in C_1(a), a_2 \in C_2(b)$ s.t $a_1 R a_2\}$.*

Redefinition of Theorem 9.10: *Let $\sigma^{BAF} \in \{$+conflict–free and closed under support, c–admissible, c–preferred, d–complete, d–grounded, stable$\}$ be a BAF semantics with identical parametrization consisting of supported and super mediated attacks and let $\sigma^{AF} \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$ be a similar AF semantics. Let SC_{σ}^{Tr} be the union casting functions for σ. The translation cd-Tr_{AF}^{BAF} is strong under $(\sigma, SC_{\sigma}^{Tr})$. It is semantics bijective under (d–) complete, (c–) preferred, (d–) grounded and stable semantics.*

Analysis of Translation 57: Under the (+) conflict–free (and closed under support), (c–) admissible, (c–) preferred, (d–) complete, (d–) grounded and stable semantics with identical parametrization consisting of supported and super mediated attacks, and union casting functions, the translation *cd-*Tr_{AF}^{BAF} is:

- full, target–subclass and overlapping

- argument domain altering, argument introducing, attack relation introducing (or removing) and support relating introducing (or removing)
- generic and semantics domain altering
- semi–structural

Our approach is not modular. Translation $cd\text{-}Tr_{AF}^{BAF}$ is classified as basic–coalition hybrid under the listed semantics and casting functions.

Explanation. Any BAF can undergo the translation and thus we consider it full. It is unfortunately target–subclass due to the domain change. Not every set of coalitions can be produced. Let us consider the sets $C_1 = \{a, b\}$, $C_2 = \{b, c\}$ and $C_3 = \{a, c\}$. The original set of argument is thus $\{a, b, c\}$ and every coalition has to be produced by one argument. We have thus two combinations; either $C_1 = C(a)$, $C_2 = C(b)$ and $C_3 = C(c)$, or $C_1 = C(b)$, $C_2 = C(c)$ and $C_3 = C(a)$. The first option means that aSb, bSc and cSa, and the conclusion is that $\{a, b, c\}$ should have been the only produced coalition. We reach a contradiction. The other option means that bSa, aSc and cSb. Again, we obtain a cycle, just in a different direction, and $\{a, b, c\}$ should have been the only coalition.

In order to show it is overlapping, let us consider the frameworks $(\{a, b, c\}, \emptyset, \{(a, b), (b, c)\})$ and $(\{a, b, c\}, \emptyset, \{(a, b), (b, c), (a, c)\})$. The latter basically changes the indirect support from the former structure into a direct one. They will both be assigned a simple framework $(\{\{a, b, c\}, \{b, c\}, \{c\}\}, \emptyset)$.

Clearly, both argument and semantics domain are altered. Due to the fact that one argument can appear in multiple coalitions, it can be represented by more than a single argument, and we classify the translation as argument introducing. We are not sure whether to classify the translation as relation removing or introducing due to the fact that although we know a given relation occurs, we do not always know whether it is direct, indirect, and who carried it out. Although a given attack or support is in a sense represented in the target framework, the way arguments themselves are represented, makes the approach imprecise. In other words, depending on the way we attempt to reconstruct the original framework, it might have somewhat different relations than the original one. As observed in the previous paragraph, the frameworks $F_1 = (\{a, b, c\}, \emptyset, \{(a, b), (b, c), (c, a)\})$ and $F_2 = (\{a, b, c\}, \emptyset, \{(b, a), (c, b), (a, c)\})$ would be translated simply into $(\{\{a, b, c\}\}, \emptyset)$. Therefore, depending on how we reconstruct the original structure, we can end up with F_1 instead of F_2, vice versa, or any other framework completing the support graph by changing indirect supports to direct ones. A similar analysis can be carried out in case of attack; if we have two coalitions $\{a, b\}$ and $\{c\}$ and attack $(\{a, b\}, \{c\})$, we do not know which of the a and b is responsible for the direct conflict. Due to the fact that they would be in a support cycle, direct conflict from one implies a supported attack from the other. Therefore, again, depending on how we want to reconstruct the original BAF, we might change indirect attacks into direct or the other way around, thus leading to relation introduction and removal respectively. We thus leave it to the reader to decide which classification feels more intuitive.

Due to the amount of handled semantics we classify the approach as generic. Based on the construction of the coalitions, we also choose to mark it as semi–structural. In order to show the lack of modularity, let us look at the frameworks $(\{a, b\}, \emptyset, \{(a, b)\})$ and $(\{b, c\}, \emptyset, \{(b, c)\})$. The union of their translations is $(\{\{a, b\}, \{b\}, \{b, c\}, \{c\}\}, \emptyset)$, while the translation of their union is $(\{\{a, b, c\}, \{b, c\}, \{c\}\}, \emptyset)$. ■

9.1.3 Defender Translation

The defender–like translation from BAFs specialized with necessary support to AFs has been proposed in [31]. The supporting link becomes a new argument, which itself attacks the target and is attacked by the source of the edge it represents:

Translation 58. *Let $BF = (A, R, S)$ be a BAF. The associated defender AF for necessary support is defined as $dn - F^{BF} = (A', R')$, where $A' = A \cup S$ and $R' = R \cup \{(b, (b, a)) \mid a, b \in A, (b, a) \in S\} \cup \{((b, a), a) \mid a, b \in A, (b, a) \in S\}$.*[24]

The original results concerning the semantics are somewhat limited. This is due to the fact that the purpose of the translation was to study certain constraints, not to show the correspondence between given extensions:

Theorem 9.11. *Let $BF = (A, R, S)$ be a BAF specialized for necessary support and $dn - F^{BF} = (A', R')$ its associated defender AF obtained through Translation 58. If $E \subseteq A'$ is admissible in $dn - F^{BF}$, then $E \cap A$ is inverse closed under support in BF.*

Unfortunately, for the general BAFs, it appears that little more can proved. We can observe that the presented construction is very similar to the ones we have seen before already for SETAFs, AFRAs and EAFs. Thus, it is natural to ask why would this, rather standard, defender translation suddenly misbehave for BAFs. The reason for it is support in general and support cycles in particular:

Example 111. Let $BF_1 = (\{a, b, c\}, \{(b, c)\}, \{(a, b), (b, a)\})$ be a BAF. Its associated defender AF is $F^{BF_1} = (\{a, b, c, (a, b), (b, a)\}, \{(b, c), ((a, b), b), ((b, a), a), (a, (a, b)), (b, (b, a))\})$. We can observe that $\{c\}$ is not a d–admissible extension of BF_1 w.r.t. secondary and extended attacks. For example, it cannot defend itself against the direct attack from b. However, $\{c, (a, b), (b, a)\}$ is an admissible extension of our AF. It is also preferred and stable in F^{BF_1}, while $\{c\}$ is neither of these things in BF_1.

This issue has been caused by the fact that arguments representing supports, being the attackers of a and b, could have been accepted despite the fact that neither b nor a are attacked. In this case, it can be addressed by turning (a, b) and (b, a) into self–attackers, as was e.g. done in Translations 54 and 55. However, this is not a long term strategy, as the next frameworks will show.

[24]Please note that it would be somehow more natural to use "b does not support a" statements rather than "b supports a" in this translation. Nevertheless, we keep the original sound of the transformation.

Let $BF_2 = (\{a, b, c\}, \{(b, c), (c, a)\}, \{(a, b), (b, a)\})$ be a modification of BF_1. The secondary attacks in this framework are (c, b) and repeated (c, a), while (a, c) and repeated (b, c) are extended. Therefore, $\{c\}$ is a d–admissible in BF_2 w.r.t. the secondary and extended attacks. The associated defender AF is $F^{BF_2} = (\{a, b, c, (a, b), (b, a)\}, \{(b, c), (b, a), ((a, b), b), ((b, a), a), (a, (a, b)), (b, (b, a))\})$. We can observe that e.g. $\{c, (b, a), (a, b)\}$ is correctly recognized as an admissible extension. If (a, b) was turned into a self–attacker, c would not be defended against b and would not appear in any admissible set of F^{BF_2}.

Let $BF_3 = (\{a, b, c\}, \{(a, b), (c, a)\}, \{(b, c)\})$ be a BAF. There are no extended attacks in this framework, only the secondary one (a, c). We can observe that thanks to this, $\{a\}$ is an admissible extension of any type of BF_1 w.r.t. the secondary and extended conflicts. If the associated defender AF was created using the aforementioned self–attack technique, the resulting framework would be $F^{BF_3} = (\{a, b, c, (b, c)\}, \{(a, b), (c, a), (b, (b, c)), ((b, c), c), ((b, c), (b, c))\})$. This means that the only argument attacking c cannot be accepted and no set containing a will be an admissible extension of F^{BF_3}. Only by removing the self–attack restriction we are able to obtain the extension $\{a, (b, c)\}$ and retrieve the desired answer.

The above example shows that in order to use the defender translation we either need to limit ourselves to BAFs that do not have support cycles, or use the self–attacker method and mix it with attack propagation (similarly as in Translation 54), or make a distinction between the treatment of links participating in support cycles, those that do not, and how indirect attacks fit into all of this. Already in BAFs, with their indiscriminating approach towards support cycles, we can observe that a defender translation capable of working with all types of frameworks would not be structural.

Nevertheless, our contribution to the existing results is minor and further analysis is left for future work. We also choose not to include this approach in our system and refer the reader to Section 10.2.2 to see how necessary support in AFNs can be translated into AFs using the defender approach.

Theorem 9.12. *Let $BF = (A, R, S)$ be a BAF specialized for necessary support, $R' = \{R^{sec}, R^{ext}\}$ the collection of secondary and extended attacks in BF and $dn - F^{BF} = (A', R'')$ the defender AF associated with BF obtained through Translation 58. If $E \subseteq A$ is an i–admissible extension of BF w.r.t. (R', R'), then there exists an admissible extension $E' \subseteq A'$ s.t. $E' \cap A = E$. If $E' \subseteq A'$ is an admissible extension of $df - F^{BF}$, then $E = E' \cap A$ might not be an i–admissible extension of BF w.r.t. (R', R').*

9.2 BAF as AFN

Although in the previous sections we could have observed how BAFs handle e.g. deductive and necessary support, it is interesting to ask how the extensions of a given specialized BAF are in relation to the actual framework built for the relevant relation. In this section we will discuss the similarities and differences between necessary support BAFs and AFNs.

One important thing we need to state is the fact that the ways support cycles are handled in both frameworks are completely different:

Example 112. Let $BF = (\{a, b, c\}, \{(b, c)\}, \{(a, b), (b, a)\})$ be a BAF. We specialize it for necessary support, i.e. use secondary attacks. In this case, there are no such indirect conflicts. The only two i–admissible extensions of this framework are $\emptyset$ and $\{a, b\}$. They are also c– and s–admissible. The d–admissible sets are $\emptyset$, $\{a\}$, $\{b\}$ and $\{a, b\}$. The preferred extension (of any type), stable and d–grounded extension is $\{a, b\}$. Please observe that the answer would have been the same if we used both secondary and extended attacks; in this case, (a, c) and repeated (b, c) would be the indirect conflicts.

Let $FN = (\{a, b, c\}, \{(b, c)\}, \{(\{a\}, b), (\{b\}, a)\})$ be an AFN of the same structure. We can observe that neither b nor a possess a powerful sequence in FN. Consequently, $\emptyset$ and $\{c\}$ are the only admissible extensions of FN, with $\{c\}$ being the sole complete, preferred, grounded and stable extension. This is not in correspondence with any of the extensions produced by BF.

Please note it does not mean we criticize any of the approaches. Arguments both for and against any type of handling of support cycles can be found. However, as a consequence, we need to assume a certain "common ground" to build a translation between BAFs and AFNs. Thus, in what follows, we will focus on such support acyclic BAFs.

Translation 59. *Let $BF = (A, R, S)$ be a support acyclic BAF. The associated AFN is $FN^{BF} = (A, R, N)$, where $N = \{(\{a\}, b) \mid (a, b) \in S\}$.*

The produced AFNs exhibit a number of desirable properties. Clearly, it will be both binary and support acyclic itself. Therefore, it satisfies various normal forms (see Theorem 4.71):

Theorem 9.13. *Let $BF = (A, R, S)$ be a support acyclic BAF and $FN^{BF} = (A, R, N)$ be its associated AFN obtained through Translation 59. Then FN^{BF} is support binary and acyclic. It is also in minimal, weakly, relation and strongly valid forms.*

Please note that the produced AFN might not necessarily be consistent. Due to the fact that the AFN semantics are built around the notion of coherence, which, among other things, requires the presence of supporters of a given argument in an extension, it is the i–family of semantics we will use in our analysis. However, our choice is to parametrize the semantics only with secondary attacks, not both secondary and extended ones. The extended conflicts did not explicitly appear in the newer versions of AFNs. Moreover, as observed in the definition of AFN semantics, the defense of an argument relies on attacking every coherent set of a given attacker. This corresponds much more to the interpretation of the secondary attack rather than extended. Our results are thus the following:

Theorem 9.14. *Let $BF = (A, R, S)$ be a support acyclic BAF, $R' = \{R^{sec}\}$ the collection of secondary attacks in BF and $FN^{BF} = (A, R, N)$ the AFN associated with BF obtained through Translation 59. Then, a set $E \subseteq A$ is:*

- *+conflict–free w.r.t. $\emptyset$ in BF iff it is conflict–free in FN^{BF}.*

- *inverse closed under S in BF iff it is coherent in FN^{BF}.*

- *+conflict–free w.r.t. R' and inverse closed under S in BF iff it is strongly coherent in FN^{BF}.*

- *an i–admissible extension of BF w.r.t. (R', R') iff it is admissible in FN^{BF}.*

- *an i–preferred extension of BF w.r.t. (R', R') iff it is preferred in FN^{BF}.*

- *a d–complete extension of BF w.r.t. (R', R') iff it is complete in FN^{BF}.*

- *a d–grounded extension of BF w.r.t. R' iff it is grounded in FN^{BF}.*

- *a stable extension of BF w.r.t. R' iff it is stable in FN^{BF}.*

We can now put the translation into our classification system. Although our AFNs may not necessarily be well–structured, they are elementary, though the depth of the support depends on the depth of the support in the source BAFs. Therefore, we will leave the depth unspecified.

Redefinition of Translation 59: *Let $SAcy^{BAF}$ be the collection of all support acyclic BAFs and $SEle^{AFN}$ the collection of elementary AFNs, both on domain $\mathcal{U}$. The translation $Tr^{BAF}_{AFN} : SAcy^{BAF} \rightarrow SEle^{AFN}$ is defined as $Tr^{BAF}_{AFN}((A, R, S)) = (A, R, N)$ for a framework $(A, R, S) \in SAcy^{BAF}$, where $N = \{(\{a\}, b) \mid (a, b) \in S\}$.*

Redefinition of Theorem 9.14: *Let $\sigma^{BAF} \in \{$inverse closed, +conflict–free and inverse closed, i–admissible, i–preferred, d–complete, d–grounded, stable$\}$ be a BAF semantics with identical parametrization consisting of secondary attacks and let $\sigma^{AFN} \in \{$coherent, strongly coherent, admissible, preferred, complete, grounded, stable$\}$ be a similar AFN semantics. Let SC^{Tr}_{σ} be the identity casting functions for σ. The translation Tr^{BAF}_{AFN} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*

The translation is quite straightforward and its properties can be easily shown. Thus, we will omit further explanations.

Analysis of Translation 59: Under the (inverse closed) coherent, (+conflict–free and inverse closed) strongly coherent, (i–) admissible, (i–) preferred, (d–) complete, (d–) grounded and stable semantics with identical parametrization consisting of secondary attacks and identity casting functions, the translation Tr^{BAF}_{AFN} is:

- source–subclass, target–subclass and injective

- argument domain and structure preserving

- generic, semantics domain preserving and exact

- structural and modular

Translation Tr^{BAF}_{AFN} is classified as basic under the listed semantics and casting functions.

9.3 BAF as EAS

Although at first it was proposed that in order to handle evidential support, additional notions need to be introduced in BAFs [30], the results in [78] showed that this form of support is not far from necessary. Thus, the translation from BAFs to EASs will also be for semantics parametrized with secondary attack. Although the BAF–AFN translation has already dealt with these semantics in a very satisfactory manner, the BAF–EAS approach can be useful in highlighting the structural differences between the support in AFNs and EASs.

In BAFs, every argument is valid, i.e. its capability to carry out an attack does not depend on the fact whether the support it receives is acyclic or not. In EASs, it is not the case. Thus, just like in the previous section, we will need to assume that we are working with frameworks that have acyclic support graphs. However, while AFNs handled BAFs rather straightforwardly, EASs need to make some modifications. The validity of arguments requires not just the ability to derive them in an acyclic manner, but also tracing back to evidence argument (see [78] and Section 2.2.3). Therefore, suitable support relations need to be added to the framework. Finally, we can observe that in BAFs, it suffices to attack a single supporter of an argument in order to secondary attack the argument itself. This pairs well with the inverse closure, where accepting an argument means accepting every of its supporters. As a result, BAFs corresponded to binary AFNs. However, the support relation in EASs is structurally a bit different. If we were to create binary EASs, then accepting any of the supporters would be sufficient to derive an argument. What we need is in fact singular EASs, where the whole supporting set needs to be present in an extension. This brings us to the following translation:

Translation 60. *Let $BF = (A, R, S)$ be a support acyclic BAF. The associated EAS is $ES^{BF} = (A \cup \{\eta\}, R', E)$, where $R' = \{(\{a\}, b) \mid (a, b) \in R\}$ and $E = \{(\{\eta\}, a) \mid a \in A, \nexists c \in A$ s.t. $cSb\} \cup \{(S^a, a) \mid S^a$ is the collection of all arguments $b \in A$ s.t. $bSa\}$.*

Theorem 9.15. *Let $BF = (A, R, S)$ be a support acyclic BAF and let $ES^{BF} = (A \cup \{\eta\}, R', E)$ be its associated EAS obtained through Translation 60. ES^{BF} is attack binary, support singular and all–supported. It is in minimal, weakly, relation and strongly valid forms.*

Please note that the produced EASs need not be consistent. We will use the same semantics in our analysis as in the BAF–AFN case. However, now, due to the addition of η to the framework, the relation between BAF and EAS extensions becomes one–to–one only when we reach complete semantics (see e.g. Section 6.4):

Theorem 9.16. *Let $BF = (A, R, S)$ be a support acyclic BAF, $ES^{BF} = (A \cup \{\eta\}, R', N)$ the EAS associated with BF obtained through Translation 60 and $R'' = \{R^{sec}\}$ the collection of secondary attacks in BF. Then, a set $X \subseteq A$ is:*

- *+conflict–free w.r.t. $\emptyset$ in BF if it is conflict–free in ES^{BF}.*

- *inverse closed under S in BF if $X \cup \{\eta\}$ is self–supporting in ES^{BF}.*
- *+conflict–free w.r.t. R'' and inverse closed under S in BF if $X \cup \{\eta\}$ is strongly self–supporting in ES^{BF}.*
- *an i–admissible extension of BF w.r.t. (R'', R'') if $X \cup \{\eta\}$ is admissible in ES^{BF}.*
- *an i–preferred extension of BF w.r.t. (R'', R'') iff $X \cup \{\eta\}$ is preferred in ES^{BF}.*
- *a d–complete extension of BF w.r.t. (R'', R'') iff $X \cup \{\eta\}$ is complete in ES^{BF}.*
- *a d–grounded extension of BF w.r.t. R'' iff $X \cup \{\eta\}$ is grounded in ES^{BF}.*
- *a stable extension of BF w.r.t. R'' iff $X \cup \{\eta\}$ is stable in ES^{BF}.*

Additionally, a set $X' \subseteq A \cup \{\eta\}$ is:

- *conflict–free in ES^{BF} if $X' \cap A$ is +conflict–free w.r.t. $\emptyset$ in BF.*
- *self–supporting in ES^{BF} if $X' \cap A$ is inverse closed under S in BF.*
- *strongly self–supporting in ES^{BF} if $X' \cap A$ is +conflict–free w.r.t. R'' and inverse closed under S in BF.*
- *admissible in ES^{BF} if $X' \cap A$ is an i–admissible extension of BF w.r.t. (R'', R'').*

We can now put the translation into our classification system:

Redefinition of Translation 60: *Let $SAcy^{BAF}$ be the collection of all support acyclic BAFs on domain $\mathcal{U}$ and $ABin^{EAS} \cap SSig^{EAS} \cap SV^{EAS}$ the collection of all support singular, attack binary and strongly valid EASs on domain $\mathcal{U} \cup \{\eta\}$. The translation $Tr^{BAF}_{EAS} : SAcy^{BAF} \to (ABin^{EAS} \cap SSig^{EAS} \cap SV^{EAS})$ is defined as $Tr^{BAF}_{EAS}((A, R, S)) = (A \cup \{\eta\}, R', E)$ for a framework $(A, R, S) \in SAcy^{BAF}$, where $R' = \{(\{a\}, b) \mid (a, b) \in R\}$ and $E = \{(\{\eta\}, a) \mid a \in A, \nexists c \in A$ s.t. $cSb\} \cup \{(S^a, a) \mid S^a$ is the collection of all arguments $b \in A$ s.t. $bSa\}$.*

Redefinition of Theorem 9.16: *Let $\sigma^{BAF} \in \{$+conflict–free, inverse closed, +conflict–free and inverse closed, i–admissible, i–preferred, d–complete, d–grounded, stable$\}$ be a BAF semantics with identical parametrization consisting of secondary attacks and let $\sigma^{EAS} \in \{$conflict–free, self–supporting, strongly self–supporting, admissible, preferred, complete, grounded, stable$\}$ be a similar EAS semantics. Let SC^{Tr}_{σ} be the removal casting functions for σ defined as $SC^{X}_{\sigma}(Y) = Y \cap A$ for $X = (A, R, S) \in SAcy^{BAF}$ and $Y \in \sigma^{EAS}(Tr^{BAF}_{EAS}(X))$. The translation Tr^{BAF}_{EAS} is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is semantics bijective under the (d–) complete, (i–) preferred, (d–) grounded and stable semantics and the defined casting functions.*

The properties of Translation 60 are significantly different from Translation 59. Most notable is the difference in strengths and loss of modularity:

Analysis of Translation 60: Under the (+) conflict–free, (inverse closed) self–supporting, (+conflict–free and inverse closed) strongly self–supporting, (i–) admissible, (i–) preferred, (d–) complete, (d–) grounded and stable semantics with identical parametrization consisting of secondary attacks and removal casting functions, the translation Tr_{EAS}^{BAF} is:

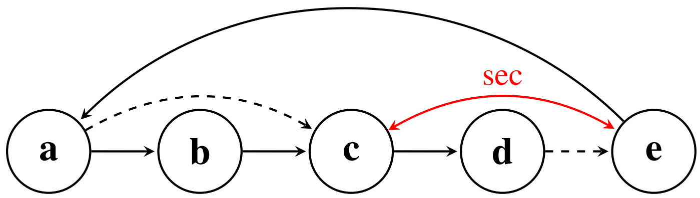
- source–subclass, target–subclass and injective
- weakly argument domain altering, argument introducing, induced support introducing
- generic and weakly semantics domain altering
- semi–structural

Translation Tr_{EAS}^{BAF} is not modular. It is faithful under the (d–) complete, (i–) preferred, (d–) grounded and stable semantics and the defined casting functions. Translation Tr_{EAS}^{BAF} is classified as basic under the listed semantics and casting functions.

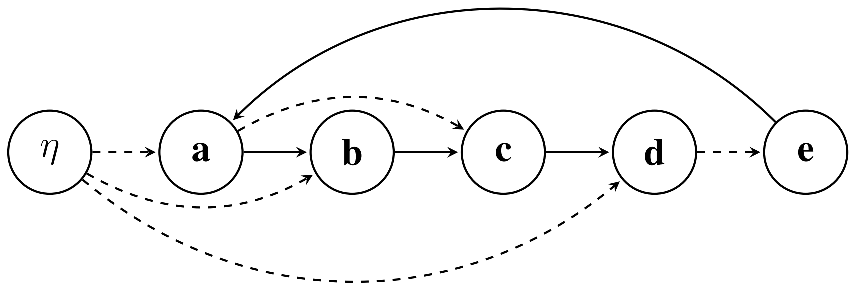
Explanation. Since we only consider support acyclic BAFs and all of the produced EASs are attack binary and support singular, our translation is both source and target–subclass. It is easy to show it is also injective. The domain is weakly altered due to the addition of evidence which we assume not to be the part of the domain of BAF arguments. Due to this auxiliary arguments and the support relations it brings it, our approach is argument and induced support introducing. The translation handles sufficiently many semantics to be classified as generic. The presence of η in the extensions also makes it weakly semantics domain altering. We choose to classify this method as semi–structural based on the reasons why evidence and the related supports need to be introduced.

Let us consider two simple BAFs $BF_1 = (\{a\}, \emptyset, \emptyset)$ and $BF_2 = (\{a, b\}, \emptyset, \{(b, a)\})$. Their corresponding EASs are $ES_1 = (\{a, \eta\}, \emptyset, \{(\{\eta\}, a)\})$ and $ES_2 = (\{a, b, \eta\}, \emptyset, \{(\{\eta\}, b), (\{b\}, a)\})$ respectively. We can observe that the EAS associated with $BF_1 \cup BF_2$ is simply ES_2. However, ES_2 differs from $ES_1 \cup ES_2$ by the $(\{\eta\}, a)$ support link. Thus, our translation is not modular. ■

Example 113. Let us consider a simple BAF $BF = (\{a, b, c, d, e\}$, $\{(a, b)$, (b, c), (c, d), $(e, a)\}$, $\{(a, c)$, $(d, e)\})$, previously analyzed in Example 108 and now depicted in Figure 92a. The secondary attacks in this framework are $R^{sec} = \{(e, c)$, $(c, e)\}$. The +conflict–free sets of BF w.r.t. $\emptyset$ are $\emptyset$, $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$, $\{a, c\}$, $\{a, d\}$, $\{b, d\}$, $\{b, e\}$, $\{c, e\}$, $\{d, e\}$ and $\{b, d, e\}$. With the exception of $\{c, e\}$, they are also +conflict–free w.r.t. secondary attacks. The sets $\emptyset$, $\{a\}$, $\{b\}$, $\{d\}$, $\{a, c\}$, $\{d, e\}$ and their combinations are inverse closed. The d–admissible extension of BF are $\emptyset$, $\{e\}$, $\{a, c\}$, $\{b, e\}$, $\{d, e\}$, $\{b, d, e\}$. From this, $\emptyset$, $\{a, c\}$, $\{d, e\}$ and $\{b, d, e\}$ are i–admissible, with $\{a, c\}$ and $\{b, d, e\}$ being i–preferred. We can observe that $\emptyset$, $\{a, c\}$ and $\{b, d, e\}$ do not defend any argument w.r.t. the direct and secondary attacks and are therefore our d–complete extensions. Concerning the remaining d–admissible sets, $\{e\}$ defends b and d $\{b, e\}$ defends d and $\{d, e\}$ defends



(a) Sample BAF with secondary attacks



(b) Associated EAS

Figure 92: Sample BAF and its associated EAS

b. The d–grounded extension of BF is $\emptyset$; we can observe that every argument is, directly or secondary, attacked in BF. Finally, both $\{a, c\}$ and $\{b, d, e\}$ are stable in BF.

The EAS associated with our BF is $ES^{BF} = (\{\eta, a, b, c, d, e\}, \{(\{a\}, b), (\{b\}, c), (\{c\}, d), (\{e\}, a)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, d), (\{a\}, c), (\{d\}, e)\})$, as visible in Figure 92b. Every argument possesses an evidential sequence; arguments a, b and d have trivial sequences (η, a), (η, b) and (η, d), while e and c possess (η, a, c) and (η, d, e). Therefore, the sets $\emptyset$, $\{\eta\}$, $\{\eta, a\}$, $\{\eta, b\}$, $\{\eta, d\}$, $\{\eta, a, c\}$, $\{\eta, d, e\}$ and their combinations are self–supporting. Thus, if we remove η, we can observe that the inverse closed sets of BF are retrieved. The conflict–free sets of ES are $\emptyset$, $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$, $\{a, c\}$, $\{a, d\}$, $\{b, d\}$, $\{b, e\}$, $\{c, e\}$, $\{d, e\}$, $\{b, d, e\}$, $\{\eta\}$, $\{\eta, a\}$, $\{\eta, b\}$, $\{\eta, c\}$, $\{\eta, d\}$, $\{\eta, e\}$, $\{\eta, a, c\}$, $\{\eta, a, d\}$, $\{\eta, b, d\}$, $\{\eta, b, e\}$, $\{\eta, c, e\}$, $\{\eta, d, e\}$ and $\{\eta, b, d, e\}$. We can see they correspond to the +conflict–free sets of BF w.r.t. $\emptyset$, not w.r.t. secondary attacks. From the strongly self–supporting sets $\emptyset$, $\{\eta\}$, $\{\eta, a\}$, $\{\eta, b\}$, $\{\eta, d\}$, $\{\eta, a, c\}$, $\{\eta, a, d\}$, $\{\eta, b, d\}$, $\{\eta, d, e\}$ and $\{\eta, b, d, e\}$, only $\emptyset$, $\{\eta\}$, $\{\eta, a, c\}$, $\{\eta, d, e\}$ and $\{\eta, b, d, e\}$ are admissible. This is in correspondence with the i–admissible sets of BF; we can observe that $\emptyset$ can be produced both from $\emptyset$ and $\{\eta\}$. The complete extension of ES^{BF} are $\{\eta\}$, $\{\eta, a, c\}$ and $\{\eta, b, d, e\}$. Thus, again we retrieve all and only d–complete extensions of BF. However, this time the relation is one–to–one. The preferred and stable extensions $\{\eta, a, c\}$ and $\{\eta, b, d, e\}$ are in agreement with the i–preferred and stable extensions of BF. Similarly, $\emptyset$ is correctly retrieved from $\{\eta\}$ as the grounded extension.

9.4 BAF as ADF

The nature of support in ADFs is not as cleanly defined as in other frameworks. Thus, in this section we will try to highlight some similarities and differences between the BAF and ADF approach towards positive relations.

Let us start with the abstract support. Although it is somewhat an atypical type of relation and its criticism was in a great deal a reason for developing other types of support and their corresponding frameworks, we believe it still has a lot of potential. In particular, one can consider abstract support as a source of a certain weak preference, and permit the preferences of arguments in an extension to define the ordering of the produced answers. One can think of a situation with two mutually attacking arguments a and b, depicting a situation such as “Let us go dine out this evening” and “Let us stay in for the evening”. We obtain three admissible extensions $\{a\}, \{b\}$ and $\emptyset$. Including an argument c that would express that we are in favor of a (e.g. “My cousin started working at a restaurant, we can get a discount if we go there.”) would make us rather choose $\{a\}$ (or $\{c, a\}$) over $\{b\}$. Nevertheless, a can still be accepted on its own, i.e. the fact that one does not have helpful cousins working at restaurants does not mean one cannot dine out at all. Up to some extent, defense attack behaves similarly as the preference through support. However, in our case, the existence of c does not make $\{b\}$ unacceptable, just $\{a, c\}$ more preferred to it. Thus, we should not introduce a supported attack from c to b, unless it is accompanied by a mediated one from b to c. At the same time being in favor of some argument does not imply that we will always accept it. Including a new attacker of a, say d (e.g. “My nice clothes are in the laundry”), would make a disappear from any admissible extension. On the other hand, the $\{c, b, d\}$ set can still be accepted as a reasonable extension (e.g. “Going out would be nice because we could get a discount, but I have nothing to wear, so lets stay home.”). This interpretation is very similar to defense attacks in EAFs. In this case the e.g. mediated attack from d to c is too powerful. Similar examples can be drawn for secondary attack, where cutting off some support weakens an argument rather than excludes it from extensions. This is a type of reasoning that is in many ways closer to value and preference based argumentation or ranking based semantics [1], hence the frameworks and approaches we are discussing now are not exactly adequate for it. However, the point was to show that a positive relation between arguments does not necessarily imply a strong dependency between them, i.e. that acceptance of one argument would lead to accepting another, and that the abstract support can still be a base for further research.

Nevertheless, the acceptance conditions in ADFs speak in terms of what should be present or not in order to be able to assume an argument, which is conceptually quite the opposite from the idea of abstract support. Moreover, the semantics are parametrized w.r.t. support cycles, not w.r.t. different types of conflicts. Consequently, without going somewhat against the design of ADFs and introducing a wide number of additional notions, the abstract support cannot be grasped in the same way it is in BAFs. Therefore, in this section we will not provide any translations. However, we will explain how the indirect attacks look like in ADFs and why not all of them appear in this setting.

In a certain sense, out of abstract, necessary and deductive supports, the positive relations in ADFs are perhaps the closest to the necessary ones. However, unlike in AFNs, one can permit support cycles to appear in ADF semantics, which might lift some of the limitations we had to consider in Section 9.2. The way ADFs grasp secondary attack can be best observed when we look at evaluations, where the pd–set or sequence contains "supporters", and the blocking set stores their "attackers". However, since ADFs can express more than just binary relations, its behavior in a more complex setting changes. Let us assume a simple ADF $(\{a, b, c\}, \{C_a = b, C_b = \neg c, C_c = \top\})$, where a is (necessarily and "binary") supported by b and b is attacked by c. Accepting c outs b, and by a chain reaction also a. Therefore a and c will never appear together in a conflict–free extension. However, please note that this state of affairs is not exactly permanent. We can introduce another argument d, not connected to c in any way, that can change this situation. For example, we can look at the framework $(\{a, b, c, d\}, \{C_a = b \vee d, C_b = \neg c, C_c = \top, C_d = \top\})$. Although b is out of the question, a can still be derived through d and even though $\{a, c\}$ is not admissible, $\{a, c, d\}$ is. Another option is to use the so–called "overpowering support" in ADFs, which was more explained in Section 8.6. We can consider another modification of our base framework $(\{a, b, c, d\}, \{C_a = b, C_b = \neg c \vee d, C_c = \top, C_d = \top\})$. The presence of d can override the conflict from c, and in this case $\{a, b, c, d\}$ is admissible. The point is, that one indirect conflict in BAFs is enough to "kill" an argument, and in ADFs it is not, even if we permit every argument to have only a single supporter.

The *supported attack* is not used in ADFs due to its counterintuitive behavior in a variety of examples. Although it is useful in modeling deductive support, in our case supporting an attacker does not have the same power as attacking a supporter. It is a threat, yes, and taking it into account can be motivated. However, just like assuming an indirect defender (i.e. defender of a defender) is not enough for admissibility, for now we have decided that unless the threat can be executed by accepting the actual attacker, it is to be treated as empty and disregarded. It is particularly important due to the fact that if the supported argument is e.g. a self–attacker, it can never appear in an extension, even if its supporter does. Moreover, we believe that supported attack has certain hidden assumptions and can lead to counterintuitive modeling in certain problems.

First of all, on many occasions, arguments leading to contradictory conclusions can share certain requirements. This is thus what we believe to be the first assumption of secondary attack, that conflicting arguments do not have any supporters in common. A fever can be a symptom of various things and hint at different diagnoses; creating a supported attack from the symptom to any diagnosis just because the diagnoses themselves can be in conflict would be highly undesirable. In addition, in order to discuss a nature of something, we first have to assume it exists. Take for example *b: the chair in the room is blue* vs *r: the chair in the room is red*, assuming that *c: there is a chair in the room*. Adding a conflict from c to any of b, r does not seem rational. Some examples can also be found in nonmonotonic logic, where adding new information can lead to a conclusion contradicting the old one, but it does not necessarily mean that what was used to derive it in the first place is now in conflict with it. Overall, the "enemy of my friend is my enemy" is not always an adequate approach. We believe it is another question for further research on relations between arguments, perhaps there might be some conditions that would ensure this “negative transitivity”.

Let us look at a different scenario and assume we are thinking about a holiday break. A travel agency gave us a few brochures on hotels A and B and we are reading through them right now. We will of course choose only one, so choosing A is in conflict with choosing B and vice versa. Now, among other requirements, we are sure we want our hotel to be quiet. We are reading the brochure on A and we find out that it satisfies our needs and thus we add a support from our condition to choosing A. This creates a supported attack from the requirement to choosing B. Should we look and the framework with the auxiliary attacks included, we would notice that our modeling implies that B is not a quiet hotel. This is of course not something we know, as we are yet to go through the brochure on the hotel. Consequently, we can see the supported attack as preemptive and hinting that we have already read the brochure and verified that the hotel is indeed not quiet. Although in defeasible reasoning we are supposed to be prepared to deal with incomplete information, there is a difference between working with what we know and being aware of incomplete knowledge, and filling in some missing information without prior verification and proceeding as if the data was complete. This brings us to the other assumption of secondary attack, namely that we have all the knowledge we need at hand; otherwise, the modeling is simply inaccurate. However, please note that given the current information, going with the A hotel would be a better option – we know it is quiet and this is more than we know about B. Thus for now, it is more preferred. However, this type of attitude can be modeled with a variety of different approaches, such as preferences or evidential support, without the drawback of supported attack. Finally, we can argue that more than one argument might be required to carry out a supported attack. However, we believe this issue to be a structural one and dependent on how the actual support relation is interpreted, i.e. whether all or any supporting arguments need be present for an argument to be accepted. Thus, this is not an issue really relevant for the current analysis.

Although necessary support can be modeled within ADFs (see Section 10.6), the deductive support is not directly handled. The mediated attack follows the idea that attack is stronger than support, which in some situations is not the best assumption (see Example 114) and it is something we have tried to avoid in ADFs up to a certain extent. However, the only way to really address this issue would be to introduce strengths in the framework and define the success of an attack based on them, the way it is done e.g. in preference–based frameworks or structured argumentation [2, 66, 67]. Consequently, this is a problem on a different scale, and the purpose of this discussion is only to warn the reader that even though support frameworks are getting closer and closer to structured argumentation frameworks, modeling a problem directly still has certain traps.

Example 114. Let us consider the following case. There was a robbery and John is a suspect. He claims he is innocent, as he was at a football match at that time and couldn't have done it (argument **i**). However, there is a witness claiming John is guilty of the crime (argument **w**). Later, it turns out that the football match was filmed and the recordings show without a doubt that John was indeed there (argument **r**). We can create the following framework $BF = (\{i, w, r\}, \{(w, i)\}, \{(r, w)\})$ describing this situation. In our case, if we accept the recording, then we need to accept the fact that John was watching the match and that he is indeed innocent. Thus, the support relation between r and i can be viewed as deductive. Since w contradicts i, we need to create a mediated attack from w to r, which is counterintuitive. Unless new facts come into play, the recording is more trustworthy than a witness testimony in this case. An attack from r to w would be much more natural, and even if it were added at this point, the symmetry of the conflict would still make w acceptable, which should not be the case.

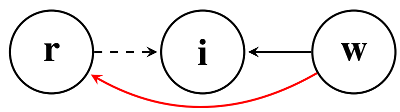


Figure 93: Sample BAF

9.5 BAF as Other Frameworks

In this section we have analyzed BAFs in context of AFs, AFNs, EASs and up to some extent, ADFs. We have limited ourselves to studying particular subsets of indirect attacks, related to deductive and necessary support, and enforced identical parametrization of the semantics. Moreover, as we have observed, not every type of semantics is defined for BAFs. For these reasons, we have not taken into account other frameworks in our analysis. In particular, we have not considered any approaches from BAFs to SETAFs due to the fact that the relations we have considered so far were binary. Consequently, the presented results should be more treated as a follow–up on the analysis carried out in [30, 31] and we hope further analysis can be carried out once the BAF semantics become more clear.

9.6 Summary

The summary of our results can be seen in Table 12. For the parametrization of a given semantics we refer the reader to the relevant section. We can observe that out of all of the proposed translations, the attack propagation approach to AFs and the transformation for necessary support to AFNs seem to be the most interesting ones in the context of the amount of handled semantics. Although the first one is definitely more general then the other due to more freedom in parametrization, the BAF–AFN translation is the only modular and structural approach we have found.

Table 12: Translations from BAFs to other frameworks

Properties		AF				AFN	EAS
Translation		53	54	55	57	59	60
Strength	+cf	exact				exact	strong
	+cf, cl				strong		
	+cf, inv cl					exact	strong
	d–adm	exact					
	d–comp	exact			strong	exact	faithful
	d–pref	exact					
	d–grd	exact			strong	exact	faithful
	stb	exact			strong	exact	faithful
	i–adm		exact			exact	strong
	i–pref		exact			exact	faithful
	c–adm			exact	strong		
	c–pref			exact	strong		
Functional		full	full	full	full	source–subclass	source–subclass
		surjective	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass
		overlapping	overlapping	overlapping	overlapping	injective	injective
Syntactical		argument domain preserving	argument domain altering	argument domain altering	argument domain altering	argument domain preserving	weakly argument domain altering
		attack introducing	argument introducing	argument introducing	argument introducing	structure preserving	argument introducing
		support removing	attack introducing	attack introducing	attack introducing(attack removing)		induced support introducing
					support introducing(support removing)		
Semantical		generic	specialized	specialized	generic	generic	generic
		semantics domain preserving	semantics domain preserving	semantics domain preserving	semantics domain altering	semantics domain preserving	weakly semantics domain altering
Computational		semi–structural	semi–structural	semi–structural	semi–structural	structural	structural
						modular	

10 Translating AFNs

In this section we will show various ways we can translate AFNs to AFs, SETAFs, BAFs, EASs and ADFs. Moreover, even though we do not create an AFN–EAF translation, we provide a comparison between the necessary supports and defense attacks. The translation to the Dung's framework will follow the standard coalition pattern. However, as we will show, for some semantics it appears to be possible to create an exact approach, though at this point we are not sure of the precise nature of such a translation. For the SETAF transformations, we will present attack propagation and defender approaches. As we might recall, an AFN argument can possess a number of powerful sequences and supporting sets. Hence, for the defender approach, we use group attack to simulate group support. For the other method, we can observe that in order to reject an argument, we need a way to attack all of its powerful sequences. Since it is not necessarily the case that they share a single attacker, we use group attack to gather the required attackers.

The translation from AFNs to BAFs that we will present is a limited, source–subclass approach which is meant as a comment on the results from [30] and highlights the fact that the necessary approach the way it is seen in BAFs is distinct from the way it is defined in AFNs. The EAS transformation is, on the other hand, capable of handling any type of an AFN. The method is quite straightforward and we can observe that the most significant difference between the necessary and evidential supports concerns more their structure rather than semantics. Finally, bearing in mind the consistency issues raised in Section 2.3.9, we present in total three approaches for shifting AFNs to ADFs, one in which the framework is assumed to be strongly consistent and two in which it is made consistent by using the bypass and self–attacker normal forms.

10.1 AFN as AF

The AFN–AF coalition–style translation has been introduced in [69]. The AF arguments now represent sets of AFN arguments that meet support requirements, i.e. are coherent. Consequently, we can observe that arguments not possessing powerful sequences will not be represented in the target framework. Nevertheless, with the exception of conflict–freeness, the semantics are preserved by the translation.

Translation 61. *Let $FN = (A, R, N)$ be an AFN. Its corresponding AF $F^{FN} = (A', R')$ is built as follows:*

- *for every argument $a \in A$, add $C \subseteq A$ to A', where C is a minimal coherent set containing a, and*
- *for any $C_1, C_2 \in A'$, $C_1 R' C_2$ iff $\exists x \in C_1, y \in C_2$ s.t. xRy.*

The existing results (Theorem 10.1) are somewhat limited; only the more advanced semantics are considered and it is not analyzed whether the relation between the source and target extensions is bijective. Therefore, we choose to complement this analysis with the study of conflict–free, strongly coherent and admissible extensions. Please note that due to the nature of the AF arguments, we do not need to focus on the coherent semantics itself. We can observe that the relation between the admissible extensions of both framework is not one–to–one (see Example 115). However, it becomes such when we consider the complete semantics.

Theorem 10.1. *Let $FN = (A, R, N)$ be an AFN and $F^{FN} = (A', R')$ its corresponding AF obtained through Translation 61. A set $E \subseteq A$ is a σ–extension, where $\sigma \in \{grounded, complete, preferred, stable\}$ of FN iff F admits a set $Y = \{C_1, ..., C_n\} \subseteq A'$ s.t. $E = \bigcup_{i=1}^{n} C_i$ is a σ–extension.*

Theorem 10.2. *Let $FN = (A, R, N)$ be an AFN and $F^{FN} = (A', R')$ its corresponding AF built from Translation 61. If $E' \subseteq A'$ is conflict–free in F^{FN}, then $\bigcup E'$ is conflict–free in FN, but not vice versa. A set $E \subseteq A$ is a strongly coherent extension of FN iff F^{FN} admits a set $E' = \{C_1, ..., C_n\} \subseteq A'$ s.t. $E = \bigcup_{i=1}^{n} C_i$ as a conflict–free extension. A set $E \subseteq A$ is an admissible extension of FN iff F^{FN} admits a set $E' = \{C_1, ..., C_n\} \subseteq A'$ s.t. $E = \bigcup_{i=1}^{n} C_i$ as an admissible extension. For every complete extension of FN there exists exactly one corresponding complete extension of F^{FN}.*

We can also notice that in order to retrieve the conflict–free extensions of a given AFN, we only need to take into account the conflict subframework:

Translation 62. *Let $FN = (A, R, N)$ be an AFN. Its conflict–corresponding AF is $F_{cf}^{FN} = (A, R)$.*

Theorem 10.3. *Let $FN = (A, R, N)$ be an AFN and $F^{FN} = (A, R)$ its conflict–corresponding AF obtained through Translation 62. A set $E \subseteq A$ is a conflict–free extension of FN iff it is a conflict–free extension of the AF F^{FN}.*

The fact that $E \subseteq A$ is a conflict–free extension of FN iff it is a conflict–free extension of (A, R) follows easily from the definition of this semantics in AFNs. We can now summarize the results:

Redefinition of Translation 61: *Let Fr^{AFN} be the collection of all AFNs on the domain $\mathcal{U}$ and Fr^{AF} the collection of all AFs on argument domain $2^{\mathcal{U}}$. The translation Tr_{AF}^{AFN} : $Fr^{AFN} \rightarrow Fr^{AF}$ $Tr_{AF}^{AFN}((A, R, N)) = (A', R')$, where $A' = \{C \mid C \subseteq A$ is a minimal coherent set for an argument $a \in A\}$ and $R' = \{(C_1, C_2) \mid C_1, C_2 \in A'$, $\exists x \in C_1, y \in C_2$ s.t. $xRy\}$ for a framework $(A, R, N) \in Fr^{AFN}$.*

Redefinition of Theorems 10.1 and 10.2: *Let $\sigma^{AFN} \in \{$strongly coherent, admissible, complete, preferred, grounded, stable$\}$ be an AFN semantics and $\sigma^{AF} \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ a similar AF semantics. Let SC_{σ}^{Tr} be the union casting functions for σ. The translation Tr_{AF}^{AFN} is strong under $(\sigma, SC_{\sigma}^{Tr})$. It is semantics bijective under complete, preferred, grounded and stable semantics and the defined casting functions.*

Analysis of Translation 61: Under the (conflict–free) strongly coherent, admissible, complete, preferred, grounded and stable semantics and union casting functions, the translation Tr_{AF}^{AFN} is:

- full, target–subclass and overlapping
- argument domain altering, argument removing, argument introducing, (possibly induced) attack removing, support relation removing, possibly attack and support relation introducing

- generic and semantics domain altering
- semantical

The translation is not modular. The translation Tr^{AFN}_{AF} is classified as coalition style under the listed semantics and casting functions.

Explanation. Any AFN can undergo the translation; thus, our approach is full. It is however target–subclass only due to the domain change. For example, a framework $(\{\{a\}, \{a, b\}, \{c\}\}, \{(\{c\}, \{a\})\})$ cannot be produced by our translation; if $\{c\}$ attacks $\{a\}$, then by construction it also has to attack $\{a, b\}$. However, if we ignored the domain completely and focused on e.g. attack paths, any type of AF would be produced. This follows simply from the fact that chaining the AF–AFN and AFN–AF translations, the difference between the initial and resulting framework would only affect the argument domain, but not structure as such. For example, an AF $(\{a, b, c\}, \{(a, b), (b, c)\})$ would become $(\{\{a\}, \{b\}, \{c\}\}, \{(\{a\}, \{b\}), (\{b\}, \{c\})\})$. Finally, our translation is also overlapping due to the removal of arguments that do not possess coherent sets – a framework such as $(\{\{a\}\}, \emptyset)$ can be obtained from a number of AFNs, such as $(\{a\}, \emptyset, \emptyset)$ and $(\{a, b\}, \emptyset, \{(\{b\}, b)\})$.

Clearly, both argument and semantics domain are affected by our approach. Due to the amount of semantics handled in a strong manner, we choose to classify it as generic. The translation is also argument removing, as visible in the previous explanation. However, it can also be seen as argument introducing, as a given source argument can be represented by a number of target ones – it merely depends on the number of minimal coherent sets it possesses and on the elements it supports. Removing arguments leads to removing the relations between. However, this removal is definitely more than just induced in the case of support, as even a valid supporter of an argument might not necessarily appear in a coherent set with it. Consider a framework $(\{a, b, c\}, \emptyset, \{(\{a, c\}, b), (\{a, b\}, c)\})$. The minimal coherent sets for a, b and c are $\{a\}$, $\{a, b\}$ and $\{a, c\}$ respectively. Therefore, the information on whether b and c are supporting each other is lost, even though both (a, b, c) and (a, c, b) are powerful sequences for them. Finally, for reasons similar as in coalition BAF–AF Translation 57, we may consider the translation attack removing (not just induced removing) and possibly relation introducing. The coalition arguments are not precise enough and the fact that now we can assume that there are no support cycles in them does not change this fact. We can consider the frameworks $(\{a, b, c\}, \{(b, a), (c, a)\}, \{(\{b\}, c)\})$ and $(\{a, b, c\}, \{(b, a)\}, \{(\{b\}, c)\})$; they will both be translated to the same AF $(\{\{a\}, \{b\}, \{b, c\}\}, \{(\{b, c\}, \{a\}), (\{b\}, \{a\})\})$ despite the fact, even though both of them are in minimal and strongly valid forms. Thus, even though the (b, a) attack will be retrieved, the (c, a) one might be lost or added depending on how we proceed. Similar situation in case of support occurs e.g. in frameworks $(\{a, b, c\}, \emptyset, \{(\{a\}, b), (\{b\}, c)\})$ and $(\{a, b, c\}, \emptyset, \{(\{a\}, b), (\{b\}, c), (\{a\}, c)\})$.

The translation is clearly semantical as it depends on the computation of minimal coherent sets. Unfortunately, it is not modular, and for reasons similar as in the case of weakly valid normal form translation (Translation 7). The frameworks $FN_1 = (\{a, b\}, \emptyset, \{(\{a\}, b)\})$ and $FN_2 = (\{a, b\}, \emptyset, \{(\{b\}, b)\})$ will be transformed into $F_1 = (\{\{a\}, \{a, b\}\}, \emptyset)$ and $F_2 = (\{a\}, \emptyset)$ respectively. However, the AF corresponding to $FN_1 \cup FN_2$ is the same as F_2 and thus different from $F_1 \cup F_2$. Consequently, our approach is not modular. ■

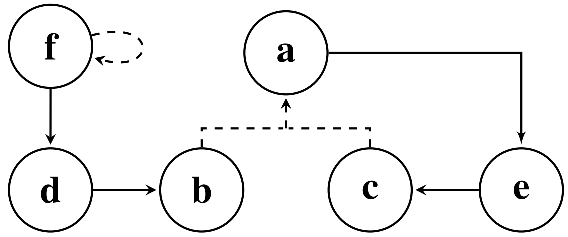
Redefinition of Translation 62: *Let Fr^{AFN} be the collection of all AFNs and Fr^{AF} the collection of all AFs, both on argument domain $\mathcal{U}$. The translation $cf\text{-}Tr_{AF}^{AFN} : Fr^{AFN} \rightarrow Fr^{AF}$ is defined as $cf\text{-}Tr_{AF}^{AFN}((A, R, N)) = (A, R)$ for a framework $(A, R, N) \in Fr^{AFN}$.*
Redefinition of Theorems 10.3: *The translation $cf\text{-}Tr_{AF}^{AFN}$ is strong and semantics bijective under the conflict–free semantics and identity casting function.*
Analysis of Translation 62: Under the conflict–free and identity casting functions, the translation $cf\text{-}Tr_{AF}^{AFN}$ is:

- full, surjective and overlapping
- argument domain preserving, support relation removing
- specialized, semantics domain preserving and exact
- structural and modular

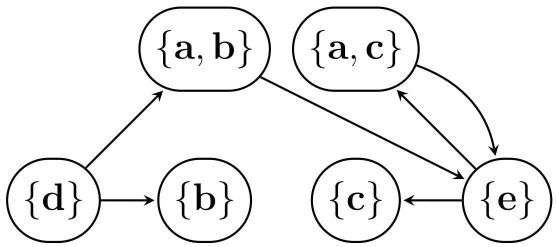
The translation $cf\text{-}Tr_{AF}^{AFN}$ is classified as basic style under the listed semantics and casting functions.

Explanation. Due to the fact that we can work with any AFN, our translation is full. It is also surjective – for any type of an AF (A, R) we can find an AFN producing it, for example $(A, R, \emptyset)$ (see Translation 21). In this translation, we are completely discarding the support relation and not altering the attack relation in any way in order to account for it. Therefore, two frameworks with same conflicts and arguments, but different supports, will be translated into the same AF and thus the approach is overlapping. Moreover, it is relation removing, even though the set of arguments remains unaffected. The nature of the arguments in the source and target framework remains unchanged; also the used casting function is an identity. Hence, we preserve both argument and semantics domains. Furthermore, the translation is specialized and exact for conflict–free semantics only (see Theorem 10.3). Finally, due to its simplicity, it is both structural and modular. We thus choose to classify the translation as basic. ■

Due to the presence of group support and special handling of support cycles, AFs are not the best target for the direct attack propagation and defender translations that we analyzed in Section 9. We will use SETAFs for that in the next section and propose to chain our results with the SETAF–AF Translation 25.



(a) Sample AFN



(b) Associated coalition AF

Figure 94: Sample AFN and its associated coalition AF

Example 115. Let us come back to the AFN $FN = (\{a, b, c, d, e, f\}, \{(a, e), (d, b), (e, c), (f, d)\}, \{(\{b, c\}, a), (\{f\}, f)\})$ from Example 16, now depicted in Figure 94a. The minimal coherent sets for a are $\{a, b\}$ and $\{a, c\}$, while for the arguments b, c, d and e it is simply the sets containing only them. We can observe that the argument f does not possess a coherent set at all. Our coalition AF is therefore $F^FN = (\{\{a, b\}, \{a, c\}, \{b\}, \{c\}, \{d\}, \{e\}\}, \{(\{d\}, \{b\}), (\{e\}, \{c\}), (\{f\}, \{d\}), (\{d\}, \{a, b\}), (\{e\}, \{b, c\}), \{(\{a, b\}, \{e\}), \{(\{a, c\}, \{e\})\}\})$, as seen in Figure 94b.

The conflict–free sets of our AF are $\emptyset$, $\{\{b\}\}$, $\{\{c\}\}$, $\{\{d\}\}$, $\{\{e\}\}$, $\{\{a, b\}\}$, $\{\{a, c\}\}$, $\{\{b\}, \{c\}\}$, $\{\{b\}, \{e\}\}$, $\{\{b\}, \{a, b\}\}$, $\{\{b\}, \{a, c\}\}$, $\{\{c\}, \{d\}\}$, $\{\{c\}, \{a, b\}\}$, $\{\{c\}, \{a, c\}\}$, $\{\{d\}, \{e\}\}$, $\{\{d\}, \{a, c\}\}$, $\{\{a, b\}, \{a, c\}\}$, $\{\{b\}, \{c\}, \{a, b\}\}$, $\{\{b\}, \{c\}, \{a, c\}\}$, $\{\{b\}, \{a, b\}, \{a, c\}\}$, $\{\{c\}, \{d\}, \{a, c\}\}$, $\{\{c\}, \{a, b\}, \{a, c\}\}$ and finally, $\{\{b\}, \{c\}, \{a, b\}, \{a, c\}\}$. They correspond to sets $\emptyset, \{b\}, \{c\}, \{d\}, \{e\}$, $\{a, b\}$, $\{a, c\}$, $\{b, c\}$, $\{b, e\}$, $\{c, d\}$, $\{d, e\}$, $\{a, b, c\}$ and $\{a, c, d\}$, which are exactly the strongly coherent sets of FN. We can observe that in some cases, a given AFN set can be produced from multiple AF extensions. We can observe that the set $\{f\}$ is conflict–free in FN, but it cannot be produced from any conflict–free extension of F^{FN}. The admissible extensions of F^{FN} are $\emptyset$, $\{\{d\}\}$, $\{\{d\}, \{e\}\}$, $\{\{a, c\}\}$, $\{\{c\}, \{a, c\}\}$, $\{\{d\}, \{a, c\}\}$ and $\{\{c\}, \{d\}, \{a, c\}\}$. They correspond to the sets $\emptyset$, $\{d\}$, $\{d, e\}$, $\{a, c\}$ and $\{a, c, d\}$, which again is in agreement with the admissible extensions of FN. Out of the possible admissible sets of our AF, $\{\{d\}\}$, $\{\{d\}, \{e\}\}$ and $\{\{c\}, \{d\}, \{a, c\}\}$ are complete. We thus can retrieve all and only complete extension of FN; we can also observe that this time, the relation between the source and target answers is one–to–one. The first set is grounded and the other two are preferred, which the desired result. Finally, both $\{\{d\}, \{e\}\}$ and $\{\{c\}, \{d\}, \{a, c\}\}$ are stable in F^{FN}. Since $\{d, e\}$ and $\{a, c, d\}$ are stable in FN, we can observe that all and only extensions of FN are retrieved.

10.1.1 Improvements

To this point, we are not aware of any full and exact translations from AFNs to AFs. Moreover, in [69] the authors claim that AFNs can express more than AFs and that it is unlikely that an exact translation exists. However, the sample frameworks they have provided produce sets of extensions consisting only of a single set containing a single argument (with the exception of admissible semantics, where $\emptyset$ was included). Such collections are trivially realizable in AFs under the complete, preferred and stable semantics. Consequently, the initial claim is in fact not supported. Therefore, we have decided to establish whether it is possible to create an approach stronger than the coalition translation on our own. Although the precise signatures of the AFN semantics are not known to us, it turns out we can show that they do meet the sufficient requirements of some of the AF semantics signatures.

Let us first start with the admissible semantics; it can be shown that if there is no conflict between the member of two admissible extensions, then their union is an admissible extension as well. Please note that we do not claim that this result also holds for e.g. BAFs. Their semantics are distinctively different from AFNs and as such pose a different challenge.

Theorem 10.4. *Let $FN = (A, R, N)$ be an AFN and $E, E' \subseteq A$ two admissible extensions of FN. If for every $a \in E$, $b \in E'$ it is not the case that $(a, b) \in R$ and $(b, a) \in R$, then $E \cup E'$ is also admissible.*

Although a similar property is true in various frameworks, AFNs use binary attack just like AFs do, and thus the redefinition of the adm–closed property (see Definitions 2.175 and 2.176) can be proved:

Theorem 10.5. *Let $FN = (A, R, N)$ be an AFN. For any two admissible extensions E_1 and E_2 of FN, if for every argument $a \in E_1, b \in E_2$ there exists an admissible extension E_3 of FN s.t. $a, b \in E_3$, then $E_1 \cup E_2$ is an admissible extension of FN.*

We can finally observe that $\emptyset$ is also a trivial AFN admissible extension. Therefore, although the precise conditions for the AFN admissible signature are not known, we know that the requirements of the AF admissible signature are satisfied. Consequently, we can conclude that a full and exact translation from AFNs to AFs under the admissible semantics is possible. Moreover, this result can be extended to the preferred semantics:

Theorem 10.6. *Let Fr^{AFN} be the collection of all AFNs on the domain $\mathcal{U}^{AFN}$ and Fr^{AF} the collection of all AFs on the domain $\mathcal{U}^{AF}$. There exists a full translation from Fr^{AFN} to Fr^{AF} that is exact under the admissible (preferred) semantics and identity casting functions.*

Oddly enough, the AFN stable semantics do not fit the AF signature – the produced sets do not need to be tight, as seen in the following example. This means that even there is no exact (and full) translation from AFNs to AFs for the stable semantics.

Example 116. Let $FN = (\{a, b, c, d, e\}, \{(a, c), (c, a), (b, d), (d, b), (c, b), (b, c)\}, \{(\{c, d\}, e)\})$ be the AFN depicted in Figure 95. Our stable extensions are $\{a, b\}$, $\{b, c, e\}$ and $\{a, d, e\}$. However, the collection is not tight (see Definition 2.176) – for example, the set $\{a, b, e\}$ is not present. It is conflict–free, but it is not coherent.

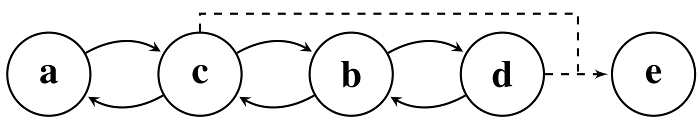


Figure 95: Sample AFN

Theorem 10.7. *Let Fr^{AFN} be the collection of all AFNs on the domain $\mathcal{U}^{AFN}$ and Fr^{AF} the collection of all AFs on the domain $\mathcal{U}^{AF}$. There does not exist a full translation from Fr^{AFN} to Fr^{AF} that is exact under the stable semantics and identity casting functions.*

Due to the fact that sufficient conditions for the AF complete signature are not known yet, we cannot say whether an appropriate exact translation is possible. Thus, along with finding an appropriate translation for the admissible and preferred semantics, this task is left for future work.

Finally, we would like to notice that the AFN–AF coalition translation is the first truly semantical approach we consider in this work. Thus, it is natural to ask under which conditions we can take it at least back to the semi–structural level. The answer is: we need to work with strongly valid frameworks. Thanks to Theorem 4.32, if this normal form/subclass is assumed, the coherent extensions can be equivalently expressed as sets in which all arguments require support. Thus, no further validity analysis is required, and we may choose to reclassify the approach as semi–structural. For now, we are not aware of any other options that would allow us to simplify the translation.

10.2 AFN as SETAF

In order to translate AFNs into SETAFs, we can use the AFN–AF coalition translation and then the AF–SETAF approach. However, there are also two alternative methods which in our opinion provide some insight into the stronger form of supports, such as necessary or evidential. Moreover, they also show why the attack propagation translation (Translation 53) for BAFs is not directly applicable , and how the defender translation (Translation 58) can be adapted to work in the AFN setting.

10.2.1 Attack Propagation

In BAFs for necessary support, every argument was considered valid, and every attacker of any supporter of an argument became an indirect attacker of this argument. In AFNs, when the semantics treat support cycles in a different manner and we are faced with group support, the situation looks differently. When we look at the definition of defense in AFNs, we can see that we consider an attack to be defended from if we can attack every coherent set containing the attacker. Thus, propagating the attacks carried out at coherent sets to the actual arguments for which they exist would address the issue of cycles. Moreover, due to the fact that a given argument can be derived in more than one way, the indirect attacks are carried out by sets of arguments rather than single elements. Let us look at an example:

Example 117. Let us consider the AFN $FN_1 = (\{a, b, c, d, e, f\}, \{(d, b), (e, c), (a, f)\}, \{(\{b, c\}, a)\})$ depicted in Figure 96a. d attacks b, however, this is insufficient to really cut off the support of a. For example, the set $\{a, c, d\}$ is still strongly coherent. Similar case is when we consider just e. It is only the presence of both d and e that makes it impossible for a to appear in a strongly coherent set. If we consider the corresponding SETAF $SF_1 = (\{a, b, c, d, e, f\}, \{(\{a\}, f), (\{d\}, b), (\{e\}, c), (\{d, e\}, a)\})$, we can see that it returns the same admissible extensions.

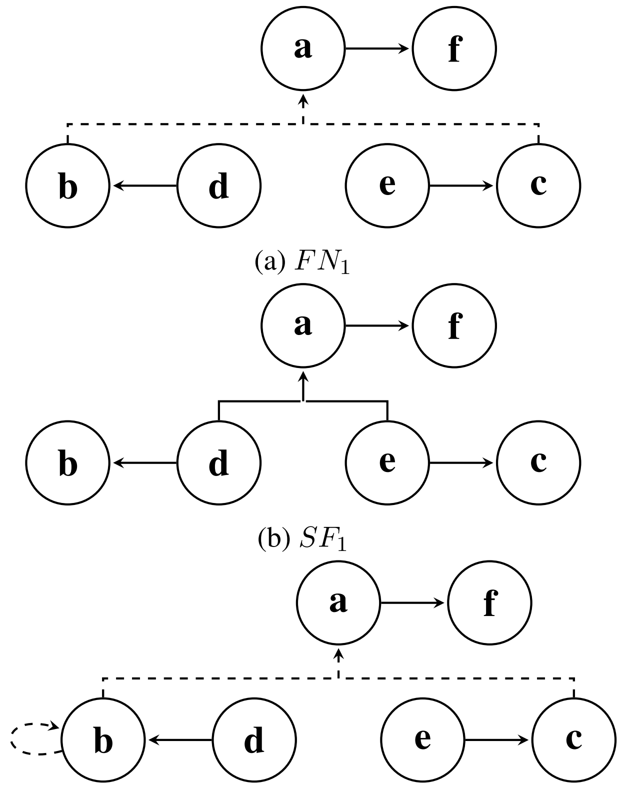
However, we can observe that tracing the supporters of an argument only structurally, without taking the semantics into account, might lead to undesirable results. Not every argument in the framework may be valid and not every support path in it may be valid (see Section 4.3). Propagating the attack based on them may lead to undesired results. Let us consider the framework FN_2, depicted in Figure 96c, which is a modification of FN_1 that changes b into a self–supporter. In this case, we would have that $\{e, f\}$ is an admissible extension – however, $\{e, f\}$ is not admissible in SF_1. Changing the group $(\{d, e\}, a)$ attack into $(\{e\}, a)$ would address this issue.

Translation 63. *Let $FN = (A, R, N)$ be an AFN and $FN^{wv} = (A', R', N')$ its weak validity form. The corresponding attack propagated SETAF is $SF^{FN} = (A', R'')$, where R'' is created as follows:*

- *for an argument $a \in A'$, let $\{X_1^a, ..., X_n^a\}$ be the collection of all coherent sets on A' s.t. $a \in X_i^a$ and let $Z_i^a = \{b \in A' \mid \exists c \in X_i^a, (b, c) \in R'\}$ be the set of all arguments attacking X_i^a in R', and*
- *$R'' = \{(Z', a) \mid a \in A', Z' \subseteq \bigcup_{i=1}^n Z_i^a$ s.t. $\forall_{i=1}^n Z' \cap Z_i^a \neq \emptyset\}$.*

Remark. Please note that for every attack $(a, b) \in R'$ there will exist an attack $(\{a\}, b) \in R''$. Since a attacks b, then it naturally attacks every coherent set containing b. This means that $\{a\}$ forms a sufficient propagated attack on b.

We would like to observe that there exists an alternative to this translation that does not remove invalid arguments. It leaves them in the framework, but allows every argument in a framework to attack them. Although the admissible and preferred extensions are



(a) FN_1

(b) SF_1

(c) FN_2

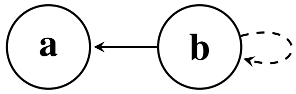
Figure 96: Sample AFNs

preserved in this manner, the complete and grounded ones are not. Thus, we consider the current approach more desirable. Let us look at an example:

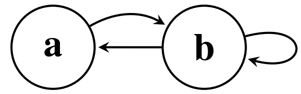
Example 118. Let us consider a simple AFN $FN = (\{a,b\}, \{(b,a)\}, \{(\{b\},b)\})$ depicted in Figure 97a . By not removing the invalid argument b and making it attacked by every other argument in the framework (including b itself), we obtain the SETAF $SF = (\{a,b\}, \{(\{a\},b), (\{b\},a), (\{b\},\{b\})\})$ visible in Figure 97b. The admissible extensions of SF are $\emptyset$ and $\{a\}$ and they are the same as in FN. In both frameworks, $\{a\}$ is a preferred extension as well. However, both $\emptyset$ and $\{a\}$ are also complete in SF, while only $\{a\}$ is complete in FN. This is due to the fact that b is in the discarded set of $\emptyset$ as well as $\{a\}$. Consequently, also the grounded extensions differ between the two frameworks.

A possible approach to address this situation would also mean the removal of attacks carried out by the invalid arguments. Nevertheless, the required modifications appear to be more invasive than simply using the weak validity form. Thus, we will keep this approach in mind, and proceed with Translation 63.

It is natural to ask whether the SETAFs produced with our method have any special properties or restrictions. To the best of our knowledge, the only real issue with the target framework is the excessive data, similarly as e.g. in Translation 8.5.1:



(a) Sample AFN



(b) Associated SETAF without argument removal

Figure 97: Sample AFN and its associated SETAF without argument removal

Example 119. Let $(\{a, b, c, d, e\}, \{(a, b), (c, d)\}, \{(\{b\}, e), (\{d\}, e)\})$ be the simple, minimal AFN depicted in Figure 98. There are three coherent sets for e – $\{b, d, e\}$, $\{a, b, d, e\}$ and $\{b, c, d, e\}$, though only the first one is minimal. All of them are attacked by sets $\{a\}$, $\{b\}$ and $\{a, b\}$. Consequently, all of those sets will be carrying out group attacks against e in the target framework, even though the $\{a, b\}$ conflict is redundant.

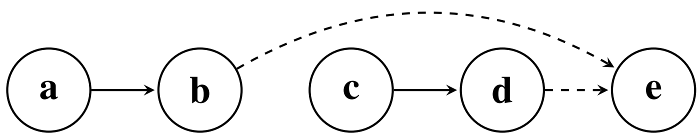


Figure 98: Sample AFN

Just like in the previous case, this issue can be addressed by assuming minimality both on the coherent sets and on the Z' set construction. Nevertheless, we will focus our analysis on the current definition of the corresponding framework. Based on the presented example, we can only conclude that the target SETAF is not necessarily in any interesting normal form, even though the source AFN might be:

Theorem 10.8. *Let $FN = (A, R, N)$ be an AFN and $SF^{FN} = (A', R')$ its corresponding AFN obtained through Translation 63. Then, SF^{FN} might not be in minimal normal form, even if FN is.*

Let us move on to the semantics. We can observe that the attack propagation translation detaches the supporters of a given argument from this argument itself, which causes certain issues with coherence, same way it did with (inverse) closure in BAFs. Moreover, indirect conflicts become direct, and as the AFN definition of conflict–freeness takes into account only the latter, this semantics is not preserved either. As a result, the translation gains strength only when we reach the complete semantics. Please note that the strongly coherent sets will be tied to conflict–free ones based on the similarity relation (see Definition 3.2).

Theorem 10.9. *Let $FN = (A, R, N)$ be an AFN and $SF^{FN} = (A', R'')$ its corresponding attack propagated SETAF obtained by Translation 63. If $E \subseteq A$ is strongly coherent in FN, then it is conflict–free in SF^{FN}. It does not necessarily hold for conflict–free semantics. If E is a σ–extension of FN, where $\sigma \in \{$admissible, preferred, complete,* *grounded, stable$\}$, then it is a σ–extension of SF^{FN}. If $E' \subseteq A'$ is a σ'–extensions of SF^{FN}, where $\sigma' \in \{$conflict–free, preferred, complete, grounded, stable$\}$, then it is also a σ'–extension of FN. It does not necessarily hold for admissible semantics. If E' is conflict–free in SF^{FN}, then it is not necessarily strongly coherent in FN.*

Let us now put these results into our system. Please note we will redefine our approach as a two–step one, similarly as in Translations 10 and 11. Consequently, we will first define a sub–translation only for the weakly valid AFNs, and then shift it to a general approach.

Translation 64. *Let WV^{AFN} be the collection of all weakly valid AFNs and Fr^{SETAF} the collection of SETAFs, both based on argument domain $\mathcal{U}$. The attack propagation translation $awv\text{-}Tr^{AFN}_{SETAF} : WV^{AFN} \rightarrow Fr^{SETAF}$ is defined as $awv\text{-}Tr^{AFN}_{SETAF}((A, R, N)) = (A, R')$ for a framework $(A, R, N) \in WV^{AFN}$, where $R' = \{(Z', a) \mid a \in A, Z' \subseteq \bigcup_{i=1}^{n} Z_i^a$ s.t. $\forall_{i=1}^{n} Z' \cap Z_i^a \neq \emptyset\}$, $\{X_1^a, ..., X_n^a\}$ is the collection of all coherent sets on A s.t. $a \in X_i^a$ and $Z_i^a = \{b \in A \mid \exists c \in X_i^a, (b, c) \in R\}$ is the set of all arguments attacking X_i^a in R.*

The semantics theorem for this translation follows straightforwardly from Theorem 10.9 and its proof:

Theorem 10.10. *Let $\sigma \in \{$complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the identity casting functions for σ. The translation $awv\text{-}Tr^{AFN}_{SETAF}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under the strongly coherent – conflict–free and admissible semantics and identity casting function. It is $\supseteq$–weak under the conflict–free semantics and identity casting functions.*

Analysis of Translation 64: Under the conflict–free, (conflict–free) strongly coherent, admissible, complete, preferred, grounded and stable semantics and their identity casting functions, the translation $awv\text{-}Tr^{AFN}_{SETAF}$ is:

- source–subclass, target–subclass, overlapping
- argument domain preserving, argument preserving, attack relation introducing and support relation removing
- generic and semantics domain preserving
- semantical

Translation $awv\text{-}Tr^{AFN}_{SETAF}$ is not modular. It is exact under the complete, preferred, grounded and stable semantics and the identity casting functions. We classify this approach as an attack propagation translation.

Explanation. We are currently limiting ourselves only to weakly valid AFNs, thus our approach is source–subclass. For now, we can also classify it as target–subclass. The reason behind it is the fact that due to the redundancies introduced by the translation, it holds that if two sets E and E' attack a given argument, then so does $E \cup E'$. Thus, we can observe a certain maximization of the conflicts, rather than minimization. If we decided to optimize our translation in the way we have discussed previously by adding minimality constraints, then the produced SETAFs would be in minimal forms, and this again produces a subclass of our SETAFs. In none of those cases the translation is injective. Let us look again at the framework $(\{a, b, c, d, e\}, \{(a, b), (c, d)\}, \{(\{b\}, e), (\{d\}, e)\})$ described in Example 119 and consider its modification $(\{a, b, c, d, e\}, \{(a, b), (c, d), (a, e)\}, \{(\{b\}, e), (\{d\}, e)\})$, which includes an additional attack from a to e. The propagated attacks for e would still come from sets $\{a\}$, $\{b\}$ and $\{a, b\}$. The target SETAF would be identical in both cases, whether we use the original translation or one with minimality restrictions.

We do not remove or add any arguments, however, previously "indirect" attacks become direct and thus we can speak about conflict introduction. We also choose to classify our approach as support relation removing, as the arguments, previously connected by support, can become completely detached from one another and no real positive interaction between them can be observed. This is particularly visible when we consider the behavior of the admissible semantics in our translation, i.e. the target admissible extensions might not be even coherent sets of the source structure.

Our translation is easily generic, semantics domain preserving and semantical. Its exactness under the listed semantics comes from Theorem 10.10. Unfortunately, the approach is not modular. We can consider two trivial AFNs $FN_1 = (\{a, b\}, \{(a, b)\}, \emptyset)$ and $FN_1 = (\{b, c\}, \emptyset, \{(\{b\}, c)\})$. Due to the absence of c in the first one and a in the other, there will be no conflict between these two arguments in the corresponding SETAFs. Thus, there will be no conflict in the union of the produced frameworks either. However, it is easy to see that there will be a propagated attack from $\{a\}$ to c in the SETAF associated with $FN_1 \cup FN_2$. Therefore, $awv\text{-}Tr^{AFN}_{SETAF}$ is not modular. ■

We can now redefine the original translation fully, reusing the weak validity translation $wv\text{-}Tr^{AFN}$ (see Translation 7):

Redefinition of Translation 63: *Let Fr^{AFN} be the collection of all AFNs and Fr^{SETAF} the collection of all SETAFs, both on domain $\mathcal{U}$. The translation $a\text{-}Tr^{AFN}_{SETAF} : Fr^{AFN} \rightarrow Fr^{SETAF}$ is defined as $a\text{-}Tr^{AFN}_{SETAF}((A, R, N)) = awv\text{-}Tr^{AFN}_{SETAF}(wv\text{-}Tr^{AFN}((A, R, N)))$ for a framework $(A, R, N) \in Fr^{AFN}$.*

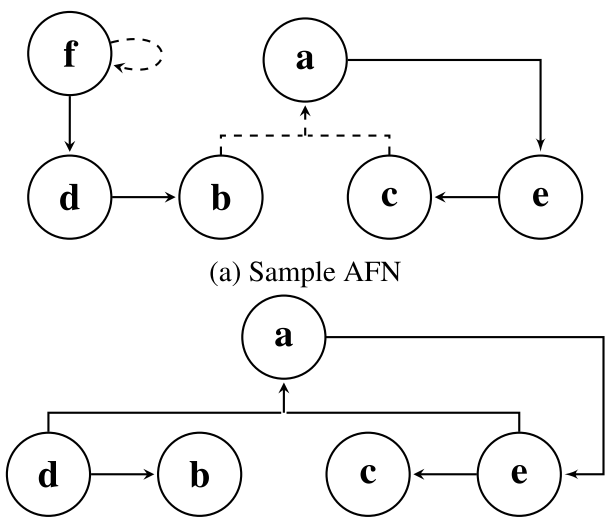
Redefinition of Theorem 10.9: *Let $\sigma \in \{$complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} be the identity casting functions for σ. The translation $a\text{-}Tr^{AFN}_{SETAF}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under the strongly coherent – conflict–free and admissible semantics and identity casting function. It is $\supseteq$–weak under the conflict–free semantics and identity casting functions.*

The properties of our transformation are now simply the result of the properties of Translations 7 and 64. Thus, we will omit further explanations.

Analysis of Translation 63: Under the conflict–free, (conflict–free) strongly coherent, admissible, complete, preferred, grounded and stable semantics and their identity casting functions, the translation $a\text{-}Tr^{AFN}_{SETAF}$ is:

- full, target–subclass, overlapping
- argument domain preserving, argument removing, induced attack relation removing, attack relation introducing and support relation removing
- generic and semantics domain preserving
- semantical

Translation $a\text{-}Tr^{AFN}_{SETAF}$ is not modular. It is exact under the complete, preferred, grounded and stable semantics and the identity casting functions. We classify this approach as an attack propagation translation.



(a) Sample AFN

(b) Associated attack propagation SETAF

Figure 99: Sample AFN and its associated attack propagation SETAF

Example 120. Let us consider the AFN $FN = (\{a, b, c, d, e, f\}, \{(a, e), (d, b), (e, c), (f, d)\}, \{(\{b, c\}, a), (\{f\}, f)\})$ depicted in Figure 99a and previously analyzed in Example 16. The minimal form of its associated attack propagation SETAF is $SF^{FN} = (\{a, b, c, d, e\}, \{(\{a\}, e), (\{d\}, b), (\{e\}, c), (\{d, e\}, a)\})$ [25] The conflict–free extensions of SF^{FN} are $\emptyset$, $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$, $\{a, b\}$, $\{a, c\}$, $\{a, d\}$, $\{b, c\}$, $\{b, e\}$, $\{c, d\}$, $\{d, e\}$, $\{a, b, c\}$ and $\{a, c, d\}$. We can observe that not every conflict–free extension of FN is conflict–free in SF^{FN}. For example, the argument f is not present in the framework. However, the set $\{a, d, e\}$ is also conflict–free in FN, even though it is not strongly coherent. Nevertheless, it is missing from the conflict–free extensions of SF^{FN}. At the same time, we can observe that all conflict–free sets of SF^{FN} are conflict–free in FN. Additionally, all of the strongly coherent sets $\emptyset$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$, $\{a, b\}$, $\{a, c\}$, $\{b, c\}$, $\{b, e\}$,

[25] We use the minimal form in order to improve readability.

$\{c,d\}$, $\{d,e\}$, $\{a,b,c\}$ and $\{a,c,d\}$ of FN are conflict–free in SF^{FN}, though please note that our SETAF produces more extensions (for example, $\{a\}$). The admissible extensions of SF^{FN} are $\emptyset$, $\{a\}$, $\{d\}$, $\{a,c\}$, $\{a,d\}$, $\{d,e\}$, and $\{a,c,d\}$. We can observe that all of the admissible sets of FN are admissible in SF^{FN}, but not the other way around – in particular, $\{a\}$ and $\{a,d\}$ are not even strongly coherent in FN. Nevertheless, when we reach the complete extensions – $\{d\}$, $\{d,e\}$ and $\{a,c,d\}$ – we obtain an exact correspondence between the answers produced by the two frameworks. We can observe that $\{d\}$ is grounded both in FN and SF^{FN}, while $\{d,e\}$ and $\{a,c,d\}$ are preferred and stable. Thus, starting with the complete semantics, our translation becomes strong and exact.

10.2.2 Defender

Attack–based argumentation framework in fact have one type of positive, indirect relation between arguments – defense. Thus, with the use of auxiliary arguments and attacks, we can connect supporters of an argument to the argument itself. However, we can recall that the defender BAF–AF Translation 58 did not behave that well due to support cycles. Since AFN semantics handle such cycles differently from BAFs, this issue is resolved by letting only "valid" arguments participate in defense. Unfortunately, this also leads to the fact that it is not just direct supporters that have to defend us, unlike in Translation 58:

Example 121. Let us consider a simple AFN $FN = (\{a,b,c\},\ \emptyset,\ \{(\{a,b\},c), (\{a,c\},b)\})$. We transform it into a defender SETAF by introducing an auxiliary attacker for every argument requiring support, which can be interpreted as "this argument is unsupported". In the case of argument c, the additional attacker will be attacked by a and b – in our AFN, the presence of either of them is sufficient to support a (i.e. both $\{a\}$ and $\{b\}$ have an element in common with every supporting set of c, which in this case is just $\{a,b\}$). Similar analysis can be carried out for c. This gives us the framework $SF^{FN} = (\{a,b,c,b',c'\},\ \{(\{b'\},b),\ (\{c'\},c),\ (\{a\},c'),\ (\{b\},c'),\ (\{a\},b'),\ (\{c\},b')\})$. However, we can observe that with this construction, the set $\{b,c\}$ is admissible in our SETAF, while it is not even coherent in FN.

While the construction above ensures that the accepted arguments are supported, it does not mean that they are supported in a valid manner. In other words, the collection of defenders, defenders of defenders etc. might not form a powerful sequence. Therefore, in our construction the attacks on auxiliary arguments will not be carried out only by the direct supporters of a given argument, but by the members of powerful sequences for this arguments. This brings us to the following formulation:

Translation 65. *Let $FN = (A,R,N)$ be an AFN. Its corresponding defender SETAF $SF^{FN} = (A',R')$ is constructed the following way:*

- $A' = A \cup \{a' \mid a \in A \wedge \exists C \subseteq A, CNa\}$. *The meaning of a' is " a is not powerful",*
- *add to R' all attacks from R and attacks from a': $R' = \{(\{a\},b) \mid (a,b) \in R\} \cup \{(\{a'\},a) \mid a \in A \wedge \exists C \subseteq A, CNa\}$, and*

- *let $a \in A$ s.t. $\exists C \subseteq A, CNa$. For any minimal powerful sequence $(a_0, ..., a_n)$ for a, add $(\{a_0, .., a_{n-1}\}, a')$ to R'.*

We can now focus on the semantics. Please note that the construction of the target arguments corresponding to a given source one is not going to be as straightforward as in e.g. certain EAF–SETAF Translation 40. We cannot use the discarded set anymore to say which primed arguments need to be added. This is because we use auxiliary arguments to represent support, not attack, and an argument itself can be attacked despite all of its supporters accepted and unchallenged.

Theorem 10.11. *Let $FN = (A, R, N)$ be an AFN and $SF^{FN} = (A', R')$ its corresponding defender SETAF obtained by Translation 65. By $E_{np} = \{a' \mid$ there is no coherent set containing $a\} \cup \{a' \mid$ for every coherent set C for a, $\exists e \in E, c \in C \setminus \{a\}, (e, c) \in R\}$ we will denote primed arguments corresponding to a subset of E^{att}, in which every argument a either has no coherent set or every of its coherent sets is attacked by E on an argument different from a.*

If a set $E \subseteq A$ is conflict–free in FN, then it is conflict–free in SF^{FN}. The set $E \cup E_{np}$ is not necessarily conflict–free in SF^{FN}. If a set E is strongly coherent in FN, then $E \cup E_{np}$ is conflict–free in SF^{FN}. If E is a σ–extension of FN, where $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$, then $E' = E \cup E_{np}$ is a σ–extension of SF^{FN}.

If a set $E' \subseteq A'$ is a σ–extension of SF^{FN}, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, then $E = E' \cap A$ is a σ–extension of FN. If E' is conflict–free, $E = E' \cap A$ does not have to be strongly coherent in FN.

Redefinition of Translation 65: *Let Fr^{AFN} be the collection of all AFNs on the domain $\mathcal{U}$ and Fr^{SETAF} the collection of all SETAFs on argument domain $\mathcal{U} \cup \mathcal{U}'$. The translation $def\text{-}Tr^{AFN}_{SETAF} : Fr^{AFN} \to Fr^{SETAF}$ is defined as $def\text{-}Tr^{AFN}_{SETAF}((A, R, N)) = (A', R')$ for a framework $(A, R, N) \in Fr^{AFN}$, where $A' = A \cup X'$ for $X' = \{a' \mid a \in A \wedge \exists C \subseteq A, CNa\}$, and $R' = \{(\{a\}, b) \mid (a, b) \in R\} \cup \{(\{a'\}, a) \mid a \in A, a' \in X'\} \cup \{(\{a_0, .., a_{n-1}\}, a') \mid a \in A,\ a' \in X,\ (a_0, ..., a_n)$ is a minimal powerful sequence for $a\}$.*
Redefinition of Theorem 10.11: *Let $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the removal casting functions for σ defined as $SC^{X}_{\sigma}(E) = E \cap A$ for $X = (A, R, N) \in Fr^{AFN}$ and $E \in \sigma(def\text{-}Tr^{AFN}_{SETAF}(X))$. The translation $def\text{-}Tr^{AFN}_{SETAF}$ is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under strongly coherent–conflict–free semantics and the defined casting functions.It is semantics bijective under complete, preferred, grounded and stable semantics and the defined casting functions.*
Analysis of Translation 65: Under the conflict–free, (conflict–free) strongly coherent, admissible, complete, preferred, grounded and stable semantics and their removal casting functions, the translation $def\text{-}Tr^{AFN}_{SETAF}$ is:

- full, target–subclass, overlapping
- weakly argument domain altering, argument introducing, induced attack relation introducing and support removing

- generic and weakly semantics domain altering
- semantical

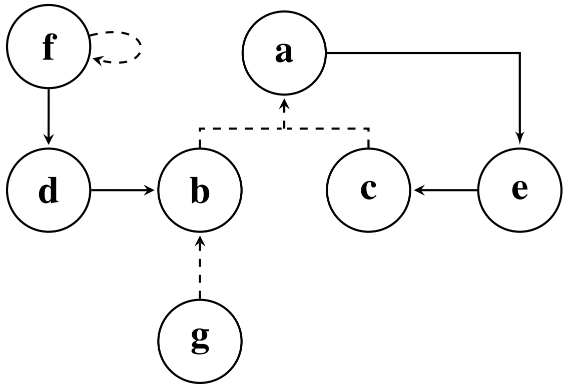
Translation $def\text{-}Tr^{AFN}_{SETAF}$ is not modular. It is faithful under the complete, preferred, grounded and stable semantics and the defined removal casting functions. We classify this approach as a defender translation.

Explanation. Any AFN can undergo the translation, and thus we classify it as full. Due to the domain change, it is also target–subclass; for example, it is not possible to create a SETAF containing only primed arguments. Unfortunately, our approach is overlapping, and it would be so independently of whether we would use minimal or all powerful sequences in the construction of attacks – knowledge on the sequences is not always enough to reconstruct the framework. We can consider two frameworks $(\{a,b,c\}, \emptyset, \{(\{a,b\},b),(\{b\},c)\})$ and $(\{a,b,c\}, \emptyset, \{(\{a\},b),(\{b\},c)\})$. The produced framework is in both cases $(\{a,b,c,b',c'\}, \{(\{b'\},b),(\{c'\},c),(\{a\},b'),(\{a,b\},c')\})$. Thus, even though due to defense from primed arguments certain positive relations can be deduced, some of them – in particular, the invalid ones – are lost. For this reason we consider the relation support removing. At the same time, we introduce auxiliary arguments, and attacks from/to them – thus, the attack introduction is induced. Our argument and semantics domain are altered only in a weak manner. The amount of handled semantics classifies the translation as generic. Due to the fact that $def\text{-}Tr^{AFN}_{SETAF}$ is based on the notion of coherence (i.e. powerful sequences), we classify it as semantical.

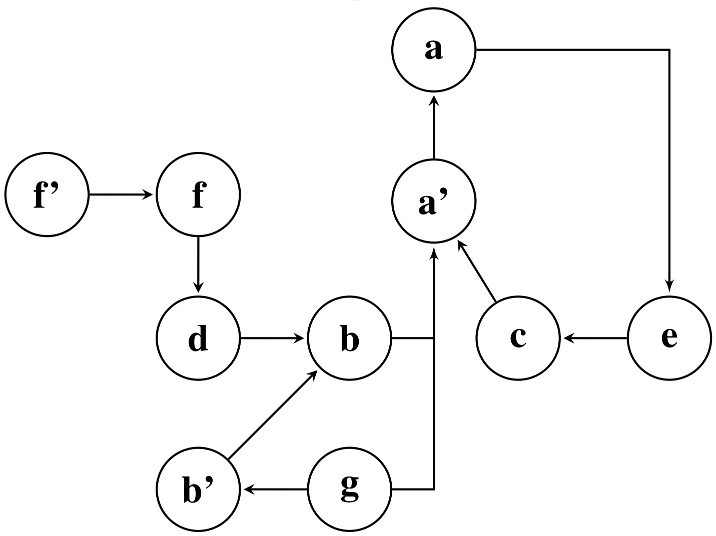
Unfortunately, the defender approach is not modular in AFNs. We can consider two frameworks $FN_1 = (\{a,b\}, \emptyset, \{(\{a\},b)\})$ $FN_2 = (\{b,c\}, \emptyset, \{(\{c\},b)\})$. Their corresponding SETAFs are $SF_1 = (\{a,b,b'\}, \{(\{a\},b'), (\{b'\},b)\})$ and $SF_2 = (\{c,b,b'\}, \{(\{c\},b'), (\{b'\},b)\})$. Thus, their union is simply $(\{a,b,c,b'\}, \{(\{a\},b'), (\{c\},b'), (\{b'\},b)\})$. However, the framework associated with $FN_1 \cup FN_2$ is $(\{a,b,c,b'\}, \{(\{a,c\},b'), (\{b'\},b)\})$, which is clearly not the same Thus, our approach is not modular. ∎

Example 122. Let us consider the AFN $FN = (\{a,b,c,d,e,f,g\}, \{(a,e), (d,b), (e,c), (f,d)\}, \{(\{b,c\},a), (\{f\},f), (\{g\},b)\})$ depicted in Figure 100a. Its coherent sets include $\emptyset$, $\{c\}$, $\{d\}$, $\{e\}$, $\{g\}$, $\{a,c\}$, $\{b,g\}$, $\{a,b,g\}$ and any of their combinations. We can observe that f does not appear in any of them - it does not possess a powerful sequence in the framework. The strongly coherent sets are $\emptyset$, $\{c\}$, $\{d\}$, $\{e\}$, $\{g\}$, $\{a,c\}$, $\{b,g\}$, $\{c,d\}$, $\{c,g\}$, $\{d,e\}$, $\{d,g\}$, $\{e,g\}$, $\{a,b,g\}$, $\{a,c,d\}$, $\{a,c,g\}$, $\{b,c,g\}$, $\{b,e,g\}$, $\{c,d,g\}$, $\{d,e,g\}$, $\{a,b,c,g\}$ and $\{a,c,d,g\}$. From this, the sets $\emptyset$, $\{d\}$, $\{g\}$, $\{a,c\}$, $\{d,e\}$, $\{d,g\}$, $\{a,c,d\}$, $\{a,c,g\}$, $\{d,e,g\}$ and $\{a,c,d,g\}$ are admissible. Fortunately, only $\{d,g\}$, $\{d,e,g\}$ and $\{a,c,d,g\}$ are complete, with the first set being grounded and the other two preferred and stable.

The SETAF associated with FN is $SF^{FN} = (\{a,b,c,d,e,f,g,a',b',f'\}, \{(\{a\},e), (\{d\},b), (\{e\},c), (\{f\},d), (\{a'\},a), (\{b'\},b), (\{f'\},f), (\{b,g\},a'), (\{c\},a')\})$, as seen in



(a) Sample AFN



(b) Associated defender SETAF

Figure 100: Sample AFN and its associated defender SETAF

Figure 100b. Due to the fact that SF^{FN} has 65 conflict–free extensions, we will not listed them here. We can however observe that sets such as $\{a\}$, $\{b\}$ and so on are conflict–free in SF^{FN}, but are not strongly coherent in FN. The admissible extensions of our SETAF are $\emptyset$, $\{f'\}$, $\{f', d\}$, $\{g\}$, $\{f', g\}$, $\{a, c\}$, $\{a, c, f'\}$, $\{a', d, e, f'\}$, $\{d, f', g\}$, $\{a, c, d, f'\}$, $\{a, c, g\}$, $\{a, c, f', g\}$, $\{a', d, e, f', g\}$ and $\{a, c, d, f', g\}$. We can observe that if we remove the auxiliary arguments, we obtain all and only admissible extensions of FN, even though certain source sets can be obtained from a number of target ones. Out of all these admissible extensions, only $\{d, f', g\}$, $\{a, c, d, f', g\}$ and $\{a', d, e, f', g\}$ are complete. They again correspond to the complete extensions of FN; this time, the relation is one–to–one. It is now easy to show that the grounded, preferred and stable extensions of our framework SF^{FN} are also in agreement with the ones produced by FN.

10.2.3 Improvements

Let us first focus on the possible ways of improving the attack propagation translation in terms of semantical and complexity properties. Although this approach is not strong under the conflict–free and admissible semantics, exact approaches can be created. In the first case, we can reuse the specialized Translation 62 that we have created for AFs and use the AF–SETAF approach. In the latter, the results presented in Section 10.1.1 point to the existence of an appropriate method, even though we are not entirely sure how it looks like. However, in the SETAF setting, we could try to repeat the approach from Translation 54 and introduce additional, self–attacking arguments. In this case the attacks at them would be carried out by the (coherent) sets from which a given argument can follow, not single arguments. Nevertheless, we would face the loss of stability, which as such is normally a more prominent semantics.

Just like in the coalition AFN–AF translation, the attack propagation AFN–SETAF approach is semantical. The new conflicts are not based on N as such, but on the coherent sets derived from it. Similarly as in the previous case, in order to focus on supporters as such rather than on their validity, we need to ensure that however we trace the support an argument receives, we will always end up with a coherent set or a powerful sequence. This means that the first restriction we need in order to obtain a semi–structural translation, is to require that the source framework is strongly valid. With this assumption, we can replace the coherent sets in Translation 63 with sets simply supporting their members thanks to Theorem 4.32. This brings us down from a semantical to a semi–structural approach according to our classification, even though some computation is still required. The translation can be further simplified by assuming that the source AFN is not only strongly valid, but also binary – in this case, we come back to the BAF–AF Translation 53 parametrized with secondary attacks (see Translation 59). A different option would be to assume that every argument providing support cannot be supported itself, which is a more general version of the AFN subclass with support depth 1. In a certain sense, this might bring us even closer to the structural level, as the creation of sets providing sufficient support becomes rather straightforward. Nevertheless, a much more interesting improvement can be obtained in the next approach – the defender translation.

Let us now consider a modification of the defender AFN–SETAF translation, tailored to strongly valid frameworks. This limitation allows us to consider only the direct supporters of a given argument in defense, not the whole powerful sequence, in the construction of the target SETAF:

Translation 66. *Let $FN = (A, R, N)$ be a strongly valid AFN. Its corresponding defender SETAF $SF^{FN} = (A', R')$ is constructed the following way:*

- $A' = A \cup \{a' \mid a \in A \wedge \exists C \subseteq A, CNa\}$*. The meaning of a' is "a is unsupported",*
- *add to R' all attacks from R and attacks from a': $R' = \{(\{a\}, b) \mid (a, b) \in R\} \cup \{(\{a'\}, a) \mid a \in A \wedge \exists C \subseteq A, CNa\}$, and*

- *let $a \in A$ s.t. $\exists C \subseteq A, CNa$, and let $\{Z_1, .., Z_n\}$ be the collection of all and only sets on A s.t. Z_iNa. For all minimal $Z' \subseteq \bigcup_{i=1}^{n}$ s.t. $\forall_{i=1}^{n} Z' \cap Z_i \neq \emptyset$, add (Z', a') to R'.*

Although now the framework looks a little bit different, a similar semantics theorem holds. Please note that due to the fact that we are dealing with strongly valid frameworks, the construction of the E_{np} set can be simplified. Most notably, the coherent sets can be replaced by sets in which every argument is just supported through the N relation, i.e. we do not need any validity checks (see Theorem 4.32). Nevertheless, for a lack of better name, we will still use the notion of coherent sets in the theorem, and only remark that the way to obtain them is easier.

Theorem 10.12. *Let $FN = (A, R, N)$ be a strongly valid AFN and $SF^{FN} = (A', R')$ its corresponding defender SETAF obtained by Translation 66. By $E_{np} = \{a' \mid$ for every coherent set C for a, $\exists e \in E, c \in C \setminus \{a\}, (e, c) \in R\}$ we will denote primed arguments corresponding to a subset of E^{att} in which for every argument a and any coherent set for it, there is a member of this set attacked by E different from a.*

If a set $E \subseteq A$ is conflict–free in FN, it is conflict–free in SF^{FN}. The set $E \cup E_{np}$ is not necessarily conflict–free in SF^{FN}. If a set E is strongly coherent in FN, then $E \cup E_{np}$ is conflict–free in SF^{FN}. If E is a σ–extension of FN, where $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$, then $E' = E \cup E_{np}$ is a σ–extension of SF^{FN}.

If a set $E' \subseteq A'$ is a σ–extension of SF^{FN}, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, then $E = E' \cap A$ is a σ–extension of FN. If E' is conflict–free, the set $E = E' \cap A$ does not have to be strongly coherent in FN.

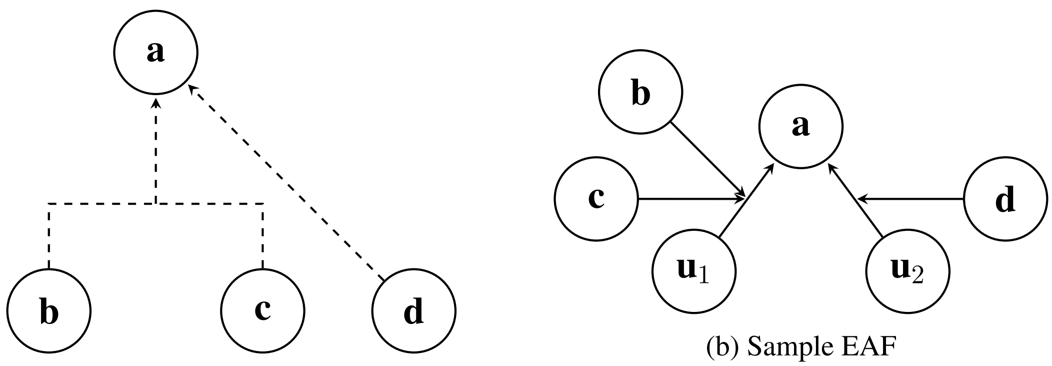
The same properties as in the original version of this translation hold and thus we will omit further analysis. The only difference is that now we can classify the approach as semi–structural and source–subclass. The transformation is still support removing due to the minimality assumption on the construction of Z' sets; nevertheless, this is for now a rather minor issue and can be addressed in future work.

10.3 AFN as EAF

Although we believe that in order for EAFs to handle AFNs one should use the AF bypass, in this section we would like to discuss a certain connection between defense attacks and necessary support. In Section 8.6.1 we have already shown that defense attack can be seen as a particular form of support, which we referred to as overpowering. We have also proved that with the use of auxiliary arguments, AFNs can handle defense attack (see Section 8.5). Moreover, what can already be seen from the name, this form of conflict can also be seen as type of defense due to its ability to overrule attacks. Thus, after the presented defender translation, it is natural to ask whether this form of conflict can handle necessary support. In this section we will sketch two methods that will highlight certain possibilities and obstacles in transforming support to this particular type of attack. The first approach basically adapts the defender translation by the use defense attacks, as shown in the example below.

Example 123. Let us consider a simple AFN $(\{a,b,c,d\}, \emptyset, \{(\{b,c\},a), (\{d\},a)\})$, as depicted in Figure 101a. Argument a is supported by sets $\{b,c\}$ and $\{d\}$. Just like in the defender approach, we can introduce "you are unsupported" arguments, though this time it will be one argument per support set. Instead of making the supporters attack the auxiliary arguments, we make it override the conflicts they carry out. This gives us the EAF $(\{a,b,c,d,u_1,u_2\}, \{(u_1,a),(u_2,a)\}, \{(b,(u_1,a)),(c,(u_1,a)),(d,(u_2,a))\})$ visible in Figure 101b.

While in the first case a cannot be accepted without b and d or c and d being in the set due to support, in the latter they have to be present in order to override the attacks from u_1 and u_2 – otherwise, u_1 and u_2 defeat a and admissibility is breached.



(a) Sample AFN

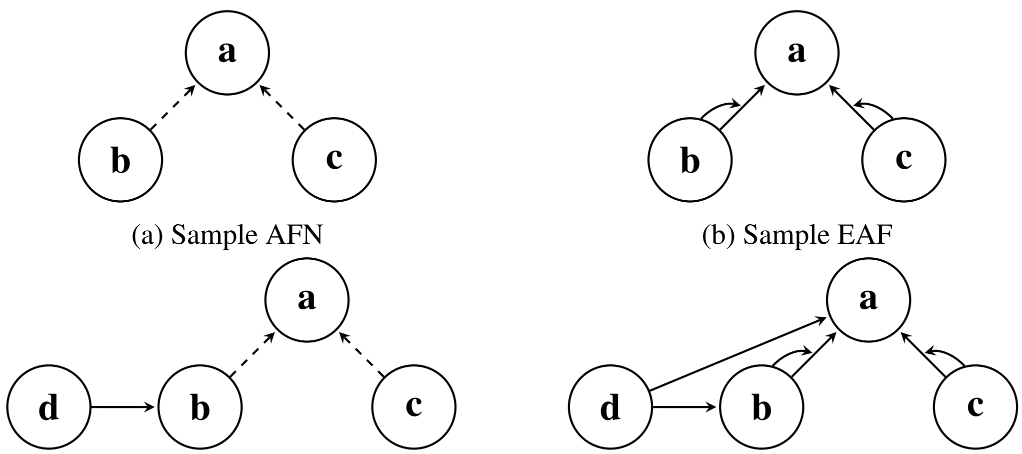
(b) Sample EAF

Figure 101: Necessary support to defense attack, first approach

Please note that this approach of course has its drawbacks. Directing the attacks from supporters not at the auxiliary arguments, but that at the relations, makes it possible for all of these arguments to appear together in a single extension. Thus, arguments saying "you are supported" and "no, you are not" can be jointly accepted and this can be somewhat confusing. Moreover, the example above deals with a rather trivial AFN, and if we wanted to create an actual translation that would handle support cycles etc., we would sooner or later come back to the original defender transformation. Consequently, there is no obvious gain to proceed further with this approach and we will limit ourselves just to this discussion.

The other approach can be seen as a merge between defender and attack propagation. In this case we have to limit ourselves to binary AFNs, i.e. ones where supporting sets contain only single arguments. The support is again simulated by defense attack, but this time aimed at the attack coming from the argument itself rather than from an auxiliary element.

Example 124. Let $(\{a, b, c\emptyset, \{(\{b\}, a), (\{c\}, a)\})$ be the AFN depicted in Figure 102a. By replacing supports with attacks and allowing the supporters to defense attack them, we obtain the EAF $(\{a, b, c\}, \{(b, a), (c, a)\}, \{(b, (b, a)), (c, (c, a))\})$ from Figure 102b. Just like a requires both of its supporters to be presented in AFN, a needs b and c to appear in an extension in order to override the attacks and be defended in EAF. However, the addition of attack makes certain things complicated. Consider our AFN extended with the (d, b) attack, as depicted in Figure 102c. If we extended our corresponding EAF with the same attack only, d would in fact defend a from b and thus make the set $\{a, c, d\}$ admissible, which is not an intended result. The fact that d indirectly renders a unacceptable in AFN needs to be propagated in the EAF. Consequently, we include also the (d, a) attack in the framework, and obtain the structure visible in Figure 102d. Only then a is properly rendered unacceptable.



(a) Sample AFN (b) Sample EAF

(c) Sample AFN (d) Sample EAF

Figure 102: Necessary support to defense attack, second approach

Although the method we have just presented has some potential, one has to bear in mind that it applies only to binary support. Moreover, there is no clear answer on how to handle the cycle issue and translating a framework that is not strongly valid would probably be highly problematic. Nevertheless, we believe this discussion provided some insight on the relation between support and defense attack.

10.4 AFN as BAF

In Section 9.2, we have discussed a translation from BAFs to AFNs, and noted that it can be done only for frameworks that have an acyclic support graph. We have also seen how secondary attack can be used to simulate attacking an argument through its coherent sets. For the same reasons as before, translating an AFN to BAF can be done only for strongly valid frameworks; the difference in handling support cycles is too prominent.

Unfortunately, the next issue arises concerning group support. As we could have observed in Example 117 concerning the AFN–SETAF attack propagation translation, the indirect attacks need to be carried out by sets of arguments, not just single arguments. This clearly does not fit the definition of secondary attacks in BAFs. Thus, for now, our best option is to limit ourselves not just to strongly valid, but also support binary AFNs. The resulting translation is almost identical to the BAF–AFN one:

Translation 67. *Let $FN = (A, R, N)$ be a strongly valid and support binary AFN. The associated BAF is $BF^{FN} = (A, R, S)$, where $S = \{(a, b) \mid (\{a\}, b) \in N\}$.*

As expected, the resulting BAF will be support acyclic:

Theorem 10.13. *Let $FN = (A, R, N)$ be a strongly valid and support binary AFN and $BF^{FN} = (A, R, S)$ its associated BAF obtained through Translation 67. Then BF^{FN} is support acyclic.*

The semantics theorem can be proved in the same manner as Theorem 9.14.

Theorem 10.14. *Let $FN = (A, R, N)$ be a strongly valid and support binary AFN, $BF^{FN} = (A, R, S)$ its associated BAF obtained through Translation 67 and $R' = \{R^{sec}\}$ the collection of secondary attacks in BF^{FN}.Then, a set $E \subseteq A$ is:*

- *+conflict–free w.r.t. $\emptyset$ in BF^{FN} iff it is conflict–free in FN.*
- *inverse closed under S in BF^{FN} iff it is coherent in FN.*
- *+conflict–free w.r.t. R' and inverse closed under S in BF^{FN} iff it is strongly coherent in FN.*
- *an i–admissible extension of BF^{FN} w.r.t. (R', R') iff it is admissible in FN.*
- *an i–preferred extension of BF^{FN} w.r.t. (R', R') iff it is preferred in FN.*
- *a d–complete extension of BF^{FN} w.r.t. (R', R') iff it is complete in FN.*
- *a d–grounded extension of BF^{FN} w.r.t. R' iff it is grounded in FN.*
- *a stable extension of BF^{FN} w.r.t. R' iff it is stable in FN.*

The translation is classified in the same manner as Translation 59:

Redefinition of Translation 67: *Let $SBin^{AFN} \cap SV^{AFN}$ the collection of all support binary and strongly valid AFNs and $SAcy^{BAF}$ the collection of all support acyclic BAFs, both on domain $\mathcal{U}$. The translation $Tr^{AFN}_{BAF} : (SBin^{AFN} \cap SV^{AFN}) \rightarrow SAcy^{BAF}$ is defined as $Tr^{AFN}_{BAF}((A, R, N)) = (A, R, S)$ for a framework $(A, R, N) \in (SBin^{AFN} \cap SV^{AFN})$, where $S = \{(a, b) \mid (\{a\}, b) \in N\}$.*

Redefinition of Theorem 10.14: *Let $\sigma^{AFN} \in \{$coherent, strongly coherent, admissible, preferred, complete, grounded, stable$\}$ be an AFN semantics and $\sigma^{BAF} \in \{$inverse closed,*

+conflict–free and inverse closed, i–admissible, i–preferred, d–complete, d–grounded, stable} be a similar BAF semantics with identical parametrization consisting of secondary attacks. Let SC_{σ}^{Tr} be the identity casting functions for σ. The translation Tr_{BAF}^{AFN} is strong and semantics bijective under $(\sigma, SC_{\sigma}^{Tr})$.

Analysis of Translation 67: Under the (inverse closed) coherent, (+conflict–free and inverse closed) strongly coherent, (i–) admissible, (i–) preferred, (d–) complete, (d–) grounded and stable semantics with identical parametrization consisting of secondary attacks and identity casting functions, the translation Tr_{BAF}^{AFN} is:

- source–subclass, target–subclass and injective
- argument domain and structure preserving
- generic, semantics domain preserving and exact
- structural and modular

Translation Tr_{BAF}^{AFN} is classified as basic under the listed semantics and casting functions.

10.5 AFN as EAS

A translation from AFNs to EASs has already been presented in our previous works [77, 78]. While shifting the binary to set–form attack is trivial, the support requires a little bit more consideration. There are two things we need to deal with; one is the structural difference between the necessary support relation N and the evidential support E, particularly visible in the construction of powerful and evidential sequences. The other concerns how the connection between other elements of the framework and the evidence argument should look like.

Let $A_1, ..., A_n$ be sets supporting an argument a in N. A set of arguments S "sufficiently supports" a if S has an element in common with every such A_i, i.e. $\forall_{i=1}^{n}$, $S \cap A_i \neq \emptyset$. Therefore, verifying whether S supports a corresponds to checking whether S satisfies a propositional formula $\bigvee A_1 \wedge ... \wedge \bigvee A_n$, where $\bigvee A_i$ should be understood as a disjunction of elements of A_i. This is also the construction we will be using for the acceptance conditions in the AFN–ADF translations (see Section 10.6). Let us now assume that $A_1, ..., A_n$ support a through E. In this case, S "sufficiently supports" a if at least one such A_i is a subset of S. Consequently, in this case we produce a formula $\bigwedge A_1 \vee ... \vee \bigwedge A_n$, where $\bigwedge A_i$ stands for the conjunction of elements of A_i. Therefore, from the technical side, the translation between the necessary and evidential relations can be seen as a conversion between CNF and DNF.

Let us now deal with the issue of the evidence argument. In EASs, this special argument is the only source of through, the sole confirmation of validity. Every other argument need to be able to trace back to it, as the evidential sequence makes explicit. In AFNs, validity is obtained through acyclicity. In other words, in order for an argument to be valid, it has to be reachable from arguments that require no support, as seen in the powerful sequence. Therefore, if we want unsupported arguments to be able to provide validity in the EAS setting, it is easy to see that they (and only they) should be backed up by η. This leads us to the following translation:

Translation 68. *Let $FN = (A, R, N)$ be an AFN. The corresponding EAS $ES^{FN} = (A', R', E)$ is created as follows:*

- $A' = A \cup \{\eta\}$,
- $R' = \{(\{a\}, b) \mid (a, b) \in R\}$, *and*
- *let a be an argument in A and $Z = \{Z_1, .., Z_n\}$ be a collection of all sets Z_i s.t. $Z_i N a$. If Z is empty, add $(\{\eta\}, a)$ to E. Otherwise, for every subset Z' of $\bigcup_{i=1}^{n} Z_i$ s.t. $\forall_{i=1}^{n} Z' \cap Z_i \neq \emptyset$, add (Z_i, a) to E.*

Although the translation of support presented above is correct and the semantics return the desired extensions in the obtained framework, it is not the most optimal one. It can create redundant elements in E. For example, given argument a s.t. $\{a, b\}Na$ and $\{c\}Na$, our intent would be to receive $\{a, c\}Ea$ and $\{b, c\}Ea$. However, the translation would also give us $\{a, b, c\}Ea$. Although the framework behaves in the desired way (see Theorem 4.11), a cleaner transformation would be more desirable. Please note that it cannot be fixed by assuming that we take into account only minimal sets Z', since the elements of N might not be incomparable in the first place. Even though again the extensions of a framework produced by such a minimal transformation would be satisfactory (see Theorems 4.11 and 4.9 on minimal forms), we would "lose" some of the relations. Therefore, an additional approach was proposed:

Translation 68 (Continued)**.** *Let a be an argument in A and $Z = \{Z_1, .., Z_n\}$ be a collection of all sets Z_i s.t. $Z_i N a$. If Z is empty, add $(\{\eta\}, a)$ to E. Otherwise, for all Z' in $Z_1 \times ... \times Z_n$, add (Z'_S, a) to E, where Z'_S is the set of all elements in Z'.*

This fix can be applied to any translation that suffered from similar issues when converting between group relations. We will use it throughout our analysis. Let us start by considering how an AFN–produced EAS looks like. Although minimality is lost independently of the chosen version of our translation, the validity and consistency forms still hold:

Theorem 10.15. *Let $FN = (A, R, N)$ be an AFN and $ES^{FN} = (A, R, E)$ its associated EAS obtained through Translation 68. Then, ES^{FN} is attack binary. If FN is weakly (relation, strongly) valid, then so is ES^{FN}. If FN is strongly consistent, then so is ES^{FN}. ES^{FN} does not have to be in minimal form, even if FN is.*

The necessary and evidential supports are closely connected. In particular, a powerful sequence can be trivially transformed into an evidential one and the other way around:

Theorem 10.16. *Let $FN = (A, R, N)$ be an AFN and $ES^{FN} = (A', R', E)$ its corresponding EAS obtained through Translation 68. Let $a \in A$ be an argument. If $(a_0, ..., a_n)$ is a powerful sequence for a on $S \subseteq A$ in FN, then $(\eta, a_0, ..., a_n)$ is an evidential sequence for a on $S \cup \{\eta\}$ in ES^{FN}. If $(\eta, a_0, ..., a_n)$ is an evidential sequence for a on $S \subseteq A'$ in ES^{FN}, then $(a_0, ..., a_n)$ is a powerful sequence for a on $S \setminus \{\eta\}$ in FN. If a set $S \subseteq A$ is coherent in FN, then $S \cup \{\eta\}$ is self–supporting in ES^{FN}. If $S' \subseteq A$ is self–supporting in ES^{FN}, then $S' \cap A$ is coherent in FN.*

However, we would like to notice one important difference in the definitions of defense between these two frameworks. In EASs, an argument a has to be e–supported by the set S. Consequently, it does not have to be the case that $S \cup \{a\}$ is, as such, self–supporting. In AFNs it is required that $S \cup \{a\}$ is coherent, which is a visibly stronger restriction. However, in order to have a chance to be an extension, a set has to be coherent (self–supporting) in the first place. Consequently, we can focus our analysis only on these sets:

Theorem 10.17. *Let $FN = (A, R, N)$ be an AFN and $ES^{FN} = (A', R', E)$ its corresponding EAS obtained through Translation 68. If $a \in A$ is defended by a coherent set $S \subseteq A$ in FN, then it is acceptable w.r.t. $S \cup \{\eta\}$ in ES^{FN}. If $a \in A$ is acceptable w.r.t. a self–supporting set $S' \subseteq A'$ in ES^{FN}, then it is defended by $S \cap A$ in FN.*

With these two theorems at hand we can now clearly state how the AFN and EAS extensions are related. We can observe that the addition of η can generate more extensions in EASs than in AFNs – for example, both $\emptyset$ and $\{\eta\}$ in an EAS will be cast back to $\emptyset$ in the source AFN. Nevertheless, the issue is resolved when we consider complete semantics:

Theorem 10.18. *Let $FN = (A, R, N)$ be an AFN and $ES^{FN} = (A', R', E)$ its corresponding EAS obtained through Translation 68. If a set $S \subseteq A$ is (strongly) coherent in FN, then $S \cup \{\eta\}$ is (strongly) self–supporting in ES^{FN}. If $S \subseteq A$ is (strongly) self–supporting in ES^{FN}, then $S \cap A$ is (strongly) coherent in FN. If $S \subseteq A$ is a σ–extension in FN, where $\sigma \in$ {conflict–free, admissible, complete, preferred, grounded, stable} then $S \cup \{\eta\}$ is a σ–extension in ES^{FN}. If $S \subseteq A$ is a σ–extension of ES^{FN}, then $S \cap A$ is a σ–extension of FN.*

We can now put these results into our system. Given an n–tuple $Z = (z_1, ..., z_n)$, with $set(Z) = \{z_1, ..., z_n\}$ we denote the set of arguments appearing in Z.
Redefinition of Translation 68: *Let Fr^{AFN} be the collection of all AFNs on domain $\mathcal{U}$ and $ABin^{EAS}$ the collection of all attack binary EASs on domain $\mathcal{U} \cup \{\eta\}$. The translation $Tr^{AFN}_{EAS} : Fr^{AFN} \rightarrow ABin^{EAS}$ is defined as $Tr^{AFN}_{EAS}((A, R, N)) = (A', R', E)$ for a framework $(A, R, N) \in Fr^{AFN}$, where $A' = A \cup \{\eta\}$, $R' = \{(\{a\}, b) \mid (a, b) \in R\}$ and $E = \{(\{\eta\}, a) \mid \nexists C \subseteq A \text{ s.t. } CNa\} \cup \{(set(Z'), a) \mid Z' \in Z' \in Z^a_1 \times ... \times Z^a_{n_a}\}$ while $\{Z^a_1 \times ... \times Z^a_{n_a}\}$ is the collection of all sets of A s.t. $Z^a_i Na$.*
Redefinition of Theorem 10.18: *Let $\sigma^{AFN} \in$ {conflict–free, coherent, strongly coherent, admissible, preferred, complete, grounded, stable} be an AFN semantics and $\sigma^{EAS} \in$ {conflict–free, self–supporting, strongly self–supporting, admissible, preferred,*

complete, grounded, stable} *be a similar EAS semantics. Let* SC_σ^{Tr} *be the removal casting functions for* σ *defined as* $SC_\sigma^X(S) = S \cap A$ *for* $X = (A, R, N) \in Fr^{AFN}$ *and* $S \in \sigma^{EAS}(Tr_{EAS}^{AFN}(X))$. *The translation* Tr_{EAS}^{AFN} *is strong* (σ, SC_σ^{Tr}). *It is semantics bijective under the complete, preferred, grounded and stable semantics and the defined casting functions.*

Analysis of Translation 68: Under the conflict–free, (self–supporting) coherent, (strongly self–supporting) strongly coherent, admissible, preferred, complete, grounded, stable and removal casting functions, the translation Tr_{EAS}^{AFN} is:

- full, target–subclass and overlapping
- weakly argument domain altering, argument introducing and induced support introducing
- generic and weakly semantics domain altering
- semi–structural

Our approach is not modular. Under the complete, preferred, grounded and stable semantics and removal casting functions, the translation is faithful. Translation Tr_{EAS}^{AFN} is classified as basic under the listed semantics and casting functions.

Explanation. Any framework can undergo the translation, and thus our approach is full. Due to the fact that we can obtain only attack binary EASs, it is also target–subclass. The way new supporting sets are constructed is not exactly unique; for example, let us consider an argument supported by sets $\{a\}$ and $\{b\}$ in one AFN and by $\{a\}$, $\{b\}$, $\{a, b\}$ in another. In both cases, the new EAS supporting set would be $\{a, b\}$. Thus, our approach is overlapping.

Due to the addition of the special argument to the domain, our translation is weakly argument and semantics domain altering. Clearly, it is also argument introducing (and related support) introducing. We are not entirely sure whether to classify the translation as support removing; a necessary supporter will remain an evidential supporter, it is only the precise structure of the sets that is not preserved. However, this change is unavoidable due to structural differences between the frameworks, similarly as in the SETAF–ADF translation. Thus, we leave this question open.

Due to the amount of handled semantics, the translation is generic. We also classify it as semi–structural due to the addition of evidence and relevant support from it. Unfortunately, the AFN–EAS approach is not modular. Let us consider an argument d necessarily supported by a set $\{a, b\}$ in one framework and $\{b, c\}$ in another. Upon translating, it will be evidentially supported by $\{a\}$, $\{b\}$ and $\{b\}$, $\{c\}$ respectively. Thus, the union of the translated structures will give us $\{a\}$, $\{b\}$ and $\{c\}$. However, if we join the original AFNs, d will be supported both by $\{a, b\}$ and $\{b, c\}$, which are later transformed into $\{a, b\}$, $\{a, c\}$, $\{b\}$ and $\{a, c\}$ (minimal version is $\{a, c\}$ and $\{b\}$). Thus, the new results are not equivalent to the previous ones in any way, and our translation is not modular. ■

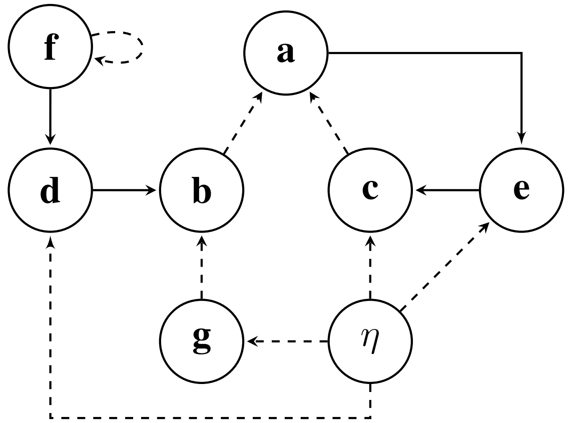


Figure 103: Sample EAS

Example 125. Let us consider the AFN $FN = (\{a, b, c, d, e, f, g\}, \{(a, e), (d, b), (e, c),$ $(f, d)\}, \{(\{b, c\}, a), (\{g\}, b)$,({f},f)}), previously described in Example 122 depicted in Figure 100a. The (minimal) powerful sequences for arguments c, d, e and g are quite straightforward and are respectively (c), (d), (e) and (g). For argument b we have the sequence (g, b) and for a we can create two approaches: (g, b, a) and (c, a). Finally, f has no powerful sequence at all.

The EAS associated with FN is $ES^{FN} = (\{\eta, a, b, c, d, e, f, g\}, \{(\{a\}, e), (\{d\}, b),$ $(\{e\}, c), (\{f\}, d)\}, \{(\{\eta\}, c), (\{\eta\}, d), (\{\eta\}, e), (\{\eta\}, g), (\{b\}, a), (\{c\}, a), (\{f\}, f),$ $(\{g\}, b)\})$ [26], as seen in Figure 103. We can observe that arguments η, c, d, e and g have trivial evidential sequences (η), (η, c), (η, d), (η, e) and (η, g). The sequence for b is (η, g, b), while for a we obtain (η, g, b, a) and (η, c, a). The argument f possesses no evidential sequence. We can therefore observe that these results are in clear correspondence with the powerful sequences of FN.

Based on the sequence analysis, we can observe that the self–supporting sets of ES^{FN} are $\emptyset$, $\{\eta\}$, $\{\eta, c\}$, $\{\eta, d\}$, $\{\eta, e\}$, $\{\eta, g\}$, $\{\eta, a, c\}$, $\{\eta, b, g\}$, $\{\eta, a, b, g\}$ and any of their combinations. The strongly self–supporting (i.e. self–supporting and conflict–free) ones are are $\emptyset$, $\{\eta\}$, $\{\eta, c\}$, $\{\eta, d\}$, $\{\eta, e\}$, $\{\eta, g\}$, $\{\eta, a, c\}$, $\{\eta, b, g\}$, $\{\eta, c, d\}$, $\{\eta, c, g\}$, $\{\eta, d, e\}$, $\{\eta, d, g\}$, $\{\eta, e, g\}$, $\{\eta, a, b, g\}$, $\{\eta, a, c, d\}$, $\{\eta, a, c, g\}$, $\{\eta, b, c, g\}$, $\{\eta, b, e, g\}$, $\{\eta, c, d, g\}$, $\{\eta, d, e, g\}$, $\{\eta, a, b, c, g\}$ and $\{\eta, a, c, d, g\}$. From this, the admissible sets are $\emptyset$, $\{\eta\}$, $\{\eta, d\}$, $\{\eta, g\}$, $\{\eta, a, c\}$, $\{\eta, d, e\}$, $\{\eta, d, g\}$, $\{\eta, a, c, d\}$, $\{\eta, a, c, g\}$, $\{\eta, d, e, g\}$ and $\{\eta, a, c, d, g\}$. We can observe that in all of these cases, by removing η from the sets we can obtain the coherent, strongly coherent and admissible sets of FN. Moreover, only one set – namely, $\emptyset$ – can be obtained from two extensions ($\emptyset$ and $\{\eta\}$). All other sets can be produced only from a single extension in ES^{FN}. From the presented complete extensions, $\{\eta, d, g\}$, $\{\eta, d, e, g\}$ and $\{\eta, a, c, d, g\}$ qualify as complete, with the first set

[26] Depending on whether the optimized construction was used or not, the pair $(\{b, c\}, a)$ would also appear in the support relation. Nevertheless, it does not affect the extensions of the framework since the two options are connected through the minimal normal form.

being grounded and the other two preferred and stable in ES^{FN}. Again, by removing η, we can exactly retrieve the extensions of FN. The only difference is that when we start considering complete extensions, the relation finally becomes one–to–one.

10.6 AFN as ADF

We will now present possible methods for translating AFNs into ADFs. Bearing in mind the issues we have described in Section 2.3.9, we will now classify AFNs w.r.t. the *consistency* criterion. While normal consistency ensures that no argument will receive a falsum acceptance condition after the translation, its strong version prevents loss of acyclic pd–evaluations.

The condition of an argument in an AFN–obtained ADF can be seen as consisting of two parts. The first one corresponds to the attackers of the argument and is basically the same as in the AF–ADF case (see Section 5.7). In other words, given an argument $a \in A$ and $X = \{x_1, .., x_n\}$ the set of arguments attacking a, the condition is simply $att_a = \neg x_1 \wedge ... \wedge \neg x_n$. We abbreviate this construction with $\bigwedge \neg X$.

The other part of the condition concerns support. Independently of validity, an argument is considered sufficiently supported in an AFN by a given set, if this set has at least one element in common with every supporting set of the argument in question. For example, if a is supported by sets $\{b, c\}$ and $\{d, e\}$, then if we want to accept a, either b and d, b and e, c and d or c and e need to be present. This can be equivalently expressed with a condition $(b \vee c) \wedge (d \vee e)$. Therefore, given an argument a and a collection of sets supporting it $Z = \{Z_1, ..., Z_n\}$, we create a formula $sup_a = \bigvee Z_1 \wedge ... \wedge \bigvee Z_n$, where $\bigvee Z_i$ is a disjunction of all arguments in Z_i.

10.6.1 Strongly Consistent AFNs

Similarly as in the EAF(C)–ADF case (see Section 8.6), we start with the translation which assumes that our source frameworks are strongly consistent:

Translation 69. *Let $FN = (A, R, N)$ be a strongly consistent AFN. The corresponding ADF $D^{FN} = (A, L, C)$ is created as follows:*

1. *for every two arguments a, b, if aRb or there exists a set $Z \subseteq A$ containing a s.t. ZNb, add (a, b) to L, and*

2. *for every argument a, the functional acceptance condition is:*

   - *every $P' \subseteq par(a)$ is mapped to out iff $\exists p \in P'$ s.t. pRa or $\exists Z \subseteq par(a)$ s.t. ZNa and $Z \cap P' = \emptyset$, and*

   - *all other subsets of P are in.*

3. *for every argument a, the propositional acceptance condition is:*

- *the attack part is $att_a = (\neg t_1 \wedge ... \wedge \neg t_n)$, where $t_1, ..., t_n$ are all elements of A s.t. $t_i R a$,*
- *the support part is $sup_a = (\bigvee Z_1 \wedge ... \wedge \bigvee Z_m)$, where $Z_1, ..., Z_m$ are all subsets of A s.t. $Z_i N a$, and*
- *the acceptance condition is $C_a = att_a \wedge sup_a$.*

In case an argument is initial, its condition is simply $\top$.

The only thing we can say with certainty is that the produced ADF is a BADF in cleansed form. All other possible properties depend on the source AFN.

Theorem 10.19. *Let $FN = (A, R, N)$ be a strongly consistent AFN and $D^{FN} = (A, L, C)$ its corresponding ADF obtained through Translation 69. D^{FN} is a BADF. It is also in cleansed form. If FN is in minimal form, then D^{FN} is redundancy–free. If FN is weakly valid, then so is D^{FN}. If it is minimal and relation valid, then D^{FN} is relation valid. If FN is strongly valid, then D^{FN} is an AADF$^+$. If it is in addition minimal, then D^{FN} is strongly valid.*

Following the analysis we presented in Section 5.7 while studying the binary attack in AFs, we can observe that any decisively in interpretation v for a (consistent) argument a will map all attackers of a from FN (i.e. all arguments in att_a) to $\mathbf{f}$. Moreover, since $v^{\mathbf{t}}$ needs to satisfy the condition of a, it also has to be the case that for every $Z \subseteq A$ s.t. ZNa, $v^{\mathbf{t}} \cap Z \neq \emptyset$. In other words, $\exists z \in Z$ s.t. $v(z) = \mathbf{t}$ (as long as we do not impose minimality on the interpretations). However, if a is consistent but not strongly consistent, it means there exists a supporting set Z containing an argument z' s.t. $z'Ra$. Consequently, for such z' it has to be the case that $v(z') = \mathbf{f}$. Thus, it will never be assigned $\mathbf{t}$ by any decisively translation, and has no chance to be used in the pd–sequence of an acyclic evaluation, despite its appearance in a powerful sequence in the source AFN. We use strong consistency in order to prevent such situations. Thus, given a minimal decisively in interpretation v_{min} for a, all attackers of a will be mapped to $\mathbf{f}$ by v_{min}, and the ones mapped to $\mathbf{t}$ will correspond to minimal sets of arguments $B \subseteq A$ s.t. for every set Z supporting a, $B \cap Z \neq \emptyset$. Consequently, since no member of a supporting set is also an attacker of a, it cannot be the case that an argument contained in a supporting set is mapped to $\mathbf{f}$ by v_{min}.

The AFN semantics are built around the notion of coherence, which expresses the requirement that an argument can be (support–wise) derived in an acyclic manner, and only such derivations need to be taken into account when we consider defense. Thus, not surprisingly, it is the aa–family of ADF semantics that will be related to AFN semantics. We start our analysis by drawing the connection between coherent and pd–acyclic, strongly coherent and pd–acyclic conflict–free extensions of both frameworks:

Lemma 10.20. *Let $FN = (A, R, N)$ be a strongly consistent AFN, $D^{FN} = (A, L, C)$ its corresponding ADF obtained through Translation 69 and $E \subseteq A$ a set of arguments. For a given powerful sequence for an argument $e \in E$ we can construct a corresponding acyclic pd–evaluation and vice versa. E is coherent in FN iff it is pd–acyclic in D^{FN}.*

Lemma 10.21. *Let $FN = (A, R, N)$ be a strongly consistent AFN, $D^{FN} = (A, L, C)$ its corresponding ADF obtained through Translation 69. A set of arguments $E \subseteq A$ is strongly coherent in FN iff it is a pd–acyclic conflict–free extension of D^{FN}.*

We can also draw the connection between the discarded sets of both frameworks, which as a result gives us also a connection between defense in AFNs and being decisively in w.r.t. a given interpretation in ADFs. However, please note that we will use the E^{att} set, not the deactivated set in case of AFNs (see Section 2.2.2.2 for analysis). The original version is too weak for establishing an exact correspondence.

Lemma 10.22. *Let $FN = (A, R, N)$ be a strongly consistent AFN, $D^{FN} = (A, L, C)$ its corresponding ADF obtained through Translation 69. Let $E \subseteq A$ be strongly coherent in FN and thus pd–acyclic conflict–free in D^{FN}. Then E^{att} coincides with the acyclic discarded set of E.*

Theorem 10.23. *Let $FN = (A, R, N)$ be a strongly consistent AFN and $D^{FN} = (A, L, C)$ its corresponding ADF obtained through Translation 69. Let $E \subseteq A$ be strongly coherent in FN and thus pd–acyclic conflict–free in D^{FN}. Then E defends an argument $a \in A$ in FN iff this argument is decisively in w.r.t. v_E^a in D^{FN}.*

With these theorems and lemmas at hand, we can finally show the correspondence between the source and target extensions:

Theorem 10.24. *Let $FN = (A, R, N)$ be a strongly consistent AFN, $D^{FN} = (A, L, C)$ its corresponding ADF obtained through Translation 69. A set of arguments $E \subseteq A$ is coherent in FN iff it is pd–acyclic in D^{FN}. E is strongly coherent in FN iff it is pd–acyclic conflict–free in D^{FN}. E is a σ–extension of FN, where $\sigma \in \{$admissible, complete, preferred$\}$ iff it is an aa–σ–extension of D^{FN}. E is stable in FN iff it is stable in D^{FN}. E is grounded in FN iff it is acyclic grounded in D^{FN}.*

Let us now redefine our results in accordance with our system and study the properties of our translation. Please note that we will use the BADF subclass as our codomain, it is not the most accurate description. For example, not every EAF–style ADF can be produced by an AFN. We leave describing a more fitting subclass for future work.
Redefinition of Translation 69: *Let $SCons^{AFN}$ be the collection of all strongly consistent AFNs and $BADF$ the collection of all bipolar ADFs, both on domain $\mathcal{U}$. The translation sc-$Tr_{ADF}^{AFN} : SCons^{AFN} \rightarrow BADF$ is defined as sc-$Tr_{ADF}^{AFN}((A, R, N)) = (A, L, C)$, where $L = \{(x, y) \mid (x, y) \in R$ or $\exists X \subseteq A, x \in X$ s.t. $(X, y) \in N\}$, and $C = \{C_a \mid a \in A\}$, where every C_a is defined as a) $C_a(P') = out$ for $P' \subseteq par(a)$ iff $\exists x \in P'$ s.t. xRa or $\exists Z \subseteq par(a)$ s.t. ZNa and $Z \cap P' = \emptyset$, otherwise, $C_a(P') = in$; or b) $C_a = (\neg t_1 \wedge ... \wedge \neg t_n) \wedge (\bigvee Z_1 \wedge ... \wedge \bigvee Z_m)$ where $t_1, ..., t_n$ are all elements of A s.t. $t_i Ra$ and $Z_1, ..., Z_m$ are all subsets of A s.t. $Z_i Na$; in case both collections are empty, $C_a = \top$.*

Redefinition of Theorem 10.24: *Let $\sigma^{AFN} \in \{$coherent, strongly coherent, admissible, complete, preferred, grounded, stable$\}$ be an AFN semantics and $\sigma^{ADF} \in \{$pd–acyclic, pd–acyclic conflict–free, aa–admissible, aa–complete, aa–preferred, acyclic grounded, stable$\}$ a similar ADF semantics. Let SC_{σ}^{Tr} be the identity casting functions for σ. The translation sc-Tr_{ADF}^{AFN} is strong and semantics bijective under $(\sigma, SC_{\sigma}^{Tr})$.*
Analysis of Translation 69: Under the (pd–acyclic) coherent, (pd–acyclic conflict–free) strongly coherent, (aa–) admissible, (aa–) complete, (aa–) preferred, (acyclic) grounded and stable semantics and the identity casting functions, the translation $sc\text{-}Tr_{ADF}^{AFN}$ is:

- source–subclass, target–subclass and overlapping
- argument domain preserving and structure preserving
- generic, semantics domain preserving and exact
- structural and $\otimes$–modular

The translation is not $\oplus$–modular. The translation $sc\text{-}Tr_{ADF}^{AFN}$ is classified as basic under the listed semantics and casting functions.

Explanation. We only consider strongly valid AFNs in our translation and naturally, BADFs do not account for all possible ADFs. Consequently, our approach is full and target–subclass. Unfortunately, it is also overlapping, for similar reasons as in Translation 31.

Clearly, $sc\text{-}Tr_{ADF}^{AFN}$ is argument and semantics domain preserving and generic. We can observe that no arguments are added or removed during the translation. Moreover, as seen from the definition of L, all connections are preserved between the arguments. The links that are considered attacking remain attacking and those that were supporting are still supporting (see Theorem 10.19), even though some are redundant. Therefore, despite the fact that the translation is overlapping, it is structure preserving, and this state of affairs is unavoidable due to the differences between ADFs and AFNs (see Section 2.3.9).

In order to show that the translation is not $\oplus$–modular, we can consider AF–style AFNs and repeat the analysis of Translation 23. Let us now focus on $\otimes$–modularity. Let $FN_1 = (A_1, R_1, N_1)$ and $FN_2 = (A_2, R_2, N_2)$ be two AFNs and $FN_3 = FN_1 \cup FN_2 = (A_1 \cup A_2, R_1 \cup R_2, N_1 \cup N_2)$. With $D_1 = (A_1, L_1, C_1)$, $D_2 = (A_2, L_2, C_2)$ and $D_3 = (A_3, L_3, C_3)$ we will denote the corresponding ADFs. It is easy to show that $A_3 = A_1 \cup A_2$ and $L_3 = L_1 \cup L_2$. What needs to be shown is that for every argument $a \in A_1 \cap A_2$, $C_1^a \otimes C_2^a = C_3^a$ – if a given argument is not present in either of the frameworks, then its condition in C_3^a is the same as in the structure it came from. Let X_1^a, X_2^a and X_3^a be the collections of attackers of a in FN_1, FN_2 and FN_3 respectively. Clearly, $X_1^a \cup X_2^a = X_3^a$. We can observe that $\bigwedge \neg X_1^a \wedge \bigwedge \neg X_2^a = \bigwedge \neg (X_1^a \cup X_2^a) = \bigwedge \neg X_3^a$; thus, the att_a part of the condition is the same in $D_1 \otimes D_2$ as in D_3. Let now $\{Z_1^a, ..., Z_m^a\}$, $\{V_1^a, ..., V_n^a\}$ and $\{Y_1^a, ..., Y_l^a\}$ be the collections of sets supporting a in FN_1, FN_2 and FN_3 respectively. Since $FN_3 = FN_1 \cup FN_2$, we can observe that $\{Z_1^a, ..., Z_m^a\} \cup \{V_1^a, ..., V_n^a\} = \{Y_1^a, ..., Y_l^a\}$. Therefore,

$(\bigvee Z_1^a \wedge .... \wedge \bigvee Z_m^a) \wedge (\bigvee V_1^a \wedge .... \wedge \bigvee V_n^a)$ is equivalent to $\bigvee Y_1^a \wedge .... \wedge \bigvee Y_l^a$. Therefore, the support parts are also equivalent in $D_1 \otimes D_2$ and D_3. Consequently, our translation can be shown to be $\otimes$–modular. ■

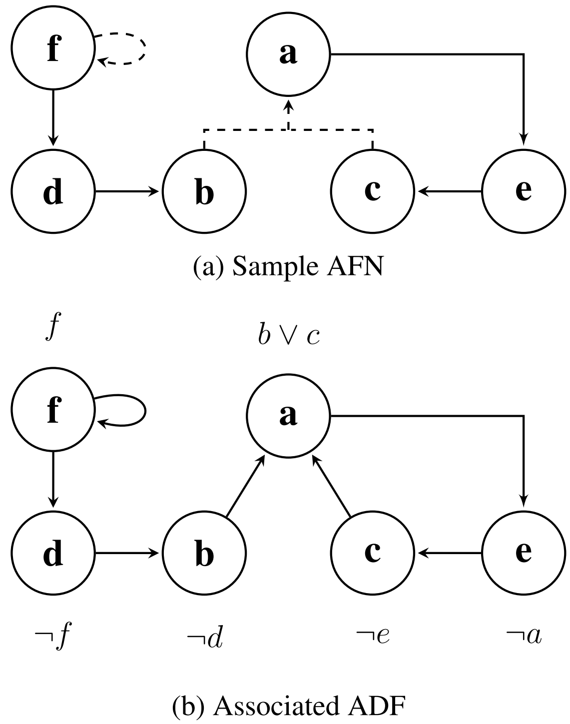


Figure 104: Sample AFN and its associated ADF

Example 126. Let us come back to the AFN $FN = (\{a, b, c, d, e, f\}, \{(a, e), (d, b), (e, c), (f, d)\}, \{(\{b, c\}, a), (\{f\}, f)\})$ from Example 16, now depicted in Figure 104a. We can observe argument f possesses no powerful sequence at all. For arguments b, c, d and e, we can create simple (minimal) sequences consisting of the arguments themselves, i.e. (b), (c), (d) and (e). Finally, argument a has two minimal powerful sequences, namely (b, a) and (c, a).

The ADF associated with our AFN is $D^{FN} = (\{a, b, c, d, e, f\}, \{C_a = b \vee c, C_b = \neg d, C_c = \neg e, C_d = \neg f, C_e = \neg a, C_f = f\})$. In order to describe the extensions, we will first list the relevant interpretations and evaluations of D^{FN}. The minimal decisively in interpretations for our arguments are as follows; $v_a^1 = \{b : \mathbf{t}\}$, $v_a^2 = \{c : \mathbf{t}\}$, $v_b = \{d : \mathbf{f}\}$, $v_c = \{e : \mathbf{f}\}$, $v_d = \{f : \mathbf{f}\}$, $v_e = \{a : \mathbf{f}\}$ and $v_f = \{f : \mathbf{t}\}$. We can observe that with the exception of a and f, every argument will possess a single minimal acyclic evaluation; we can create $((b), \{d\})$ for b, $((c), \{e\})$ for c, $((d), \{f\})$ for d and $((e), \{a\})$ for e. Argument f possesses no acyclic evaluation at all, while a has two: $((b, a), \{d\})$ and $((c, a), \{e\})$.

The pd–acyclic conflict–free extensions of our framework are $E_1 = \emptyset$, $E_2 = \{b\}$, $E_3 = \{c\}$, $E_4 = \{d\}$, $E_5 = \{e\}$, $E_6 = \{a, b\}$, $E_7 = \{a, c\}$, $E_8 = \{b, c\}$, $E_9 = \{b, e\}$,

$E_{10} = \{c, d\}$, $E_{11} = \{d, e\}$, $E_{12} = \{a, b, c\}$ and $E_{13} = \{a, c, d\}$. We can observe they correspond exactly to the strongly coherent sets of FN. Their associated acyclic discarded sets are now $E_1^{a+} = E_2^{a+} = E_3^{a+} = E_8^{a+} = \{f\}$, $E_4^{a+} = E_{10}^{a+} = \{b, f\}$, $E_5^{a+} = E_9^{a+} = \{c, f\}$, $E_6^{a+} = E_7^{a+} = E_{12}^{a+} = \{e, f\}$, $E_{11}^{a+} = \{b, c, f\}$ and $E_{13}^{a+} = \{b, e, f\}$.

With this at hand we can show that E_1, E_4, E_7, E_{11} and E_{13} are the only aa–admissible extensions of our framework. For every other set, we can find an argument that is not sufficiently "defended". For the extensions E_2, E_6, E_8, E_9 and E_{12}, it is the argument b – as visible from its evaluation, d would need to be included in the discarded sets of these extensions, but is not. In the case of extensions E_3 and E_{10}, the issue lies with c and the fact that argument e is not included neither in E_3^{a+} nor in E_{10}^{a+}. Finally, we have the set E_5 and argument e. Unfortunately, E_5 blocks only one of the evaluations of a, not all of them, thus making it impossible for e to be protected from it. We can observe that the produced aa–admissible sets are the same as the admissible extensions of FN.

From the aa–admissible extensions E_1, E_4, E_7, E_{11} and E_{13}, the aa–complete ones are E_4, E_{11} and E_{13}. In case of extensions E_1 and E_7, we can observe that they do not contain the argument d, despite the fact it is decisively in w.r.t. their acyclic range interpretations. Again, these answers agree with the extensions produced by FN. From here, we can easily identify E_4 as the acyclic grounded extensions and E_{11} and E_{13} as the aa–preferred ones, which are the desired extensions. D^{FN} has three models; $\{d, e\}$ (i.e. E_{11}), $\{a, c, d\}$ (i.e. E_{13}), and $\{a, b, c, f\}$. Only the first two are pd–acyclic conflict–free and are thus the stable extensions of D^{FN}. The same sets are stable in FN. Therefore, we retrieve all and only the desired extensions of FN in a number of semantics.

10.6.2 General AFNs

In order to translate AFNs that do not necessarily meet the consistency restrictions, we will make use of consistency normal form translations. We first bring a given AFN into the consistency form, and then proceed with the previously described approach for strongly consistent AFNs. However, we have two possible normal form translations for AFNs – Translations 13 and 15. Consequently, the way the source AFN extensions are related to the target ADF ones depends on which method we use.

Translation 70. *Let $FN = (A, R, N)$ be an AFN. The corresponding ADF $D^{FN} = (A', L, C)$ is created as follows:*

1. *we obtain the strongly consistent AFN $FN^{sc} = (A', R, N')$ corresponding to FN by Translation 13 or 15, and*

2. *D^{FN} results from transforming FN^{sc} into ADF by Translation 69.*

Due to Theorems 4.51 and 4.57, the normal form theorem (Theorem 10.19) holds in this case as well, independently of the used consistency form. Moreover, by joining Theorems 4.50 and 10.24, we can connect the extensions of the source AFN and target ADF:

Theorem 10.25. *Let $FN = (A, R, N)$ be an AFN and $D^{FN} = (A', L, C)$ its corresponding ADF obtained by Translations 13 and 70. Let $E \subseteq A$, $E' \subseteq A'$ be sets of arguments and E^b the (possibly empty) set of bypass arguments generated by E in A'. The following holds:*

- *if E is coherent in FN, then $E \cup E^b$ is pd–acyclic in D^{FN}.*
- *if E is strongly coherent in FN, then $E \cup E^b$ is pd–acyclic conflict–free in D^{FN}.*
- *if E is a σ–extension of FN, where $\sigma \in \{$admissible, preferred, complete$\}$, then $E \cup E^b$ is an aa–σ–extension of D^{FN}.*
- *if E is grounded in FN, then $E \cup E^b$ is acyclic grounded in D^{FN}.*
- *if E is stable in FN, then $E \cup E^b$ is stable in D^{FN}.*
- *if E is pd–acyclic in D^{FN}, then $E' \cap A$ is coherent in FN.*
- *if E is pd–acyclic conflict–free in D^{FN}, then $E' \cap A$ is strongly coherent in FN.*
- *if E' is an aa–σ–extension of D^{FN}, then $E' \cap A$ is a σ–extension of FN.*
- *if E is acyclic grounded in D^{FN}, then $E' \cap A$ is grounded in FN.*
- *if E is stable in D^{FN}, then $E' \cap A$ is stable in FN.*

And from Theorems 4.56 and 10.24, we get the following:

Theorem 10.26. *Let $FN = (A, R, N)$ be an AFN and $D^{FN} = (A', L, C)$ its corresponding ADF obtained by Translations 15 and 70. Let E^b the (possibly empty) set of bypass arguments generated by a set $E \subseteq A$ in A'. If a set of arguments E is coherent in FN, then $E \cup E^b$ is pd–acyclic in D^{FN}. If $E' \subseteq A'$ is pd–acyclic in D^{FN}, then $E' \cap A$ is coherent in FN. $E \subseteq A$ is strongly coherent in FN iff it is pd–acyclic conflict–free in D^{FN}. E is a σ–extension of FN, where $\sigma \in \{$admissible, complete, preferred$\}$ iff it is an aa–σ–extension of D^{FN}. E is grounded in FN iff it is acyclic grounded in D^{FN}. Every stable extension E of FN is stable in D^{FN} but not vice versa.*

We can now put the translation and the theorems into our system. The properties of the resulting approaches will be basically a result of the attributes of the used translations. We choose to classify the merged approaches as semi–structural due to consistency issues.

Redefinition of Translation 70: *Let Fr^{AFN} be the collection of all AFNs on domain $\mathcal{U}$ and $BADF$ the collection of all bipolar ADFs on domain $\mathcal{U} \cup \mathcal{U}^b$. The translation b-$Tr^{AFN}_{ADF}$: $Fr^{AFN} \rightarrow BADF$ is defined as b-$Tr^{AFN}_{ADF}((A, R, N))$ = sc-Tr^{AFN}_{ADF}(bc-$Tr^{AFN}((A, R, N))$). The translation sa-Tr^{AFN}_{ADF} : $Fr^{AFN} \rightarrow BADF$ is defined as sa-$Tr^{AFN}_{ADF}((A, R, N))$ = sc-Tr^{AFN}_{ADF}(sa-$Tr^{AFN}((A, R, N))$).*

Redefinition of Theorem 10.25: *Let $\sigma^{AFN} \in \{$coherent, strongly coherent, admissible, complete, preferred, grounded, stable$\}$ be an AFN semantics and $\sigma^{ADF} \in \{$pd–acyclic,* *pd–acyclic conflict–free, aa–admissible, aa–complete, aa–preferred, acyclic grounded, stable$\}$ a similar ADF semantics. Let SC_σ^{Tr} be the removal casting functions for σ defined as $SC_\sigma^X(E) = E \cap A$ for $X = (A, R, N) \in Fr^{AFN}$ and $E \in \sigma^{ADF}(b\text{-}Tr_{ADF}^{AFN}(X))$. The translation $b\text{-}Tr_{ADF}^{AFN}$ is strong under (σ, SC_σ^{Tr}). It is semantics bijective under the (aa–) complete, (aa–) preferred, (acyclic) grounded and stable semantics and the defined casting functions.*

Redefinition of Theorem 10.26: *Let $\sigma^{AFN} \in \{$coherent, strongly coherent, admissible, complete, preferred, grounded$\}$ be an AFN semantics and $\sigma^{ADF} \in \{$pd–acyclic, pd–acyclic conflict–free, aa–admissible, aa–complete, aa–preferred, acyclic grounded$\}$ a similar ADF semantics. Let SC_σ^{Tr} be the identity casting functions for σ. The translation $sa\text{-}Tr_{ADF}^{AFN}$ is strong under (σ, SC_σ^{Tr}). With the exception of coherent–pd–acyclic semantics, it is also semantics bijective. It is $\subseteq$–weak under the stable semantics and defined casting functions.*

Analysis of Translation 70 with Translation 13 as intermediary: Under the (pd–acyclic) coherent, (pd–acyclic conflict–free) strongly coherent, (aa–) admissible, (aa–) complete, (aa–) preferred, (acyclic) grounded and stable semantics and the removal casting functions, the translation $b\text{-}Tr_{ADF}^{AFN}$ is:

- full, target–subclass and overlapping
- weakly argument domain altering, argument introducing, induced relation introducing, relation removing
- generic and weakly semantics domain altering
- semi–structural

The translation is neither $\otimes$ nor $\oplus$–modular. It is faithful under the (aa–) complete, (aa–) preferred, (acyclic) grounded and stable semantics and the removal casting functions. The translation $b\text{-}Tr_{ADF}^{AFN}$ is classified as basic under the listed semantics and casting functions.

Analysis of Translation 70 with Translation 15 as intermediary: Under the (pd–acyclic) coherent, (pd–acyclic conflict–free) strongly coherent, (aa–) admissible, (aa–) complete, (aa–) preferred, (acyclic) grounded and stable semantics and the identity casting functions, the translation $sa\text{-}Tr_{ADF}^{AFN}$ is:

- full, target–subclass and overlapping
- weakly argument domain altering, argument introducing, induced relation introducing, relation removing
- generic and semantics domain preserving
- semi–structural

The translation is neither $\otimes$ nor $\oplus$–modular. It is exact under the (pd–acyclic conflict–free) strongly coherent, (aa–) admissible, (aa–) complete, (aa–) preferred and (acyclic) grounded semantics and their identity casting functions. The translation $sa\text{-}Tr_{ADF}^{AFN}$ is classified as basic under the listed semantics and casting functions.

10.6.3 Improvements

In this section we have presented in total three approaches for translating AFNs into ADFs; one for strongly consistent frameworks, and two for general ones. In doing so, we have constructed an exact and full translation for every type of semantics with the exception of stable. However, since there exists a full and exact translation for stable semantics from AFNs to SETAFs and SETAFs to ADFs, we believe that a suitable direct approach can be devised. Unfortunately, this task needs to be left for future work.

10.7 AFN as Other Frameworks

The only framework that we have not discussed in this section is AFRA. However, we do not believe there is any gain in transforming necessary support into recursive attack, particularly due to the fact that if we try to translate recursive attack into a positive relation, then abstract support is a much more natural target. Thus, we propose to use a chained translation in order to convert AFNs into AFRAs.

10.8 Summary

In this section we have presented a number of translations from AFNs to other frameworks. We could have observed that the best results w.r.t. semantics we have obtained in case of two, unfortunately source–subclass, AFN–BAF and AFN–ADF translations. While the first one permitted only support binary and strongly valid frameworks, the other assumed that the source AFNs are strongly consistent. These are also the only generic and modular methods we have established. For full approaches, the next in line are the AFN–ADF translation (with self–attacker consistency form) and the attack propagation AFN–SETAF transformation, followed by the AFN–EAS and AFN–ADF (with bypass consistency form). The results are visible in Table 13.

Although our results are quite interesting, the analysis in Section 10.1.1 shows that there are still possibilities for improvement. In particular, we know that it is possible to create an exact translation from AFNs to AFs under the admissible and preferred semantics. We suspect it to be a combination of attack propagation and defender approaches; nevertheless, we are yet to create a satisfactory method. Therefore, we hope that the our results can be improved in the future.

Table 13: Translations from AFNs to other frameworks

Properties		AF		SETAF			BAF	EAS	ADF		
Translation		61	62	63	65	66	67	68	69	70 with 13	70 with 15
Strength	cf		exact	⊇-weak	strong	strong	exact	strong			
	coh						exact	strong	exact	strong	strong
	str coh	strong		⊆-weak	⊆-weak	⊆-weak	exact	strong	exact	strong	exact
	adm	strong		⊆-weak	strong	strong	exact	strong	exact	strong	exact
	comp	strong		exact	faithful	faithful	exact	faithful	exact	faithful	exact
	pref	strong		exact	faithful	faithful	exact	faithful	exact	faithful	exact
	grd	strong		exact	faithful	faithful	exact	faithful	exact	faithful	exact
	stb	strong		exact	faithful	faithful	exact	faithful	exact	faithful	⊆-weak
Functional		full	full	full	full	source–subclass	source–subclass	full	source–subclass	full	full
		target–subclass	surjective	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass
		overlapping	injective	overlapping	overlapping	overlapping	injective	overlapping	overlapping	overlapping	overlapping
Syntactical		argument domain altering	argument domain preserving	argument domain preserving	weakly argument domain altering	weakly argument domain altering	argument domain preserving	weakly argument domain altering	argument domain preserving	weakly argument domain altering	weakly argument domain altering
		argument removing	support removing	argument removing	argument introducing	argument introducing	structure preserving	argument introducing	structure preserving	argument introducing	argument introducing
		argument introducing		induced attack removing	induced attack introducing	induced attack introducing		induced support introducing		induced relation introducing	induced relation introducing
		(induced) attack removing		attack introducing	support removing	support removing				relation removing	relation removing
		support removing		support removing							
Semantical		generic	specialized	generic	generic	generic	generic	generic	generic	generic	generic
		semantics domain altering	semantics domain preserving	semantics domain preserving	weakly semantics domain altering	weakly semantics domain altering	semantics domain preserving	weakly semantics domain altering	semantics domain preserving	weakly semantics domain altering	semantics domain preserving
Computational		semantical	structural	semantical	semantical	structural	structural	semi–structural	structural	semi–structural	semi–structural
			modular				modular		⊗–modular		

11 Translating EASs

In this section we will show how evidential frameworks can be translated to AFs, SETAFs, BAFs, AFNs and ADFs. Many of our results mimic those obtained for AFNs in Section 10, which is unsurprising considering the connections between the evidential and necessary supports. Nevertheless, due to the fact that one framework uses binary and the other group attack, there are certain differences in our results. First of all, unlike AFNs, EASs cannot be exactly translated to Dung's framework. Whether the same holds for SETAFs, we are not yet able to say. Moreover, in order to translate EASs to SETAFs, we also consider a coalition approach in addition to the defender and attack propagation ones. This allows us to decrease the amount of auxiliary arguments created during the transformation with regard to the EAS–AF coalition approach. The difference in the used attacks also leads to two EAS–AFN translations, one that is source–subclass and permits only attack binary EASs and the other that works with any type of a framework. Finally, we again consider three translations to ADFs, one with consistency restrictions and two that use bypass or self–attacker normal forms as intermediary.

11.1 EAS as AF

The translation of EASs into AFs has been presented in [73] and it follows the coalition approach. A single AF argument will correspond to a set of EAS arguments s.t. its elements are related one to another by the support relation and are able to trace back to the evidence. Consequently, they will form self–supporting sets. The algorithm we will recall is the most general one. Its analysis and possible optimizations, particularly w.r.t. the amount of created AF arguments, can be found in [73]. Although some of the EAS definitions were later corrected in [78], the translation still holds. The only difference is that in the new approach, the empty set is also a self–supporting set, and there is no need for an empty coalition argument.

Translation 71. *Let $ES = (A, R, E)$ be an EAS. Its corresponding AF $F^{ES} = (A', R')$ is created as follows:*

- *for every nonempty self–supporting set $S \subseteq A$, add S as an argument in A', and*
- *for every $(X, y) \in R$, add (X', Y') to R', where $X', Y' \in A'$, $X \subseteq X'$ and $y \in Y'$.*

It is easy to see that arguments that are not valid in ES (i.e. are not e–supported at all) will not appear in F^{ES}. Although this does not affect the behavior of the semantics, we can see that a certain portion of data is lost. Moreover, it is important to note that unlike in AFNs, we cannot use minimal self–supporting sets in our translation. This is due to the fact that now we are working with group, not binary support. Consequently, the coalition arguments need to be formed in such a way that not only elements required for support, but also for attacks, are present:

Example 127. Let $ES = (\{a, b, c, \eta\}, \{(\{a, b\}, c)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, c)\})$ be the simple EAS depicted in Figure 105. It has in total nine self–supporting sets: $\emptyset$, $\{\eta\}$, $\{\eta, a\}$, $\{\eta, b\}$, $\{\eta, c\}$, $\{\eta, a, b\}$, $\{\eta, a, c\}$, $\{\eta, b, c\}$ and $\{\eta, a, b, c\}$. If we were to consider only those that are minimal for a given argument, we would be left with four – $\{\eta\}$, $\{\eta, a\}$, $\{\eta, b\}$ and $\{\eta, c\}$. If we treat these sets as the basis for constructing the coalition AF, our framework would not contain any attacks, despite the fact that a and b jointly attack c. Thus, $\{\{\eta, c\}\}$ would emerge as an admissible extension of the target Dung's framework, even though its corresponding set $\{\eta, c\}$ is not admissible in ES.

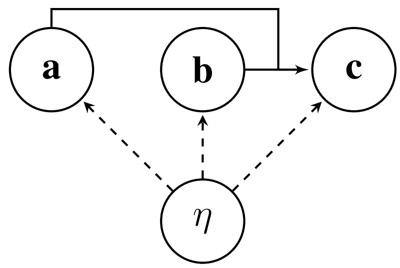


Figure 105: Sample EAS

Please observe that a possible optimization, alternative to the one from [78], is to consider only minimal self–supporting sets for given arguments and minimal e–supported attacks (not targeted at invalid arguments) as new elements of the target AF.

Let us now move on to the semantics analysis; we reformulate the original results from [73] and include additional remarks concerning conflict–freeness. Unfortunately, conflict–freeness is preserved only one–way, similarly as in the basic SETAF–AF approach (see Theorem 6.1).

Theorem 11.1. *Let $ES = (A, R, E)$ be an EAS and $F^{ES} = (A', R')$ its corresponding AF obtained by Translation 71. If $S \subseteq A$ is strongly self–supporting in ES, then there exists a conflict–free extension $S' \subseteq A'$ of F^{ES} s.t. $S = \bigcup S'$. If S is a σ–extension of ES, where $\sigma \in \{$admissible, complete, preferred, grounded stable$\}$ [27], then there exists a σ–extension $S' \subseteq A'$ of F^{ES} s.t. $S = \bigcup S'$.*

If $S' \subseteq A'$ is conflict–free in F^{ES}, it is self–supporting in ES, but not necessarily conflict–free. If $S' \subseteq A'$ is a σ–extension of F^{ES}, then $\bigcup S'$ is a σ–extension of ES.

We can now put the results into our system and analyze the translation. The analysis is the same as in case of Translation 61 and thus we will omit further explanations.

Redefinition of Translation 71: *Let Fr^{EAS} be the collection of all EASs based on domain $\mathcal{U}$ and Fr^{AF} the collection of all AFs based on the domain $2^{\mathcal{U}}$. The translation $Tr^{EAS}_{AF} : Fr^{EAS} \rightarrow Fr^{AF}$ is defined as $Tr^{EAS}_{AF}((A, R, E)) = (A', R')$ for a framework $(A, R, E) \in Fr^{EAS}$, where $A' = \{S \mid S \subseteq A$ is self–supporting in $(A, R, R)\}$ and $R' = \{(X, Y) \mid \exists X' \subseteq X, y \in Y\ s.t. (X', y) \in R\}$.*

Redefinition of Theorem 11.1: *Let $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the union casting functions for σ. The translation Tr^{EAS}_{AF} is strong under $(\sigma, SC^{Tr}_{\sigma})$ and $\subseteq$–weak under the strongly self–supporting – conflict–free semantics and union casting functions. It is semantics bijective under the complete, preferred, grounded and stable semantics and union casting functions.*

[27] Although formally complete semantics for EASs is defined only later in [78], the results still hold.

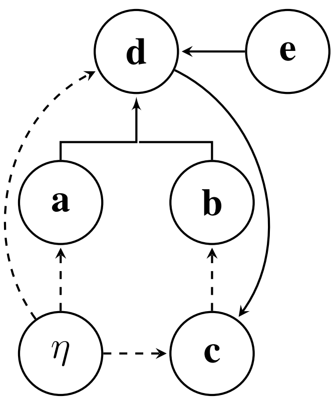
Analysis of Translation 71: Under the (conflict–free) strongly self–supporting, admissible, preferred, complete, grounded and stable semantics and union casting functions, the translation Tr_{AF}^{EAS} is:

- full, target–subclass and overlapping
- argument domain altering, argument removing, argument introducing, attack removing, support relation removing, possibly attack and support relation introducing
- generic and semantics domain altering
- semantical

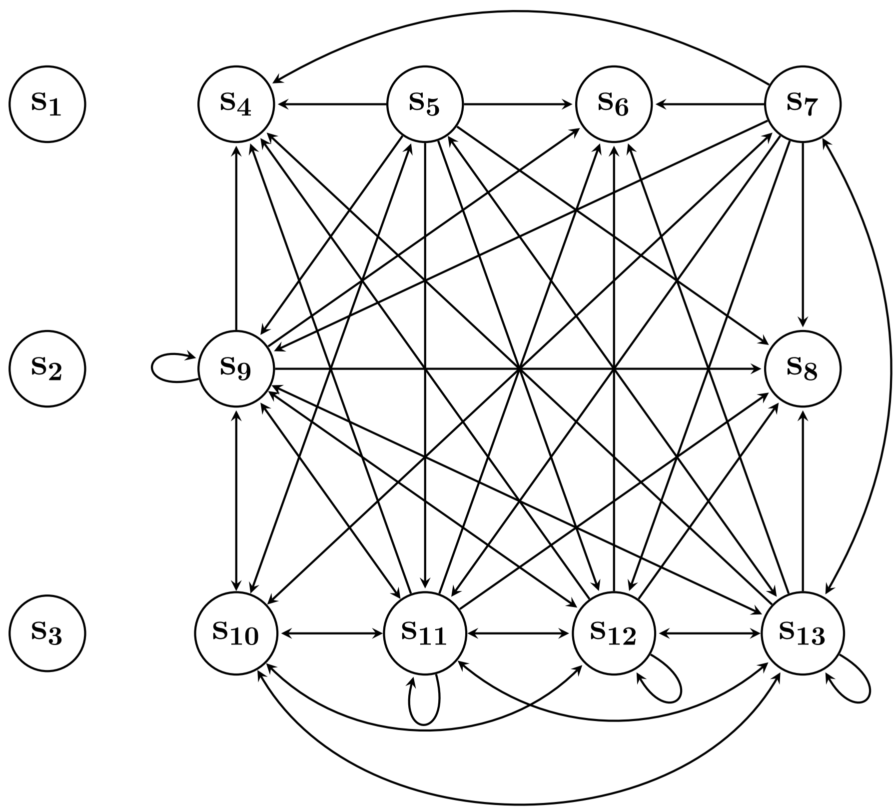
The translation Tr_{AF}^{EAS} is not modular. We classify Tr_{AF}^{EAS} as a coalition translation under the listed semantics and casting functions.

Example 128. Let $ES = (\{\eta, a, b, c, d, e\}, \{(\{a, b\}, d), (\{d\}, c), (\{e\}, d), \{(\{\eta\}, a), (\{\eta\}, c), (\{\eta\}, d), (\{c\}, b)\})$ be the EAS depicted in Figure 106a. The self–supporting sets of ES are $s_1 = \emptyset$, $s_2 = \{\eta\}$, $s_3 = \{\eta, a\}$, $s_4 = \{\eta, c\}$, $s_5 = \{\eta, d\}$, $s_6 = \{\eta, a, c\}$, $s_7 = \{\eta, a, d\}$, $s_8 = \{\eta, b, c\}$, $s_9 = \{\eta, c, d\}$, $s_{10} = \{\eta, a, b, c\}$, $s_{11} = \{\eta, a, c, d\}$, $s_{12} = \{\eta, b, c, d\}$ and $s_{13} = \{\eta, a, b, c, d\}$. We can observe that argument e does not appear in any of these sets due to the fact it does not possess an evidential sequence at all. From these extensions, the sets s_1 to s_8 and s_{10}, are strongly self–supporting; all of the remaining ones can be easily excluded due to the conflict from d to c. When we move to admissibility, we need to further remove the sets s_4, s_6 and s_8, i.e. all of those that contain c without containing both a and b at the same time – without those arguments present, c cannot be defended from the minimal e–supported attack $\{\eta, d\}$. This leaves us with the sets s_1, s_2, s_3, s_5, s_7 and s_{10}. Next, we come to the complete extensions, namely $s_3 = \{\eta, a\}$, $s_7 = \{\eta, a, d\}$ and $s_{10} = \{\eta, a, b, c\}$. The set s_3 is now our grounded extension and s_7 and s_{10} are preferred and stable.

Let us now focus on our associated coalition AF. The collection of the listed self–supporting sets will form its set of arguments. We can now create the following conflicts between them; first of all, s_5 attacks every set containing c, which is s_4, s_6 and s_8 to s_{13}. The same holds for s_7, s_9 and s_{11} to s_{13}. Then, every set containing both a and b attacks those that contain d. This means that sets s_{10} and s_{13} attack s_5, s_7, s_9 and s_{11} to s_{13}. We can observe that certain conflicts are in fact duplicated. This gives us our, somewhat complicated, coalition AF, depicted in Figure 106b. For a moment, let us ignore the set arguments s_1, s_2 and s_3. The admissible extensions of the remaining part of the framework are $\emptyset$, $\{s_5\}$, $\{s_7\}$, $\{s_{10}\}$, $\{s_5, s_7\}$, $\{s_4, s_{10}\}$, $\{s_6, s_{10}\}$, $\{s_8, s_{10}\}$, $\{s_4, s_6, s_{10}\}$, $\{s_4, s_8, s_{10}\}$, $\{s_6, s_8, s_{10}\}$ and $\{s_4, s_6, s_8, s_{10}\}$. We can freely combine them with the initial arguments s_1, s_2 and s_3 to obtain all 96 admissible extensions of our AF. We can now use the fact that $s_1 \subset s_2 \subset s_3 \subset s_7$, $s_2 \subset s_5 \subset s_7$, $s_3 \subset s_6 \subset s_{10}$, $s_2 \subset s_4 \subset s_6 \subset s_{10}$ and $s_4 \subset s_8 \subset s_{10}$ in order to show that our admissible extensions indeed correspond to the admissible sets s_1, s_2, s_3, s_5, s_7 and s_{10} of ES. Fortunately, only three admissible extensions of our AF are



(a) Sample EAS



(b) Associated AF

Figure 106: Sample EAS and its associated AF

complete, namely $\{s_1, s_2, s_3\}$, $\{s_1, s_2, s_3, s_5, s_7\}$ and $\{s_1, s_2, s_3, s_4, s_6, s_8, s_10\}$. The first set is grounded, while the other two are stable and preferred. Once we perform the union of all the set arguments in the complete extensions, we obtain the sets $\{\eta, a\}$, $\{\eta, a, d\}$ and $\{\eta, a, b, c\}$, which were the original complete extensions of our framework ES. We can observe that the grounded, preferred and stable extensions are also correctly retrieved.

11.1.1 Improvements

The coalition translation from EASs to AFs is strong and from some point on, even semantics bijective. Based on our observations, we can say that these two properties usually point to the fact that a faithful translation can be created. In this case, it is true; we can chain some of the EAS–SETAF and SETAF–AF translations in order to obtain the desired results. The question is, whether we can go further, i.e. create an exact translation. This is not an unreasonable request, given the results for the AFN case (see Section 10.1.1). Unfortunately, even though the support relations in both frameworks are quite closely connected (see Translation 68 and Section 11.4), the attack relations are not. Independently of the actual evidential support, EASs work with group, not binary, conflict. This, as seen in Section 6.1.3, puts it already beyond the reach of AFs. We can thus easily adapt the examples from Section 6.1.3 by the means of Translation 30 in order to show that no exact EAS–AF translation can exist:

Theorem 11.2. *Let Fr^{EAS} be the collection of all EASs on a domain $\mathcal{U}^{EAS}$ and Fr^{AF} the collection of all AFs on a domain $\mathcal{U}^{AF}$. There exists no full translation from Fr^{EAS} to Fr^{AF} that is exact under conflict–free, admissible, complete, preferred and stable semantics and identity casting functions for them.*

11.2 EAS as SETAF

11.2.1 Coalition Translation

In the previous section on the coalition EAS–AF translation, we have noted that if we used only the minimal self–supporting sets in the construction of the target frameworks, we would not obtain a structure producing the desired extensions. This was a side effect of the fact that EASs work with group, and AFs with binary attack. However, as we now work with SETAFs, we can safely assume minimality; the price of that is the need to re–adapt the attack relation. A given set of coalition arguments carries out a group attack if the collection of the arguments it represents contains a subset that carries out the attack in the source framework. Moreover, we restrict ourselves to those coalition arguments that are in fact related to the conflict itself.

Translation 72. *Let $ES = (A, R, E)$ be an EAS. Its corresponding coalition SETAF $SF^{ES} = (A', R')$ is created as follows:*

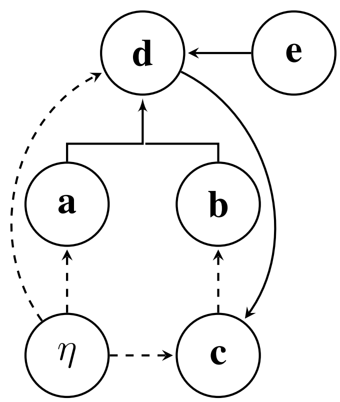
- *for every minimal self–supporting set $S \subseteq A$ for an argument $a \in A$, add S as an argument in A', and*
- *for every $(X, y) \in R$, add (X', Y') to R', where $X', Y' \in A'$, $X \subseteq \bigcup X'$, $\forall V \in X', X \cap V \neq \emptyset$ and $y \in Y'$.*

The results for the EAS–SETAF translation are similar as in the EAS–AF case. The only difference is the improvement in strength of the strongly self–supporting sets:

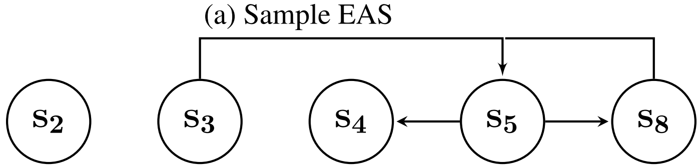
Theorem 11.3. *Let $ES = (A, R, E)$ be an EAS and $SF^{ES} = (A', R')$ its corresponding SETAF obtained by Translation 72. If $S \subseteq A$ is conflict–free in ES, then there might not exist a conflict–free extension $S' \subseteq A'$ of SF^{ES} s.t. $S = \bigcup S'$. If $S \subseteq A$ is strongly self–supporting in ES, then there exists a conflict–free extension $S' \subseteq A'$ of SF^{ES} s.t. $S = \bigcup S'$. If S is a σ–extension of ES, where $\sigma \in \{$admissible, complete, preferred, grounded stable$\}$, then there exists a σ–extension $S' \subseteq A'$ of SF^{ES} s.t. $S = \bigcup S'$.*

If $S' \subseteq A'$ is conflict–free in SF^{ES}, then $\bigcup S'$ is strongly self–supporting (and thus also conflict–free) in ES. If $S' \subseteq A'$ is a σ–extension of SF^{ES}, then $\bigcup S'$ is a σ–extension of ES.

Despite the improvements in the number of coalition arguments we need to create, this translation has the same properties as Translation 71 (the only exception is the strength of the strongly self–supporting semantics). Thus, we can omit any further analysis.



(a) Sample EAS



(b) Associated SETAF

Figure 107: Sample EAS and its associated coalition SETAF

Example 129. Let $ES = (\{\eta, a, b, c, d, e\}, \{(\{a, b\}, d), (\{d\}, c), (\{e\}, d), \{(\{\eta\}, a), (\{\eta\}, c), (\{\eta\}, d), (\{c\}, b)\})$ be the EAS from Example 128, now depicted in Figure 107a. Its self–supporting sets were $s_1 = \emptyset$, $s_2 = \{\eta\}$, $s_3 = \{\eta, a\}$, $s_4 = \{\eta, c\}$, $s_5 = \{\eta, d\}$, $s_6 = \{\eta, a, c\}$, $s_7 = \{\eta, a, d\}$, $s_8 = \{\eta, b, c\}$, $s_9 = \{\eta, c, d\}$, $s_{10} = \{\eta, a, b, c\}$, $s_{11} = \{\eta, a, c, d\}$, $s_{12} = \{\eta, b, c, d\}$ and $s_{13} = \{\eta, a, b, c, d\}$. To every argument we can assign the minimal self–supporting sets containing it; for η it is simply s_2, for a it is s_3, s_8 for b, s_4 for c and s_5 for d. These are now the sets that will appear as arguments in the coalition SETAF associated with ES. Let us now construct the attacks; the (d, c) attack leads to conflicts from s_5 to s_4 and s_8. The $(\{a, b\}, d)$ one is now represented with $(\{s_3, s_8\}, s_5)$. We now obtain the SETAF depicted in Figure 107b, distinctively more straightforward than the coalition AF from Example 128.

The admissible extensions of our SETAF are $\emptyset$, $\{s_2\}$, $\{s_3\}$, $\{s_5\}$, $\{s_2, s_3\}$, $\{s_2, s_5\}$, $\{s_3, s_5\}$, $\{s_3, s_8\}$, $\{s_2, s_3, s_5\}$, $\{s_2, s_3, s_8\}$, $\{s_3, s_4, s_8\}$ and $\{s_2, s_3, s_4, s_8\}$. Once we cast them back to extensions consisting of normal arguments, not sets of them, we obtain $s_1 = \emptyset$, $s_2 = \{\eta\}$, $s_3 = \{\eta, a\}$, $s_5 = \{\eta, d\}$, $s_7 = \{\eta, a, d\}$ and $s_{10} = \{\eta, a, b, c\}$. We can observe these are exactly the admissible extensions of ES. It is worth mentioning that many of them can be obtained from more than one admissible extension of our SETAF. The complete extensions of the coalition framework are $\{s_2, s_3\}$, $\{s_2, s_3, s_4, s_8\}$ and $\{s_2, s_3, s_5\}$. Their corresponding sets are $s_3 = \{\eta, a\}$, $s_{10} = \{\eta, a, b, c\}$ and $s_7 = \{\eta, a, d\}$, which again are the correct answers. We can now easily check that the grounded, preferred and stable extensions between the two frameworks are also in agreement. Moreover, it is worth nothing that the complete extensions are in one–to–one relation, even though the admissible ones were not.

11.2.2 Attack Propagation Translation

Just like we did in the case of AFNs, we can create an attack propagation translation from EASs to SETAFs. Due to the similarity of the two frameworks w.r.t. the support relation (see Translation 68 and possibly Translation 78, which we will analyze soon), the construction is not much different from the one presented in Translation 63.

Translation 73. *Let $ES = (A, R, E)$ be an EAS and $ES^{wv} = (A', R', E')$ its weak validity form. The corresponding attack propagated SETAF is $SF^{ES} = (A', R'')$, where R'' is created as follows:*

- *for an argument $a \in A'$, let $\{X_1^a, ..., X_n^a\}$ be the collection of all self–supporting sets on A' s.t. $a \in X_i^a$ and let $Z_i^a = \{Z_{i,1}^a, ..., Z_{i,n_i}^a\}$ be the collection of all sets of arguments attacking X_i in R', and*
- *$R'' = \{(\bigcup Z', a) \mid a \in A', Z' \subseteq \bigcup_{i=1}^n Z_i^a$ s.t. $\forall_{i=1}^n Z' \cap Z_i^a \neq \emptyset\}$.*

Remark. We can safely exclude η from the second step of the translation – $\{\eta\}$ is a self–supporting set for η and by definition it cannot be attacked by anyone, no attack propagation will occur. We can also note that if a set of arguments X attacks an argument a, then it will be the case that $XR''a$. This comes from the fact that if X attacks a, then it also attacks every coherent set containing a, and thus can be picked during the propagation step. Finally, a possible optimization could include focusing only on the minimal self–supporting sets for a given argument, not necessarily all of them.

Theorem 11.4. *Let $ES = (A, R, E)$ be an EAS and $SF^{ES} = (A', R'')$ its corresponding attack propagated SETAF obtained by Translation 73. If $S \subseteq A$ is strongly self–supporting in ES, then it is conflict–free in SF^{ES}. It does not necessarily hold for conflict–free semantics. If S is a σ–extension of ES, where $\sigma \in \{$admissible, complete, preferred, grounded, stable$\}$, then it is a σ–extension of SF^{ES}. If $S' \subseteq A'$ is a σ'–extensions of SF^{ES},where $\sigma' \in \{$conflict–free, complete, preferred, grounded, stable$\}$, then it is also a σ'–extension of ES. Not every conflict–free extension of SF^{ES} is strongly self–supporting in ES.*

We can now put the translation into our classification system. Just like in the case of Translation 63, we will first limit ourselves to weakly valid EASs, and then proceed with the general approach.

Translation 74. *Let WV^{EAS} be the collection of all weakly valid EASs and Fr^{SETAF} the collection of SETAFs, both based on argument domain $\mathcal{U}$. The attack propagation translation awv-$Tr^{EAS}_{SETAF} : WV^{EAS} \rightarrow Fr^{SETAF}$ is defined as awv-$Tr^{EAS}_{SETAF}((A, R, E)) = (A, R')$ for a framework $(A, R, E) \in WV^{EAS}$, where $R'' = \{(\bigcup Z', a) \mid a \in A', Z' \subseteq \bigcup_{i=1}^{n} Z_i^a$ s.t. $\forall_{i=1}^{n} Z' \cap Z_i^a \neq \emptyset\}$, $\{X_1^a, ..., X_n^a\}$ is the collection of all self–supporting sets on A' s.t. $a \in X_i$ and $Z_i^a = \{Z_{i,1}^a, ..., Z_{i,n_i}^a\}$ is the collection of all sets of arguments attacking X_i in R.*

The semantics theorem for this translation follows straightforwardly from the Theorem 11.4 and its proof. The analysis of our translation is the same as in the case of Translation 64. Thus, again we can omit further explanations.

Theorem 11.5. *Let $\sigma \in \{$complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the identity casting functions for σ. The translation awv-Tr^{EAS}_{SETAF} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under the strongly self–supporting– conflict–free and admissible semantics and identity casting function. It is $\supseteq$–weak under the conflict–free semantics and identity casting functions.*

Analysis of Translation 74: Under the conflict–free, (conflict–free) strongly self–supporting, admissible, complete, preferred, grounded and stable semantics and their identity casting functions, the translation awv-Tr^{EAS}_{SETAF} is:

- source–subclass, target–subclass, overlapping
- argument domain preserving, argument set preserving, attack relation introducing and support relation removing
- generic and semantics domain preserving
- semantical

Translation $awv\text{-}Tr^{EAS}_{SETAF}$ is not modular. It is exact under the complete, preferred, grounded and stable semantics and the identity casting functions. We classify this approach as an attack propagation translation.

The redefinition of the original translation is now as follows; please note we will use the transformation $wv\text{-}Tr^{EAS}$ (see Translation 8):

Redefinition of Translation 73: *Let Fr^{EAS} be the collection of all EASs and Fr^{SETAF} the collection of all SETAFs, both on domain $\mathcal{U}$. The translation $a\text{-}Tr^{EAS}_{SETAF} : Fr^{EAS} \rightarrow Fr^{SETAF}$ is defined as $a\text{-}Tr^{EAS}_{SETAF}((A, R, E)) = awv\text{-}Tr^{EAS}_{SETAF}(wv\text{-}Tr^{EAS}((A, R, E)))$ for a framework $(A, R, E) \in Fr^{EAS}$.*

Redefinition of Theorem 11.4: *Let $\sigma \in \{$complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} be the identity casting functions for σ. The translation $a\text{-}Tr^{EAS}_{SETAF}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under the strongly self–supporting – conflict–free and admissible semantics and identity casting function. It is $\supseteq$–weak under the conflict–free semantics and identity casting functions.*

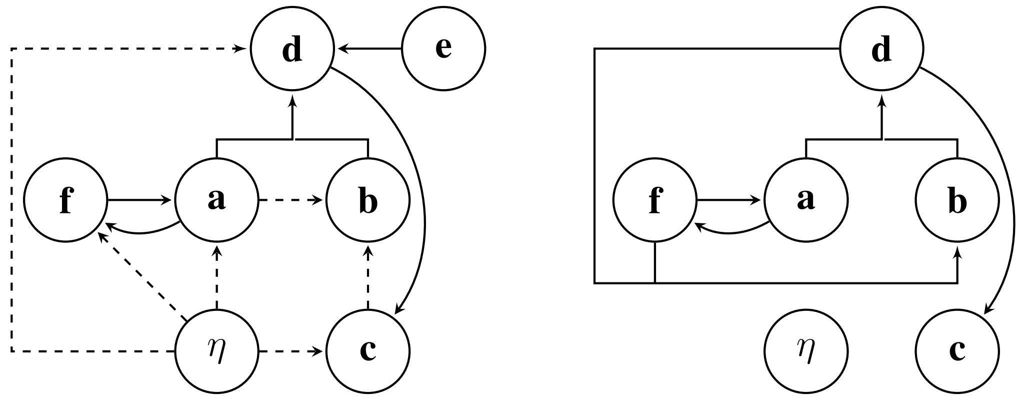
Analysis of Translation 73: Under the conflict–free, (conflict–free) strongly self–supporting, admissible, complete, preferred, grounded and stable semantics and their identity casting functions, the translation $a\text{-}Tr^{EAS}_{SETAF}$ is:

- full, target–subclass, overlapping
- argument domain preserving, argument removing, induced attack relation removing, attack relation introducing and support relation removing
- generic and semantics domain preserving
- semantical

Translation $a\text{-}Tr^{EAS}_{SETAF}$ is not modular. It is exact under the complete, preferred, grounded and stable semantics and the identity casting functions. We classify this approach as an attack propagation translation.

Example 130. Let us consider the EAS $ES = (\{\eta, a, b, c, d, e, f\}, \{(\{a, b\}, d), (\{a\}, f), (\{d\}, c), (\{e\}, d), (\{f\}, a)\}, \{(\{\eta\}, a), (\{\eta\}, c), (\{\eta\}, d), (\{\eta\}, f), (\{a\}, b), (\{c\}, b)\})$, depicted in Figure 108a. The admissible extensions of ES are $\emptyset$, $\{\eta\}$, $\{\eta, a\}$, $\{\eta, f\}$, $\{\eta, a, b\}$, $\{\eta, d, f\}$ and $\{\eta, a, b, c\}$. We can observe that $\{\eta, d\}$ is not an admissible extension; since b possesses two self–supporting sets ($\{\eta, a, b\}$ and $\{\eta, b, c\}$), the attack carried out by d is in fact insufficient in order to prevent the group attack. From this, the sets $\{\eta\}$, $\{\eta, a, b, c\}$ and $\{\eta, d, f\}$ are complete. $\{\eta\}$ is the grounded extension of ES, while $\{\eta, a, b, c\}$ and $\{\eta, d, f\}$ are both preferred and stable.

Let us now focus on describing our attack propagation SETAF. In order to improve readability, we will focus on its minimal form. We can observe that argument e will not appear in the framework due to the fact that it does not possess a self–supporting set. The argument a has only one minimal self–supporting set, namely $\{\eta, a\}$. It will be a subset of every other self–supporting set for a. Consequently, every set of arguments carrying out



(a) Sample EAS

(b) Minimal associated SETAF

Figure 108: Sample EAS and the minimal form of its associated attack propagation SETAF

a propagated attack at a in our SETAF will contain f (the full list of sets is $\{f\}$, $\{f, a\}$, $\{f, d\}$, $\{f, a, b\}$, $\{f, a, d\}$, $\{f, a, b, d\}$). This gives us the first attack $(\{f\}, a)$. Let us now consider b; its minimal self–supporting sets are $\{\eta, a, b\}$ and $\{\eta, b, c\}$. From this, we can construct a (minimal) propagated attack $(\{d, f\}, b)$. Similarly, we can create the conflicts $(\{d\}, c)$, $(\{a, b\}, d)$ and $(\{a\}, f)$. Hence, we obtain our minimal form attack propagation SETAF $SF^{ES} = (\{a, b, c, d, f\}, \{(\{f\}, a), (\{d, f\}, b), (\{d\}, c), (\{a, b\}, d), (\{a\}, f)\})$, depicted in Figure 108b. The admissible extensions of SF^{ES} are $\emptyset$, $\{\eta\}$, $\{a\}$, $\{f\}$, $\{\eta, a\}$, $\{\eta, f\}$, $\{a, b\}$, $\{d, f\}$, $\{\eta, a, b\}$, $\{\eta, d, f\}$, $\{a, b, c\}$ and $\{\eta, a, b, c\}$. We can observe that not all of them are strongly self–supporting in the source EAS, let alone admissible. Out of these sets, $\{\eta\}$, $\{\eta, d, f\}$ and $\{\eta, a, b, c\}$ are complete. We can observe we retrieve the correspondence between the extensions of ES and SF^{ES}. It is now easy to verify that the grounded, preferred and stable extensions of both frameworks are also in agreement.

11.2.3 Defender Translation

The defender translation for EASs to SETAFs follows the design of Translation 65. The only difference lies in the creation of primed arguments. In AFNs, arguments that do not need any support at all are valid by default. Thus, the introduction of primed arguments was required only for those elements that were necessarily supported by some set of arguments. In EASs, only η is valid from the start; every other argument needs to come back to it. In particular, non–η elements that receive no support at all are invalid by default. Consequently, unlike in AFNs, we need to add the *I am not supported* auxiliary argument for every non–η argument in the framework.

Translation 75. *Let $ES = (A, R, E)$ be an EAS. Its corresponding defender SETAF $SF^{ES} = (A', R')$ is constructed the following way:*

- $A' = A \cup \{a' \mid a \in A \setminus \{\eta\}\}$. *The meaning of a' is "a is not e–supported",*

- *add to R' all attacks from R and attacks from a': $R' = R \cup \{(\{a'\}, a) \mid a \in A \setminus \{\eta\}\}$, and*

- *let $a \in A \setminus \{\eta\}$. For any minimal evidential sequence $(a_0, ..., a_n)$ for a, add $(\{a_0, ..., a_{n-1}\}, a')$ to R'.*

The semantics are preserved by the defender EAS–SETAF translation in the same way they were in the AFN–SETAF one (see Theorem 10.11):

Theorem 11.6. *Let $ES = (A, R, E)$ be an EAS and SF^{ES} its corresponding defender SETAF obtained by Translation 75. By $S_{np} = \{a' \mid$ there is no self–supporting set containing $a\} \cup \{a' \mid$ for every self–supporting set C for a, $\exists S' \subseteq S, c \in C \setminus \{a\}$ s.t. $(S', c) \in R\}$ we will denote the primed arguments corresponding to a subset of S^+ in which every argument a either has no self–supporting set or every such set is attacked by S on an argument different from a.*

If a set $S \subseteq A$ is conflict–free in ES, then it is conflict–free in SF^{ES}. The set $S \cup S_{np}$ is not necessarily conflict–free in SF^{ES}. If a set $S \subseteq A$ is strongly self–supporting in ES, then $S \cup S_{np}$ is conflict–free in SF^{ES}. If S is a σ–extension of ES, where $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$, then $S \cup S_{np}$ is a σ–extension of SF^{ES}.

If a set $S' \subseteq A'$ is a σ'–extension of SF^{ES}, where $\sigma' \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, then $S = S' \cap A$ is a σ'–extension of ES. If S' is conflict–free in SF^{ES}, then $S = S' \cap A$ might not be strongly self–supporting in ES.

The analysis of our approach is also similar to the AFN case. Thus, we will omit further explanations.

Redefinition of Translation 75: *Let Fr^{EAS} be the collection of all EASs on the domain $\mathcal{U}$ and Fr^{SETAF} the collection of all SETAFs on argument domain $\mathcal{U} \cup \mathcal{U}'$. The translation $def\text{-}Tr^{EAS}_{SETAF} : Fr^{EAS} \rightarrow Fr^{SETAF}$ is defined as $def\text{-}Tr^{EAS}_{SETAF}((A, R, E)) = (A', R')$ for a framework $(A, R, E) \in Fr^{EAS}$, where $A' = A \cup X'$ for $X' = \{a' \mid a \in A \setminus \{\eta\}\}$, and $R' = R \cup \{(\{a'\}, a) \mid a \in A \setminus \{\eta\}, a' \in X'\} \cup \{(\{a_0, .., a_{n-1}\}, a') \mid a \in A \setminus \{\eta\}, a' \in X$, $(a_0, ..., a_n)$ is a minimal evidential sequence for $a\}$.*

Redefinition of Theorem 11.6: *Let $\sigma^{EAS} \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the removal casting functions for σ defined as $SC^{X}_{\sigma}(S) = S \cap A$ for $X = (A, R, E) \in Fr^{EAS}$ and $S \in \sigma(def\text{-}Tr^{EAS}_{SETAF}(X))$. The translation $def\text{-}Tr^{EAS}_{SETAF}$ is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under strongly self–supporting–conflict–free semantics and the defined casting functions. It is semantics bijective under complete, preferred, grounded and stable semantics and the defined casting functions.*

Analysis of Translation 75: Under the conflict–free, (conflict–free) strongly self–supporting, admissible, complete, preferred, grounded and stable semantics and their removal casting functions, the translation $def\text{-}Tr^{EAS}_{SETAF}$ is:

- full, target–subclass, overlapping

- weakly argument domain altering, argument introducing, induced attack relation introducing and support removing
- generic and weakly semantics domain altering
- semantical

Translation $def\text{-}Tr^{EAS}_{SETAF}$ is not modular. It is faithful under the complete, preferred, grounded and stable semantics and the defined removal casting functions. We classify this approach as a defender translation.

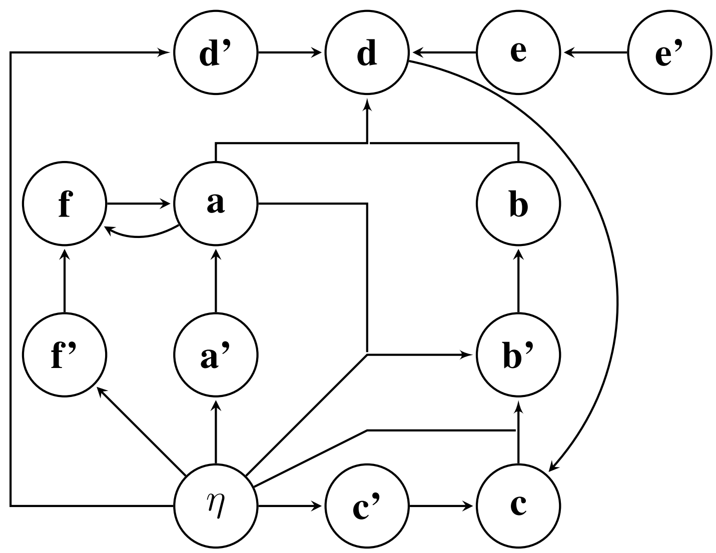


Figure 109: Sample defender SETAF

Example 131. Let us consider the EAS $ES = (\{\eta, a, b, c, d, e, f\}, \{(\{a, b\}, d), (\{a\}, f), (\{d\}, c), (\{e\}, d), (\{f\}, a)\}, \{(\{\eta\}, a), (\{\eta\}, c), (\{\eta\}, d), (\{\eta\}, f), (\{a\}, b), (\{c\}, b)\})$, previously depicted in Figure 108a and described in Example 130. We can now construct the defender SETAF for our framework. First of all, as an addition to the arguments $\{\eta, a, b, c, d, e, f\}$, we need to introduce a primed attacker for every non–η argument in ES. We then allow them to be attacked by the members of (minimal) evidential sequences of the arguments they were created for. In other words, to the conflicts already in ES and the collection $\{(\{a'\}, a), (\{b'\}, b), (\{c'\}, c), (\{d'\}, d), (\{e'\}, e), (\{f'\}, f)\}$ we also need to add the attacks $\{(\{\eta\}, a'), (\{\eta\}, c'), (\{\eta\}, d'), (\{\eta\}, f'), (\{\eta, a\}, b'), (\{\eta, c\}, b')\}$. We thus obtain the SETAF depicted in Figure 109.

The admissible extensions of our SETAF are $\emptyset$, $\{\eta\}$, $\{e'\}$, $\{\eta, a\}$, $\{\eta, e'\}$, $\{\eta, f\}$, $\{\eta, a, b\}$, $\{\eta, a, e'\}$, $\{\eta, e', f\}$, $\{\eta, a, b, c\}$, $\{\eta, a, b, e'\}$, $\{\eta, d, e', f\}$, $\{\eta, a, b, c, e'\}$ and $\{\eta, b', d, e', f\}$. Once we remove the primed arguments, we obtain the sets $\emptyset$, $\{\eta\}$, $\{\eta, a\}$, $\{\eta, f\}$, $\{\eta, a, b\}$, $\{\eta, a, b, c\}$ and $\{\eta, d, f\}$, which were the original admissible extensions of ES. We can observe that more than one SETAF set can correspond to an EAS one.

From all of these extensions, the sets $\{\eta, e'\}$, $\{\eta, b', d, e', f\}$ and $\{\eta, a, b, c, e'\}$ are complete, with the first one being also grounded and the remaining two preferred and stable. We can see that these answers are again the ones we would expect. Additionally, the extensions of ES and our SETAF are now in a one–to–one relation.

11.2.4 Improvements

In this section we have proposed three translations from EASs to SETAFs; the coalition, attack propagation, and defender approaches. Although we have not discussed it, a specialized translation for the conflict–free semantics can be introduced in the same way as done for AFNs in Translation 62. Unfortunately, none of the other approaches we have presented were modular, as it is often the case with semantical translations. This situation is caused by the fact that the validity of an argument can change with the addition and removal of framework elements. Although we are not yet sure how a modular translation can be constructed, we can still propose certain improvements that bring us closer to a structural approach.

All of the translations can be brought into a semi–structural level if we assume that the source EASs are strongly valid. This lifts the validity checks from the extensions and we can simply use sets of arguments s.t. each argument is supported by the set the way it is defined by the evidence support relation. This is unfortunately the only improvement that can be considered in the coalition approach unless we drop the support relation altogether. The situation improves slightly in the attack propagation case; if we were to further limit ourselves to the support singular EASs, then every argument would have a single (minimal) evidential sequence. Thus, the propagation of attacks would be simplified. However, just like in the AFN–SETAF case, the greatest improvements can be observed in the defender translation. Due to strong validity, a given argument can now be defended from the auxiliary primed argument by its nearest supporters, not necessarily the whole self–supporting sets. This brings us to the following translation:

Translation 76. *Let $ES = (A, R, E)$ be a strongly valid EAS. Its corresponding defender SETAF $SF^{ES} = (A', R')$ is constructed the following way:*

- $A' = A \cup \{a' \mid a \in A \setminus \{\eta\}\}$*. The meaning of a' is "a is not e–supported",*
- *add to R' all attacks from R and attacks from a': $R' = R \cup \{(\{a'\}, a) \mid a \in A \setminus \{\eta\}\}$, and*
- *for every set $C \subseteq A$ s.t. CEa, add (C, a') to R'.*

Although now the framework looks a little bit different, a similar semantics theorem holds. Due to the fact that we deal with the strongly valid frameworks, every argument will possess an evidential sequence. Thus, the definition of S_{np} can be simplified. Although we will still use the notion of self–supporting sets in constructing it, please note that their computation is simplified due to Theorem 4.37.

Theorem 11.7. *Let $ES = (A, R, E)$ be a strongly valid EAS and SF^{ES} its corresponding defender SETAF obtained by Translation 76. By $S_{np} = \{a' \mid$ for every self–supporting set C for a, $\exists S' \subseteq S, c \in C \setminus \{a\}$ s.t. $(S', c) \in R\}$ we will denote the primed arguments corresponding to a subset of S^+ in which every self–supporting set for an argument a is attacked by S on an argument different from a.*

If a set $S \subseteq A$ is conflict–free in ES, then it is conflict–free in SF^{ES}. The set $S \cup S_{np}$ is not necessarily conflict–free in SF^{ES}. If a set $S \subseteq A$ is strongly self–supporting in ES, then $S \cup S_{np}$ is conflict–free in SF^{ES}. If S is a σ–extension of ES, where $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$, then $S \cup S_{np}$ is a σ–extension of SF^{ES}.

If a set $S' \subseteq A'$ is a σ'–extension of SF^{ES}, where $\sigma' \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, then $S = S' \cap A$ is a σ'–extension of ES. If S' is conflict–free in SF^{ES}, then $S = S' \cap A$ might not be strongly self–supporting in ES.

In the AFN case, the strongly valid translation had almost the same properties as the original one. In the EAS case, the differences are much more prominent. Thus, we provide an additional analysis:

Redefinition of Translation 76: *Let SV^{EAS} be the collection of all strongly valid EASs on the domain $\mathcal{U}$ and Fr^{SETAF} the collection of all SETAFs on argument domain $\mathcal{U} \cup \mathcal{U}'$. The translation $sdef\text{-}Tr^{EAS}_{SETAF} : SV^{EAS} \rightarrow Fr^{SETAF}$ is defined as $sdef\text{-}Tr^{EAS}_{SETAF}((A, R, E)) = (A', R')$ for a framework $(A, R, E) \in SV^{EAS}$, where $A' = A \cup X'$ for $X' = \{a' \mid a \in A \setminus \{\eta\}\}$, and $R' = R \cup \{(\{a'\}, a) \mid a \in A \setminus \{\eta\}, a' \in X'\} \cup \{(C, a') \mid (C, a) \in E, a \in A \setminus \{\eta\},\ a' \in X\}$.*

Redefinition of Theorem 11.7: *Let $\sigma^{EAS} \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ be a semantics and SC^{Tr}_{σ} the removal casting functions for σ defined as $SC^{X}_{\sigma}(S) = S \cap A$ for $X = (A, R, E) \in Fr^{EAS}$ and $S \in \sigma(sdef\text{-}Tr^{EAS}_{SETAF}(X))$. The translation $sdef\text{-}Tr^{EAS}_{SETAF}$ is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under strongly self–supporting–conflict–free semantics and the defined casting functions. It is semantics bijective under complete, preferred, grounded and stable semantics and the defined casting functions.*

Analysis of Translation 76: Under the conflict–free, (conflict–free) strongly self–supporting, admissible, complete, preferred, grounded and stable semantics and their removal casting functions, the translation $sdef\text{-}Tr^{EAS}_{SETAF}$ is:

- source–subclass, target–subclass, injective
- weakly argument domain altering, argument introducing, induced attack relation introducing
- generic and weakly semantics domain altering
- semi–structural and modular

Translation $sdef\text{-}Tr^{EAS}_{SETAF}$ is faithful under the complete, preferred, grounded and stable semantics and the defined removal casting functions. We classify this approach as a basic defender translation.

Explanation. We are now considering only the strongly valid EASs as our input; thus, the translation is source–subclass. It is also target–subclass, similarly to the original version. However, the translation now becomes injective. The original arguments and attacks can be retrieved by removing primed arguments and the conflicts connected to them; this much has been possible in the original translation as well. However, now the sets of arguments attacking a given primed argument correspond precisely to the original evidential supports, nothing is removed or added. From this we can observe that our approach is injective and no longer support relation removing, even though further structural properties are the same as in the previous version of the defender EAS–SETAF approach.

The most prominent changes w.r.t. Translation 75 can be observed in the computational properties. We are now dealing with a semi–structural translation (we choose not to classify it as structural due to the fact that the translation exploits defense). However, up to a certain point, we retrieve modularity. Let $ES_1 = (A_1, R_1, E_1)$ and $ES_2 = (A_2, R_2, E_2)$ be two strongly valid frameworks s.t. $ES_3 = ES_1 \cup ES_2 = (A_3, R_3, E_3)$ is also strongly valid. The union of the SETAFs associated with ES_1 and ES_2 is (A', R'), where $A' = A_1 \cup A_2 \cup \{a' \mid a \in (A_1 \cup A_2) \setminus \{\eta\}\}$ and $R' = R_1 \cup R_2 \cup \{(\{a'\}, a) \mid a \in A_1 \setminus \{\eta\}\} \cup \{(\{a'\}, a) \mid a \in A_2 \setminus \{\eta\}\} \cup \{(C, a') \mid (C, a) \in E_1\} \cup \{(C, a') \mid (C, a) \in E_2\}$. R' can be equivalently written down as $R_1 \cup R_2 \cup \{(\{a'\}, a) \mid a \in (A_1 \cup A_2) \setminus \{\eta\}\} \cup \{(C, a') \mid (C, a) \in E_1 \cup E_2\}$. Since $A_3 = A_1 \cup A_2$, $R_3 = R_1 \cup R_2$ and $E_3 = E_1 \cup E_2$, then $A' = A_3 \cup \{a' \mid a \in A_3 \setminus \{\eta\}\}$ and $R' = R_3 \cup \{(\{a'\}, a) \mid a \in A_3 \setminus \{\eta\}\} \cup \{(C, a') \mid (C, a) \in E_3\}$, which is exactly what we would have obtained by translating ES_3. Therefore, our translation is easily modular under the given assumptions. ■

Finally, we would like to comment on the strength of our approaches. The only exact results we have obtained were for the complete, preferred, grounded and stable semantics. One can repeat the construction from Translation 62 to obtain an exact, though specialized, method for the conflict–free semantics. This leaves us with admissibility. Although we cannot say with absolute certainty whether a full and exact translation exists under this particular semantics (unlike in the AF case, SETAF signatures are not yet described), we believe it is possible based on the results in the AFN case from (see Section 10.1.1). Establishing an appropriate translation is left for future work.

11.3 EAS as BAF

The translation from EASs to BAFs is not much different than from AFNs to BAFs (see Section 10.4). Initially, it was proposed that in order to handle the evidential support one needs to introduce new notions to BAFs [30]. However, the results in [78] showed that the evidential and necessary supports are more tightly connected than it was first anticipated. Therefore, we can exploit this relation and create an approach that will not require modifying BAFs in any particular way. Nevertheless, for similar reasons as in Section 10.4, we need to limit ourselves to strongly valid and support singular EASs. Additionally, we will consider only the attack binary frameworks, though please note that one can exploit e.g. the SETAF–AFN and AFN–BAF Translations 29 and 67 in order to bypass this particular restriction (see also Section 6.2 for some discussion).

Translation 77. *Let $ES = (A, R, E)$ be a support singular, attack binary and strongly valid EAS. The associated BAF is $BF^{ES} = (A, R', S)$, where $R' = \{(a, b) \mid (\{a\}, b) \in R\}$ and $S = \{(a, b) \mid \exists X \subseteq A$ s.t. $a \in X$ and $(X, b) \in E\}$.*

We can notice that this translation produces BAFs with support acyclic subgraphs. Based on Lemma 4.73, we can also observe there is only one argument, namely η, that will not be on the receiving end of any support link:

Theorem 11.8. *Let $ES = (A, R, E)$ be a support singular, attack binary and strongly valid EAS and $BF^{ES} = (A, R', S)$ its associated BAF created with Translation 77. Then BF^{ES} is support acyclic and $\eta \in A$ is the only argument s.t. $\nexists a \in A, aS\eta$.*

We can easily adapt the proof of Theorem 9.16 in order to show the behavior of the semantics. The only thing worth noticing is the fact that in the EAS–BAF direction, we do not need to to add any new arguments or relations to the framework, unlike in the BAF–EAS case. Thus, the current version is somewhat stronger, at least when we consider the particular subclass that we want to translate:

Theorem 11.9. *Let $ES = (A, R, E)$ be a support singular, attack binary and strongly valid EAS, $BF^{ES} = (A, R', S)$ its associated BAF obtained through Translation 77 and $R'' = \{R^{sec}\}$ the collection of secondary attacks in BF^{ES}. Then, a set $Y \subseteq A$ is:*

- *+conflict–free w.r.t. $\emptyset$ in BF^{ES} iff it is conflict–free in ES.*
- *inverse closed under S in BF^{ES} iff it is self–supporting in ES.*
- *+conflict–free w.r.t. R'' and inverse closed under S in BF^{ES} iff it is strongly self–supporting in ES.*
- *an i–admissible extension of BF^{ES} w.r.t. (R'', R'') iff it is admissible in ES.*
- *an i–preferred extension of BF^{ES} w.r.t. (R'', R'') iff it is preferred in ES.*
- *a d–complete extension of BF^{ES} w.r.t. (R'', R'') iff it is complete in ES.*
- *a d–grounded extension of BF^{ES} w.r.t. R'' iff it is grounded in ES.*
- *a stable extension of BF^{ES} w.r.t. R'' iff it is stable in ES.*

The properties of BAF–AFN and AFN–BAF translations were the same. In the EAS case, even though some properties of Translation 60 are true for Translation 77 as well, there are some significant differences as well.

Redefinition of Translation 77: *Let $SSig^{EAS} \cap ABin^{EAS} \cap SV^{EAS}$ be the collection of all support singular, attack binary and strongly valid EASs and $SAcy^{BAF}$ the collection* *of all support acyclic BAFs, both on domain $\mathcal{U}$. The translation $Tr^{EAS}_{BAF} : (SSig^{EAS} \cap ABin^{EAS} \cap SV^{EAS}) \rightarrow SAcy^{BAF}$ is defined as $Tr^{EAS}_{BAF}((A,R,E)) = (A,R,S)$ for a framework $(A,R,N) \in (SBin^{AFN} \cap SV^{AFN})$, where $S = \{(a,b) \mid (\{a\},b) \in N\}$.*

Redefinition of Theorem 11.9: *Let $\sigma^{EAS} \in \{$self–supporting, strongly self–supporting, admissible, preferred, complete, grounded, stable$\}$ be an EAS semantics and $\sigma^{BAF} \in \{$inverse closed, +conflict–free and inverse closed, i–admissible, i–preferred, d–complete, d–grounded, stable$\}$ be a similar BAF semantics with identical parametrization consisting of secondary attacks. Let SC^{Tr}_{σ} be the identity casting functions for σ. The translation Tr^{EAS}_{BAF} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$. It is also strong and semantics bijective under the +conflict–free – conflict–free semantics with empty parametrization and identity casting functions.*

Analysis of Translation 77: Under the (inverse closed) self–supporting, (+conflict–free and inverse closed) strongly self–supporting, (i–) admissible, (i–) preferred, (d–) complete, (d–) grounded and stable semantics with identical parametrization consisting of secondary attacks, (+conflict–free) conflict–free semantics with empty parametrization and identity casting functions, the translation Tr^{EAS}_{BAF} is:

- source–subclass, target–subclass and injective
- argument domain and structure preserving
- generic, semantics domain preserving and exact
- structural and modular

Translation Tr^{EAS}_{BAF} is classified as basic under the listed semantics and casting functions.

Explanation. The fact that the translation is both source and target–subclass can be easily seen from its redefinition. We can observe that the set of arguments is the same both in a given EAS and its associated BAF. In a certain sense, so is the set of conflicts – we are dealing with attack binary EASs and the only modification the R relation undergoes is a structural shift from single element sets to just single elements. Furthermore, as the source frameworks are support singular, the original supporting set in the E relation can be easily retrieved from the target BAF by collecting all of the supporters from the S relation. Thus, our translation is injective.

It is easy to see from the redefinitions of Translation 77 and Theorem 11.9 that both argument and semantics domain are preserved by our transformation. Based on the above explanation we also qualify this approach as structure preserving. The simplicity of this translation makes us classify it as basic. Moreover, unlike the BAF–EAS Translation 60, this approach is modular, at least under the assumption we do not breach the domain restrictions by joining two EASs. ■

11.4 EAS as AFN

In our previous work we have already studied the problem of translating EASs into AFNs [77, 78]. Our focus was mostly on how to shift between the necessary and evidential supports. Consequently, we have only considered a subclass of EASs which made use of binary attacks. In this section we will present the existing translation and propose a general approach.

11.4.1 Attack Binary EASs

Structurally speaking, transforming the evidential support relation E into the necessary one N is like shifting between DNF and CNF forms, as already stated in Section 10.5. However, there is a slight difference in what is considered a valid argument in AFNs and EASs that needs to be dealt with. In AFNs, the ability to derive an argument (through support) in an acyclic manner was sufficient. In EASs, it is further strengthened by requiring not just acyclicity, but grounding in evidence. In other words, an argument that receives no support at all would be considered valid in AFNs and quite the opposite in EASs. An argument not coming from evidence, which is treated as the only source of truth, claims validity on its own, thus in a sense behaving as a self–supporter. We draw on this observation and modify N by introducing additional links for such arguments, which renders them (and whatever is based on them) AFN–invalid as desired.

Translation 78. *Let $ES = (A, R, E)$ be an attack binary EAS. The corresponding AFN is $FN^{ES} = (A, R', N)$, where R' and N are created as follows:*

- *for every two arguments a, b s.t. $(\{a\}, b) \in R$, put (a, b) in R', and*
- *let $a \neq \eta$ be an argument in A and $Z = \{Z_1, .., Z_n\}$ be a collection of all sets Z_i s.t. $Z_i E a$. If Z is empty, add $(\{a\}, a)$ to N. Otherwise, for every subset Z' of $\bigcup_{i=1}^{n} Z_i$ s.t. $\forall_{i=1}^{n} Z' \cap Z_i \neq \emptyset$, add (Z', a) to N.*

Again, we may note that the way the N relation is created can lead to certain redundancies in the framework. Nevertheless, it can be addressed by assuming minimality of the created Z' sets or using constructions such as in Section 10.5. Whichever way we choose, we still produce frameworks that are assigned the same minimal normal form, and thus from the semantical perspective this is not an important issue. Moreover, our target framework still satisfies certain normal forms:

Theorem 11.10. *Let $ES = (A, R, E)$ be an attack binary EAS and $FN^{ES} = (A, R', N)$ its associated AFN obtained through Translation 78. If FN^{ES} is strongly consistent, then so is ES. However, it is not the case that if ES is strongly consistent, then so is FN^{ES}. If ES is all–supported and strongly consistent, then FN^{ES} is strongly consistent. FN^{ES} might not be in minimal form, even if ES is. If ES is weakly (strongly) valid, then so is FN^{ES}. If ES is weakly and relation valid, then FN^{ES} is relation valid.*

It is worth noting that due to the differences between necessary and evidential supports, the produced AFNs will all have one thing in common – there will be only one argument, namely η, that will not receive any support and will not participate in any conflict. Although the property can seem trivial, the fact that this single argument will be at the beginning of every powerful sequence can be exploited for the computational purposes.

Let us now look at the semantics. We can observe that even though the necessary support contains certain links not present in the evidential one, the valid arguments are the same in both structures and there is a strong relation between self–supporting and coherent sets [77, 78]:

Theorem 11.11. *Let $ES = (A, R, E)$ be an attack binary EAS and $FN^{ES} = (A, R', N)$ its associated AFN obtained through Translation 78. $(a_0, ..., a_n)$ is an evidential sequence for a on $S \subseteq A$ in ES iff $(a_0, ..., a_n)$ is a powerful sequence for a on S in FN^{ES}. S is self–supporting in ES iff it is coherent FN^{ES}.*

Since the attack relation is not altered in any particular way, we can also connect defense in AFNs with acceptability in EASs quite easily. However, please note that there is a particular difference in the support requirements of these two notions; in AFNs, defense leads to the coherence of the defending set, while EAS acceptability is more relaxed. Therefore, we need to assume that we are working with self–supporting (coherent) sets, just like in the case of Translation 68.

Theorem 11.12. *Let $ES = (A, R, E)$ be an attack binary EAS and $FN^{ES} = (A, R', N)$ its associated AFN obtained through Translation 78. Let $S \subseteq A$ be a self–supporting (coherent) set in ES (FN^{ES}). An argument $a \in A$ is acceptable w.r.t. S in ES iff it is defended by S in FN^{ES}.*

The above results bring us to the correspondence between the extensions of the two frameworks [77, 78]:

Theorem 11.13. *Let $ES = (A, R, E)$ be an attack binary EAS and $FN^{ES} = (A, R', N)$ its corresponding AFN obtained through Translation 78. A set $S \subseteq A$ is (strongly) self–supporting in ES iff it is (strongly) coherent in FN^{ES}. S is a σ–extension in ES, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, iff it is a σ–extension in FN^{ES}.*

We complete our previous results by an additional analysis of the properties of this translation:

Redefinition of Translation 78: *Let $ABin^{EAS}$ be the collection of all attack binary EASs and Fr^{AFN} the collection of all AFNs, both based on the domain $\mathcal{U}$. The translation bin-$Tr^{EAS}_{AFN} : ABin^{EAS} \rightarrow Fr^{AFN}$ is defined as bin-$Tr^{EAS}_{AFN}((A, R, E)) = (A, R', N)$ for a framework $(A, R, E) \in Fr^{EAS}$, where $R' = \{(a, b) \mid (\{a\}, b) \in R\}$ and $N = \{(\{a\}, a) \mid a \in A \setminus \{\eta\}, Z^a = \emptyset\} \cup \{(Z', a) \mid a \in A, Z' \subseteq \bigcup_{i=1}^{n_a} Z_i^a, \forall_{i=1}^{n_a} Z' \cap Z_i^a \neq \emptyset\}$ given that $Z^a = \{Z_1^a, .., Z_n^a\}$ is the collection of all subsets of A s.t. $Z_i^a E a$.*

Redefinition of Theorems 11.11 and 78: *Let $\sigma^{EAS} \in \{$conflict–free, self–supporting, strongly self–supporting, admissible, preferred, complete, grounded, stable$\}$ be an EAS semantics and $\sigma^{AFN} \in \{$conflict–free, coherent, strongly coherent, admissible, preferred, complete, grounded, stable$\}$ be a similar AFN semantics. Let SC_{σ}^{Tr} be the identity casting functions for σ. The translation bin-Tr_{AFN}^{EAS} is strong and semantics bijective under $(\sigma, SC_{\sigma}^{Tr})$.*

Analysis of Translation 78: Under the conflict–free, (coherent) self–supporting, (strongly coherent) strongly self–supporting, admissible, preferred, complete, grounded and stable semantics and identity casting functions, the translation $bin\text{-}Tr_{AFN}^{EAS}$ is:

- source–subclass, target–subclass and overlapping
- argument domain preserving and support relation introducing
- generic, semantics domain preserving and exact
- semi–structural

The translation $bin\text{-}Tr_{AFN}^{EAS}$ is not modular. We classify it as basic under the listed semantics and casting functions.

Explanation. Our translation works only for the attack binary EASs, and thus we classify it as source–subclass. Based on the previous explanations, we also decide to classify it as target–subclass. Our approach is also overlapping for two reasons. First of all, we introduce additional support links into the framework for argument that are not supported at all. This modification means that e.g. two different EASs $(\{\eta, a\}, \emptyset, \emptyset)$ and $(\{\eta, a\}, \emptyset, \{(\{a\}, a)\})$ would be assigned the same AFN $(\{\eta, a\}, \emptyset, \{(\{a\}, a)\})$. Another reason why our translation is overlapping comes from the way new supporting sets are created from the old ones. In this case if two frameworks have the same minimal form, they might (though not in every situation) be assigned the same target structure. We can imagine an argument c supported by sets $\{a\}$, $\{b\}$ and $\{a, b\}$ in one EAS and by $\{a\}$, $\{b\}$ in the other. In both of the target AFNs, c will be assigned the set $\{a, b\}$.

The translation clearly preserves both of the argument and semantics domains. It is also support introducing due to the fact that new links are created for the unsupported arguments. Aside from that, one can choose to perceive certain redundancies in the way the N relation is created as adding support as well. Nevertheless, for reasons similar as in Translation 68, we choose not to classify our approach as support removing, even though e.g. in the example above two different collections of supporting sets in E can be transformed into a single one in N.

The amount of handled semantics classifies our translation as generic. Its exactness follows easily form the redefinition of Theorems 11.11 and 78. Although our approach is rather simple, we choose to classify it as semi–structural due to the required adaptation stemming from the way validity of arguments is interpreted in EASs and AFNs. Finally, our translation is not modular, similarly as in the case of Translation 68. ■

Example 132. Let $ES = (\{\eta, a, b, c, d, e, f, g\}, \{(\{d\}, e), (\{e\}, a), (\{f\}, c), (\{g\}, f)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{b\}, c), (\{\eta\}, e), (\{a, b\}, d), (\{c\}, d), (\{e\}, f)\})$ be the EAS depicted in Figure 110a. Its strongly self–supporting sets are $\emptyset$, $\{\eta\}$, $\{\eta, a\}$, $\{\eta, b\}$, $\{\eta, e\}$, $\{\eta, a, b\}$, $\{\eta, b, c\}$, $\{\eta, b, e\}$, $\{\eta, e, f\}$, $\{\eta, a, b, c\}$, $\{\eta, a, b, d\}$, $\{\eta, b, c, d\}$, $\{\eta, b, c, e\}$, $\{\eta, b, e, f\}$ and $\{\eta, a, b, c, d\}$. From this, the sets $\emptyset$, $\{\eta\}$, $\{\eta, b\}$, $\{\eta, e, f\}$, $\{\eta, a, b, d\}$, $\{\eta, b, c, d\}$, $\{\eta, b, e, f\}$ and $\{\eta, a, b, c, d\}$ are admissible. Finally, the complete extensions are $\{\eta, b\}$, $\{\eta, b, e, f\}$ and $\{\eta, a, b, c, d\}$, with the first set being also grounded and the other two preferred and stable in ES.

Let us now talk about the construction of the AFN FN^{ES} associated with ES. The set of arguments and the attack relation stay the same. Due to the fact that the argument a is supported only by one, single–element set $\{\eta\}$, the same set is its only supporter in FN^{ES}. Similar holds for arguments b, c, e and f. Argument g does not posses an evidential sequence in ES. Consequently, it will become a self–supporter in FN^{ES} and the support $(\{g\}, g)$ needs to be added to the framework. Let us now consider argument d; its new supporting sets are $\{a, c\}$, $\{b, c\}$ and $\{a, b, c\}$. This finally gives us the framework $FN^{ES} = (\{\eta, a, b, c, d, e, f, g\}, \{(d, e), (e, a), (f, c), (g, f)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, e), (\{g\}, g), (\{b\}, c), (\{e\}, f), (\{a, c\}, d), (\{b, c\}, d), (\{a, b, c\}, d)\})$. We can observe that the $(\{a, b, c\}, d)$ support is redundant and can be removed in order to obtain the minimal form of our AFN, depicted in Figure 110b. The (minimal) powerful sequences for d are (η, c, d) and (η, a, b, d). We can easily check that they correspond to the minimal evidential sequences for d in ES.

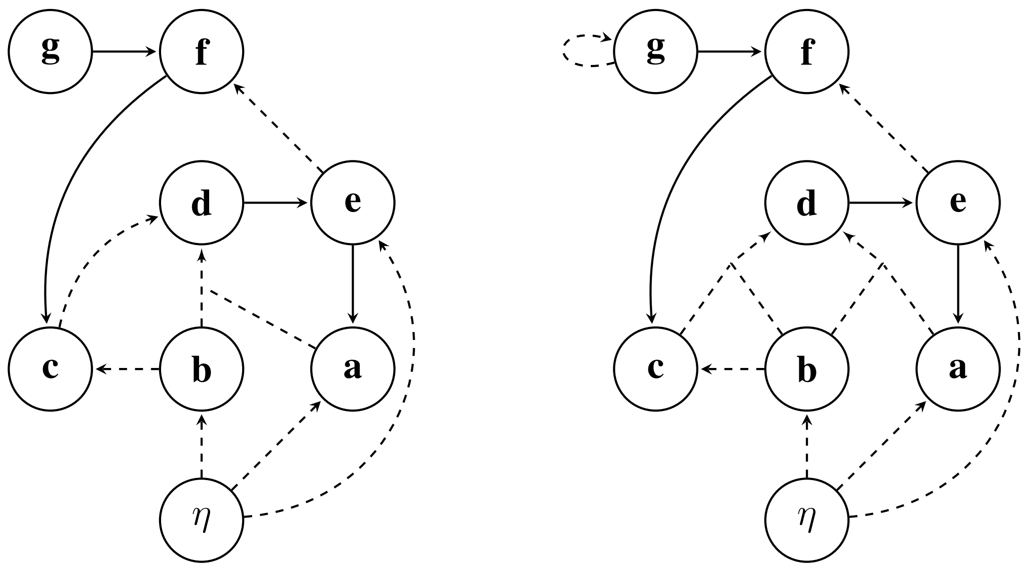
The easy admissible extensions of FN^{ES} are $\emptyset$, $\{\eta\}$ and $\{\eta, b\}$; neither η nor b are attacked in the framework. Another possible set is $\{\eta, a, b, d\}$; all arguments are e–supported by the set and d attacks the only minimal sequence for f. Due to this, it attacks the sequence for f as well. Thus, $\{\eta, b, c, d\}$ and $\{\eta, a, b, c, d\}$ can be shown to be admissible in FN^{ES}. The next admissible set is $\{\eta, e, f\}$ – we can observe that f attacks c and thus every self–supporting set for d. The final extension is $\{\eta, b, e, f\}$. We can observe that these results are the same as in ES.

We can observe that $\emptyset$ defends η, $\{\eta\}$ and $\{\eta, e, f\}$ defend b, $\{\eta, a, b, d\}$ defends c and $\{\eta, b, c, d\}$ defends a. Therefore, our complete extensions are $\{\eta, b\}$, $\{\eta, a, b, c, d\}$ and $\{\eta, b, e, f\}$, which are exactly the ones that ES produces. We can easily check that the grounded, preferred and stable are also the same between ES and FN^{ES}.

11.4.2 General EASs

Translation 78 highlighted the differences between the evidential and necessary supports and showed how one can transform one into the other. However, it left out one important difference between EASs and AFNs, namely the fact that the former deal with group attack, and the latter only with binary. In order to address that, we will merge Translations 29 and 78.

Translation 79. *Let $ES = (A, R, E)$ be an EAS. The corresponding AFN is $FN^{ES} = (A', R', N)$, where A', R' and N are created as follows:*



(a) Sample EAS

(b) Minimal form associated AFN

Figure 110: Sample attack binary EAS and its minimal form associated AFN

- *let* $att(S) = \{S' \mid S' \subseteq S \wedge |S'| > 1 \wedge \exists y \in A\, s.t.\, (S', y) \in R\}$*, where* $S \subseteq A$*, be a function returning subsets of* S *of size bigger than 1, which attack an argument in* A*,*
- $A' = A \cup att(A)$*,*
- $R' = \{(X, y) \mid (X, y) \in R$ *or* $(\{X\}, y) \in R\}$*, and*
- $N = \{(\{X\}, Y) \mid X \in A, Y \in att(A), X \in Y\} \cup \{(\{a\}, a) \mid a \in A \setminus \{\eta\}, Z^a = \emptyset\} \cup \{(Z', a) \mid a \in A, Z' \subseteq \bigcup_{i=1}^{n_a} Z_i^a, \forall_{i=1}^{n_a} Z' \cap Z_i^a \neq \emptyset\}$ *given that* $Z^a = \{Z_1^a, .., Z_n^a\}$ *is the collection of all subsets of* A *s.t.* $Z_i^a E a$*.*

The AFN produced with Translation 79 preserves most of the normal forms that the AFN created with Translation 78 did. However, one can note that previously, strong consistency of the target AFN implied strong consistency of the source EAS. Unfortunately, due to the modifications caused by group attacks, it is no longer the case.

Theorem 11.14. *Let* $ES = (A, R, E)$ *be an EAS and* $FN^{ES} = (A, R', N)$ *its associated AFN obtained through Translation 79. It is not the case that if* ES *is strongly consistent, then so is* FN^{ES}*. Moreover, it is not the case that if* FN^{ES} *is strongly consistent, then so is* ES*. If* ES *is all–supported and strongly consistent, then* FN^{ES} *is strongly consistent.* FN^{ES} *might not be in minimal form, even if* ES *is. If* ES *is weakly (strongly) valid, then so is* FN^{ES}*. If* ES *is weakly and relation valid, then* FN^{ES} *is relation valid.*

The following relation between the extensions of two frameworks can be shown by adapting the proofs Theorems 6.6, 11.11 and 11.13. The properties of our approach will be a result of properties of Translations 29 and 78 and thus we will omit further explanations.

Theorem 11.15. *Let $ES = (A, R, E)$ be an EAS and $FN^{ES} = (A, R', N)$ its corresponding AFN obtained through Translation 79. If $S \subseteq A$ is conflict–free in ES, then both S and $S' = S \cup att(S)$ are conflict–free in FN^{ES}. If $S \subseteq A$ is (strongly) self–supporting in ES, then S and $S' = S \cup att(S)$ are (strongly) coherent in FN^{ES}. If S is a σ–extension of ES, where $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$, then $S' = S \cup att(S)$ is a σ–extension of FN^{ES}. If $S' \subseteq A'$ is a conflict–free extension of FN^{ES}, then $S = S' \cap A$ might not be conflict–free in ES. If S' is coherent in FN^{ES}, then $S = S' \cap A$ is self–supporting in ES. If S' is strongly coherent in FN^{ES}, then $S = S' \cap A$ might not be strongly self–supporting in ES. If S' is σ–extension of FN^{ES}, then $S = S' \cap A$ is a σ–extension of ES.*

Redefinition of Translation 79: *Let Fr^{EAS} be the collection of all EASs based on domain $\mathcal{U}$ and Fr^{AFN} the collection of all AFNs based on the domain $\mathcal{U} \cup (2^{\mathcal{U}} \setminus \emptyset)$. The translation $Tr^{EAS}_{AFN} : Fr^{EAS} \rightarrow Fr^{AFN}$ is defined as $Tr^{EAS}_{AFN}((A, R, E)) = (A', R', N)$ for a framework $(A, R, E) \in Fr^{EAS}$, where $A' = A \cup att(A)$ for $att(A) = \{S' \mid S' \subseteq A \wedge |S'| > 1 \wedge \exists y \in A\, s.t.\, (S', y) \in R\}$, $R' = \{(X, y) \mid (X, y) \in R$ or $(\{X\}, y) \in R\}$ and $N = \{(\{X\}, Y) \mid X \in A, Y \in att(A), X \in Y\} \cup \{(\{a\}, a) \mid a \in A \setminus \{\eta\}, Z^a = \emptyset\} \cup \{(Z', a) \mid a \in A, Z' \subseteq \bigcup_{i=1}^{n_a} Z_i^a, \forall_{i=1}^{n_a} Z' \cap Z_i^a \neq \emptyset\}$ given that $Z^a = \{Z_1^a, .., Z_n^a\}$ is the collection of all subsets of A s.t. $Z_i^a E a$.*

Redefinition of Theorem 11.15: *Let $\sigma^{EAS} \in \{$self–supporting, admissible, preferred, complete, grounded, stable$\}$ be an EAS semantics and $\sigma^{AFN} \in \{$coherent, admissible, preferred, complete, grounded, stable$\}$ be a similar AFN semantics. Let SC^{Tr}_{σ} be the removal casting functions for σ defined as $SC^X_{\sigma}(S) = S \cap A$ for $X = (A, R, E) \in Fr^{EAS}$ and $S \in \sigma^{AFN}(Tr^{EAS}_{AFN}(X))$. The translation Tr^{EAS}_{AFN} is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under the conflict–free and strongly self–supporting – strongly coherent semantics and the defined casting functions. It is semantics bijective under the complete, preferred, grounded and stable semantics and the defined casting functions.*

Analysis of Translation 79: Under the conflict–free, (coherent) self–supporting, (strongly coherent) self–supporting conflict–free, admissible, preferred, complete, grounded and stable semantics and removal casting functions, the translation Tr^{EAS}_{AFN} is:

- full, target–subclass and overlapping
- argument domain altering, argument introducing, support relation introducing, attack relation preserving
- generic and semantics domain altering
- semi–structural

The translation Tr^{EAS}_{AFN} is not modular. It is faithful under the complete, preferred, grounded and stable semantics and removal casting functions. We classify the translation as a basic–coalition approach under the listed semantics and casting functions.

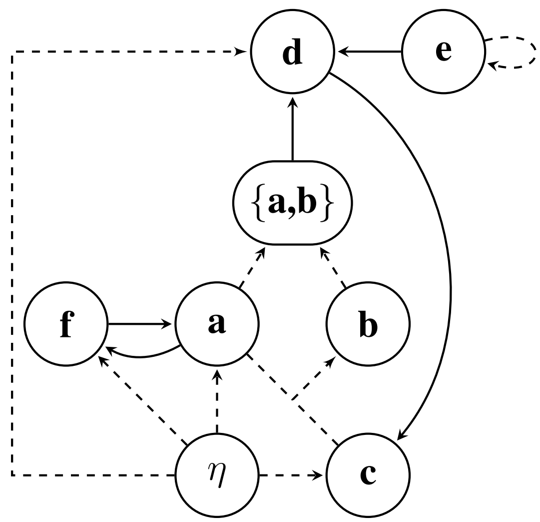


Figure 111: Sample AFN

Example 133. Let us consider the EAS $ES = (\{\eta, a, b, c, d, e, f\}, \{(\{a, b\}, d), (\{a\}, f), (\{d\}, c), (\{e\}, d), (\{f\}, a)\}, \{(\{\eta\}, a), (\{\eta\}, c), (\{\eta\}, d), (\{\eta\}, f), (\{a\}, b), (\{c\}, b)\})$ previously analyzed in Example 130. Its associated AFN is $FN^{ES} = (\{\eta, a, b, c, d, e, f, \{a, b\}\}, \{(\{a, b\}, d), (a, f), (d, c), (e, d), (f, a)\}, \{(\{\eta\}, a), (\{\eta\}, c), (\{\eta\}, d), (\{\eta\}, f), (\{a, c\}, b), (\{e\}, e), (\{a\}, \{a, b\}), (\{b\}, \{a, b\})\})$, as seen in Figure 111.

The admissible extensions of this framework are $\emptyset$, $\{\eta\}$, $\{\eta, a\}$, $\{\eta, f\}$, $\{\eta, a, b\}$, $\{\eta, a, b, \{a, b\}\}$, $\{\eta, d, f\}$ and $\{\eta, a, b, c, \{a, b\}\}$. They correspond to the sets $\emptyset$, $\{\eta\}$, $\{\eta, a\}$, $\{\eta, f\}$, $\{\eta, a, b\}$, $\{\eta, d, f\}$ and $\{\eta, a, b, c\}$, which were the original admissible extensions of ES. We can observe that some of the EAS extensions can be produced from more than one AFN set. The complete extensions of FN^{ES} are $\{\eta\}$, $\{\eta, d, f\}$ and $\{\eta, a, b, c, \{a, b\}\}$. We can again retrieve precisely the desired complete sets of ES; however, this time the relation between the answers of both frameworks is one to one. We can now easily check that grounded, preferred and stable extensions of ES and FN^{ES} are also related in the desired manner.

11.4.3 Improvements

Although we have obtained an exact translation from attack binary EASs to AFNs, the general method is faithful at best. However, based on the results in Sections 6.1.3 and 10.1.1, it is unlikely we can improve the strength of the translation, at least where conflict–free, admissible and preferred semantics are concerned. Unfortunately, it is difficult to say anything conclusive concerning the complete and stable semantics. Future research on semantics signatures will help us solve these issues.

11.5 EAS as ADF

Finally, we can present the possible methods for translating EASs into ADFs. Just like in the case of EAFs and AFNs, we need to classify our approaches w.r.t. the consistency criterion (see Section 2.3.9). Not surprisingly, we will focus on the aa–family of semantics on the ADF side in order to grasp the validity of arguments stemming from support cycles. However, just like in AFNs, an argument that requires no support (i.e. has a minimal decisively in interpretation with an empty $\mathbf{t}$ part) is "good" in ADFs and can start a pd–acyclic sequence. Thus, we need to repeat the EAS–AFN construction and use the self–supporter method in order to ensure that the arguments not backed by evidence are not considered valid.

Similarly as in the AFN–ADF translations, an acceptance condition in EAS–produced ADFs can be seen as consisting of two parts. The first one – the attack part – will be created as in the SETAF–ADF Translation 31. In other words, given the collection of attacking sets $Z = \{Z_1, ..., Z_n\}$ for an argument a, the subformula associated with $Z_i = \{z_1^i, ..., z_{n_i}^i\}$ is $\neg z_1^i \vee .... \vee \neg z_{n_i}^i$ (abbreviated with $\bigvee \neg Z_i$), and the formula associated with Z altogether is $\bigvee \neg Z_1 \wedge ... \wedge \bigvee \neg Z_n$.

In the AFN case, the support part of an acceptance condition was a conjunction of (disjunctive) clauses. In the EAS case, it is a disjunction of conjunctive clauses, as in this framework at least one (full) supporting set of a given argument needs to be present in order to consider the argument sufficiently supported. Therefore, given the collection of all sets $Z = \{Z_1, ..., Z_n\}$ supporting a given argument through the evidential relation, the subformula associated with $Z_i = \{z_1^i, ..., z_{n_i}^i\}$ is $z_1^i \wedge .... \wedge z_{n_i}^i$ (abbreviated with $\bigwedge Z_i$) and the formula associated with Z altogether is $\bigwedge Z_1 \vee ... \vee \bigwedge Z_n$.

Having explained the notation, we can move on to the translations themselves.

11.5.1 Strongly Consistent EASs

We will start with presenting the translation limited to strongly consistent EASs only, as in this case we require no auxiliary arguments and it distinctively shows how evidential support can be simulated within ADFs. The creation of acceptance conditions will in principle follow the construction we have shown before; the only required adaptation is the same as in the case of EAS–AFN Translation 78, i.e. the addition of a support link to those arguments that receive no support at all.

Translation 80. *Let $ES = (A, R, E)$ be a strongly consistent EAS. The corresponding ADF $D^{ES} = (A, L, C)$ is created as follows:*

1. *for every two arguments a, b, if there exists a set $Z \subseteq A$ containing a s.t. ZEb or ZRb, add (a, b) to L. If there is no set $Z \subseteq A$ s.t. ZEb and $b \neq \eta$, add (b, b) to L, and*

2. *for an argument a, the functional acceptance condition is:*

- *if $a = \eta$, then there is no argument attacking or supporting it and the acceptance condition simply maps $\emptyset$ to in,*
- *if $a \neq \eta$ and there exists $B \subseteq A$ s.t. BEa, then:*
  - *every $P' \subseteq par(a)$ is mapped to out iff $\exists B \subseteq P'$ s.t. BRa or $\nexists Z \subseteq P'$ s.t. ZEa, and*
  - *all other subsets of $par(a)$ are in.*
- *if $a \neq \eta$ and there is no subset $B \subseteq A$ s.t. BEa, we include self–support:*
  - *every $P' \subseteq par(a)$ is mapped to out iff $\exists B \subseteq P'$ s.t. BRa or $a \notin P'$, and*
  - *all other subsets of $par(a)$ are in.*

3. *for an argument $a \in A$, the propositional condition is:*
   - *if $a = \eta$, then $C_a = \top$, and*
   - *if $a \neq \eta$, then $C_a = att_a \wedge sup_a$, where att_a and $\sup_a$ are defined as follows:*
     - *let $B = \{B_1, ..., B_n\}$ be the collection of all subsets of A s.t. B_iRa. The attack subformula of the acceptance condition is $att_a = (\bigvee \neg B_1 \wedge ... \wedge \bigvee \neg B_n)$ or $att_a = \top$ if $B = \emptyset$, and*
     - *let $Z = \{Z_1, ..., Z_m\}$ be the collection of all subsets of A s.t. Z_iEa. If $Z = \emptyset$, then $sup_a = a$. If $Z \neq \emptyset$, it is $sup_a = (\bigwedge Z_1 \vee ... \vee \bigwedge Z_m)$.*

Please note that due to the introduction of self–support to unsupported arguments, it might be the case that an argument that was originally strongly consistent receives a falsum acceptance condition. Thus, the target ADF might not be in the cleansed form. However, please note that due to the acyclicity of the semantics, this does not make a difference. A self–supporter is automatically falsified in the acyclic range interpretation due to lack of any acyclic pd–evaluation to start with. A falsum argument is automatically falsified in any type of range interpretation, thus still giving us the desired behavior in this case.

Unfortunately, even though the semantics are not affected, the "accidental" introduction of falsum arguments in this case does break some of the normal forms. If introducing the self–link causes a falsum acceptance condition, then an argument has at least one redundant parent, namely itself. Thus, we breach the redundancy–free form, even if the source EAS was minimal, and this affects the conditions under which we can show that the produced ADF is in certain validity forms.

Theorem 11.16. *Let $ES = (A, R, E)$ be a strongly consistent EAS and $D^{ES} = (A, L, C)$ its corresponding ADF obtained through Translation 80. Then D^{ES} is a BADF. If ES is all–supported, then D^{ES} is cleansed. If ES is all–supported and minimal, then D^{ES} is redundancy–free. If ES is weakly valid, then D^{ES} is weakly valid. If it is minimal, all–supported and relation valid, then D^{ES} is relation valid. If ES is strongly valid, then D^{ES} is an AADF$^+$. If it is in addition minimal, then D^{ES} is strongly valid.*

Despite the issues with normal forms, the semantics behave in a satisfactory manner under this translation. Similarly as in the EAS–AFN Translation 78, we can observe that the relation between the self–supporting and pd–acyclic sets is strong:

Theorem 11.17. *Let $ES = (A, R, E)$ be a strongly consistent EAS and $D^{ES} = (A, L, C)$ its corresponding ADF obtained through Translation 80. Let $S \subseteq A$ be a set of arguments. For a given evidential sequence on S for an argument $s \in S$ we can construct a corresponding acyclic pd–evaluation and vice versa. S is self–supporting in ES iff it is pd–acyclic in D^{ES}.*

We can recall that the conflict–free semantics in ADFs had a slightly broader meaning than in e.g. EASs in AFNs. We focus on satisfying the acceptance conditions and not on checking what is the nature of the relations between the arguments in a given set. Thus, if we wanted to draw connections between EAS conflict–free and ADF conflict–free semantics, we would have to resort to translating the attack subgraph into an ADF, i.e. use Translation 31. In the current approach, we connect (pd–acyclic) conflict–freeness to strongly self–supporting sets:

Theorem 11.18. *Let $ES = (A, R, E)$ be a strongly consistent EAS and $D^{ES} = (A, L, C)$ its corresponding ADF obtained through Translation 80. A set of arguments $S \subseteq A$ is strongly self–supporting in ES iff it is pd–acyclic conflict–free in D^{ES}.*

Having explained the relation between self–supporting and pd–acyclic sets, we can now connect the discarded sets in both frameworks and relate acceptability to being decisively in:

Lemma 11.19. *Let $ES = (A, R, E)$ be a strongly consistent EAS and $D^{ES} = (A, L, C)$ its corresponding ADF obtained through Translation 80. Let $S \subseteq A$ be self–supporting conflict–free in ES and thus pd–acyclic conflict–free in D^{ES}. S attacks $a \in A$ or every set of arguments e–supporting a in ES iff the acyclic range v_S^a of S blocks every acyclic pd–evaluation of a in D^{ES}.*

Lemma 11.20. *Let $ES = (A, R, E)$ be a strongly consistent EAS and $D^{ES} = (A, L, C)$ its corresponding ADF obtained through Translation 80. Let $S \subseteq A$ be self–supporting conflict–free in ES and thus pd–acyclic conflict–free in D^{ES}. An argument $a \in A$ is acceptable w.r.t. S in ES iff it is decisively in w.r.t. v_S^a in D^{ES}.*

With these partial results we can finally state the relation between EAS and ADF semantics:

Theorem 11.21. *Let $ES = (A, R, E)$ be a strongly consistent EAS and $D^{ES} = (A, L, C)$ its corresponding ADF obtained through Translation 80. A set of arguments $S \subseteq A$ is a σ–extension of ES, where $\sigma \in \{$admissible, preferred, complete$\}$ iff it is an aa–σ–extension of D^{ES}. S is stable in ES iff it is stable in D^{ES}. S is grounded in ES iff it is acyclic grounded in D^{ES}.*

We can now put our results into the classification system. Although a lot of properties of our EAS–ADF translation will be similar as in the AFN–ADF Translation 69, there are certain differences, most notably the loss of modularity.

Redefinition of Translation 80: *Let $SCons^{EAS}$ be the collection of all strongly consistent EASs and $BADF$ the collection of all bipolar ADFs, both on domain $\mathcal{U}$. The translation sc-$Tr^{EAS}_{ADF} : SCons^{EAS} \rightarrow BADF$ is defined as sc-$Tr^{EAS}_{ADF}((A, R, E)) = (A, L, C)$, where $L = \{(x, y) \mid \exists X \subseteq A, x \in X$ s.t. $(X, y) \in E \cup R\} \cup \{(x, x) \mid x \neq \eta, \nexists X \subseteq A$ s.t. $XEx\}$, and $C = \{C_a \mid a \in A\}$ is the set of acceptance conditions s.t. given the collection $B = \{B_1, ..., B_n\}$ of all subsets of A s.t. B_iRa and the collection $Z = \{Z_1, ..., Z_m\}$ of all subsets of A s.t. Z_iEa, the condition C_a is defined as a) $C_a(P') = out$ for $P' \subseteq par(a)$ iff $\exists X \subseteq P'$ s.t. XRa or ($Z \neq \emptyset$ and $\nexists Z \subseteq P'$ s.t. ZEa) or ($Z = \emptyset$ and $a \notin P'$), otherwise, $C_a(P') = in$; or b) $C_a = att_a \wedge sup_a$ where*

$$att_a = \begin{cases} \bigvee \neg B_1 \wedge ... \wedge \bigvee \neg B_n & \text{if } B \neq \emptyset \\ \top & \text{if } B = \emptyset \end{cases}$$

$$sup_a = \begin{cases} \bigwedge Z_1 \vee ... \vee \bigwedge Z_m & \text{if } Z \neq \emptyset \\ a & \text{if } Z = \emptyset \text{ and } a \neq \eta \\ \top & \text{otherwise} \end{cases}$$

Redefinition of Theorems 11.17, 11.18 and 11.21: *Let $\sigma^{EAS} \in \{$self–supporting, strongly self–supporting, admissible, preferred, complete, grounded, stable$\}$ be an EAS semantics and $\sigma^{ADF} \in \{$pd–acyclic, pd–acyclic conflict–free, aa–admissible, aa–preferred, aa–complete, acyclic grounded, stable$\}$ a similar ADF semantics. Let SC^{Tr}_{σ} the identity casting functions for σ. The translation sc-Tr^{EAS}_{ADF} is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$.*

Analysis of Translation 80: Under the (pd–acyclic) self–supporting, (pd–acyclic conflict–free) strongly self–supporting, (aa–) admissible, (aa–) preferred, (aa–) complete, (acyclic) grounded and stable semantics and identity casting functions, the translation sc-Tr^{EAS}_{ADF} is:

- source–subclass, target–subclass and overlapping
- argument domain preserving and relation introducing
- generic, semantics domain preserving and exact
- semi–structural

The translation sc-Tr^{EAS}_{ADF} is neither $\oplus$ nor $\otimes$–modular. We classify sc-Tr^{EAS}_{ADF} as basic under the listed semantics and casting functions.

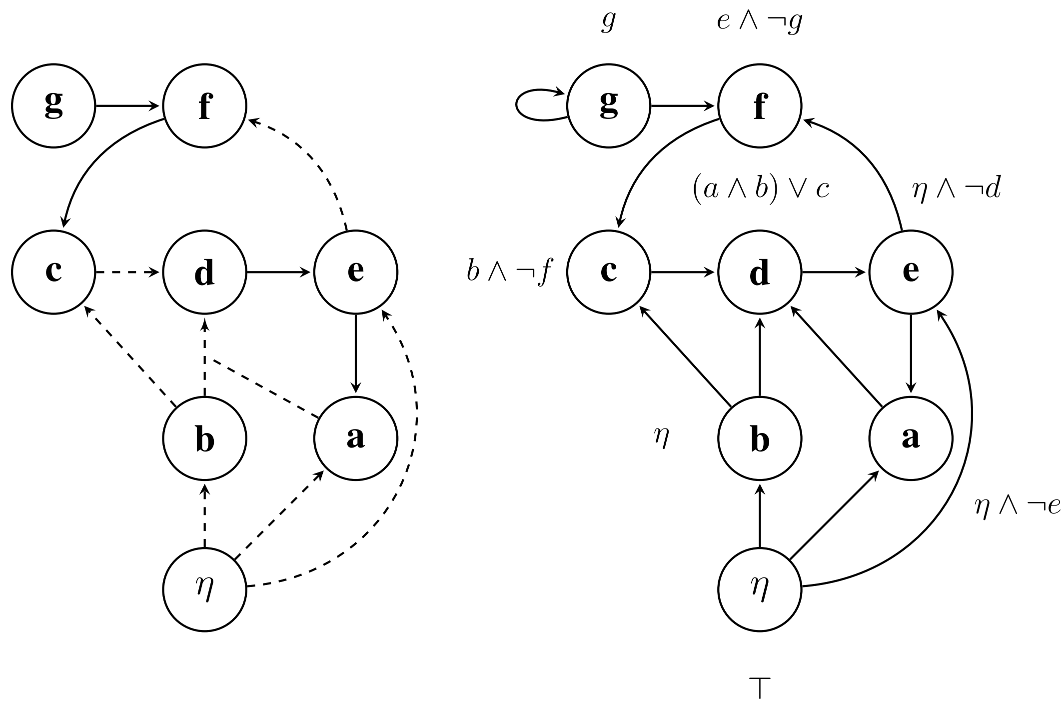
Explanation. Most of the explanations are the same as in Translation 69. There are only three differences between this and the mentioned approach. Translation sc-Tr^{EAS}_{ADF} is relation introducing – some modifications are done concerning the unsupported arguments. Due to this, we can also see the approach as semi–structural. Moreover, unlike the strongly consistent AFN–ADF approach, the EAS–ADF translation is not modular. We can repeat the analysis done in Translation 31 to show the loss of $\oplus$–modularity. Concerning $\otimes$–modularity, let us consider an argument a supported by a set $\{b\}$ in one framework and $\{c\}$ in the other. This produces two acceptance conditions, $C_1 = b$ and $C_1 = c$ respectively. Joining them with the $\otimes$ approach produces a condition $C_3 = b \wedge c$, while the formula associated with receiving support from sets $\{b\}$ and $\{c\}$ in a given EAS would be $b \vee c$. Hence, our approach is neither $\oplus$ nor $\otimes$–modular. In the future, this issue can be addressed by separating an acceptance condition and applying the $\otimes$–method to the attack part and $\oplus$–one to the support part. ■

Example 134. Let $ES = (\{\eta, a, b, c, d, e, f, g\}, \{(\{d\}, e), (\{e\}, a), (\{f\}, c), (\{g\}, f)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{b\}, c), (\{\eta\}, e), (\{a, b\}, d), (\{c\}, d), (\{e\}, f)\})$ be the EAS depicted in Figure 112a, previously analyzed in Example 132. Its associated ADF is $D^{ES} = (\{\eta, a, b, c, d, e, f, g\}, \{C_\eta = \top, C_a = \eta \wedge \neg e, C_b = \eta, C_c = b \wedge \neg f, C_d = (a \wedge b) \vee c, C_e = \eta \wedge \neg d, C_f = e \wedge \neg g, C_g = g\})$ [28], depicted in Figure 112b. We can observe that a self–supporting link is added to argument g due to its lack of an evidential sequence in ES.

The (minimal) acyclic evaluations for arguments in D^{ES} are $ev_\eta = ((\eta), \emptyset)$, $ev_a = ((\eta, a), \{e\})$, $ev_b = ((\eta, b), \emptyset)$, $ev_c = ((\eta, b, c), \{f\})$, $ev_d^1 = ((\eta, a, b, d), \{e\})$, $ev_d^2 = ((\eta, b, c, d), \{f\})$, $ev_e = ((\eta, e), \{d\})$ and $ev_f = ((\eta, e, f), \{d, g\})$. The argument g possesses no acyclic evaluation. We can therefore construct the following pd–acyclic conflict–free extensions: $E_1 = \emptyset$, $E_2 = \{\eta\}$, $E_3 = \{\eta, a\}$, $E_4 = \{\eta, b\}$, $E_5 = \{\eta, e\}$, $E_6 = \{\eta, a, b\}$, $E_7 = \{\eta, b, c\}$, $E_8 = \{\eta, b, e\}$, $E_9 = \{\eta, e, f\}$, $E_{10} = \{\eta, a, b, c\}$, $E_{11} = \{\eta, a, b, d\}$, $E_{12} = \{\eta, b, c, d\}$, $E_{13} = \{\eta, b, c, e\}$, $E_{14} = \{\eta, b, e, f\}$ and $E_{15} = \{\eta, a, b, c, d\}$. Their acyclic discarded sets are respectively $E_1^{a+} = E_2^{a+} = E_3^{a+} = E_4^{a+} = E_6^{a+} = E_7^{a+} = E_{10}^{a+} = \{g\}$, $E_5^{a+} = E_8^{a+} = E_{13}^{a+} = \{a, g\}$, $E_9^{a+} = E_{14}^{a+} = \{a, c, d, g\}$ and $E_{11}^{a+} = E_{12}^{a+} = E_{15}^{a+} = \{e, f, g\}$.

Using the above information, we can see that E_3, E_6 and E_{10} are not aa–admissible; the argument a is not decisively in w.r.t. their acyclic range interpretations. Similarly, e is not decisively in w.r.t. the ranges of E_5, E_8 and E_{13}, while c is not decisively in w.r.t. the ranges of E_7 and E_{13}. This leaves E_1, E_2, E_4, E_9, E_{11}, E_{12}, E_{14} and E_{15} as the aa–admissible extensions of D^{ES}. We can observe that exactly the same sets are admissible in ES. To the set E_1 we can add η; thus, this extension is not aa–complete in D^{ES}. To E_2 and E_9 we can introduce b, which disqualifies these sets as well. Finally, c is decisively in w.r.t. the acyclic range of E_{11} and a w.r.t. the range of E_{12}. Hence, E_4, E_{14} and E_{15} are left as the aa–complete extensions of D^{ES}. This is again in agreement with the extensions of ES. The acyclic grounded extension of D^{ES} is E_4, while E_{14} and E_{15} are aa–preferred in D^{ES}. This corresponds to the grounded and preferred sets of ES. Finally, by looking at the discarded sets, we can see that E_{14} and E_{15} are stable in D^{ES}, which is the desired answer.

[28] Please note we ignore explicitly stating the link set in this case; the connections can be seen in the associated figure.



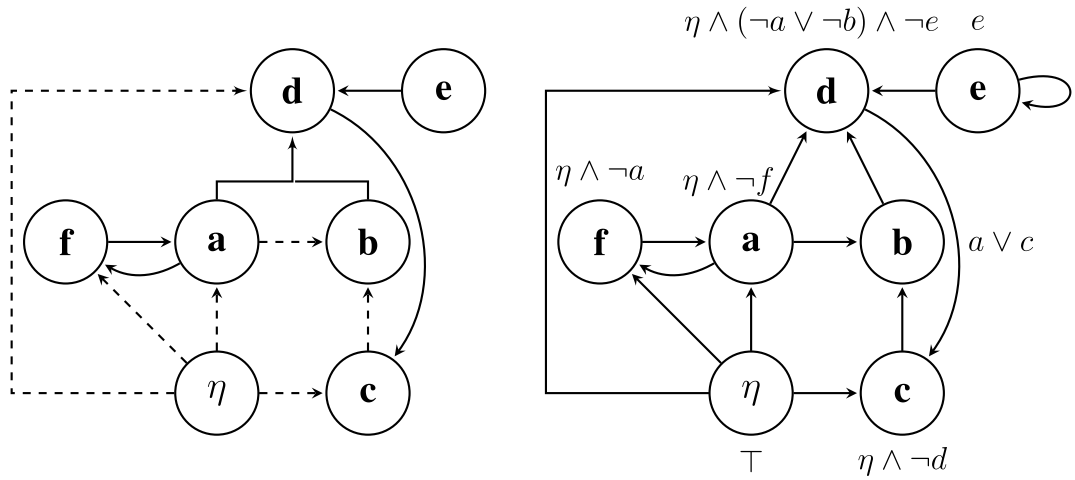
(a) Sample EAS (b) Associated ADF

Figure 112: Sample EAS and its associated ADF

Example 135. Let us consider the EAS $ES = (\{\eta, a, b, c, d, e, f\}, \{(\{a, b\}, d), (\{a\}, f), (\{d\}, c), (\{e\}, d), (\{f\}, a)\}, \{(\{\eta\}, a), (\{\eta\}, c), (\{\eta\}, d), (\{\eta\}, f), (\{a\}, b), (\{c\}, b)\})$ previously analyzed in Example 130 and now depicted in Figure 113a. Its associated ADF is $D^{ES} = (\{\eta, a, b, c, d, e, f\}, \{C_\eta = \top, C_a = \eta \wedge \neg f, C_b = a \vee c, C_c = \eta \wedge \neg d, C_d = \eta \wedge (\neg a \vee \neg b) \wedge \neg e, C_e = e, C_f = \eta \wedge \neg a\})$, which is visible in Figure 113b. The aa–admissible extensions of this framework are $\emptyset$, $\{\eta\}$, $\{\eta, a\}$, $\{\eta, f\}$, $\{\eta, a, b\}$, $\{\eta, d, f\}$ and $\{\eta, a, b, c\}$, which are also the admissible extensions of ES. The aa–complete sets are $\{\eta\}$, $\{\eta, d, f\}$ and $\{\eta, a, b, c\}$, which is again the correct answer. The set $\{eta\}$ is the acyclic grounded extension of D^{ES}, while $\{\eta, d, f\}$ and $\{\eta, a, b, c\}$ are aa–preferred and stable. Thus, we retrieve all and only desired extensions of ES.

11.5.2 General EASs

Just like in the AFN case (see Translation 70), in order to translate into ADFs those EASs that do not necessarily meet the strong consistency requirements, we will make use of the normal form transformations. In other words, we bring a given EAS into a strong consistency form using either Translation 14 or 15, and then proceed with the previously introduced Translation 80 to obtain the final result.



(a) Sample EAS

(b) Associated ADF

Figure 113: Sample EAS and its associated ADF

Translation 81. *Let $ES = (A, R, N)$ be an EAS. The corresponding ADF $D^{ES} = (A', L, C)$ is created as follows:*

1. *we obtain the strongly consistent EAS $ES^{sc} = (A', R, N')$ corresponding to ES by Translation 14 or 16, and*
2. *D^{ES} results from transforming ES^{sc} into ADF by Translation 80.*

Thanks to Theorems 4.55 and 4.57, the normal form theorem (Theorem 11.16) is true also for EASs that are not necessarily strongly consistent, independently of whether we use Translation 14 or 16 in order to construct our target ADF.

Let us now focus on the semantics. If we use Translation 14 in order to bring the source EAS into consistency form, then the relationship between the source and target extensions is defined by Theorems 4.54 and 11.21:

Theorem 11.22. *Let $ES = (A, R, E)$ be an EAS and $D^{ES} = (A', L, C)$ its corresponding ADF obtained by Translations 14 and 81. Let $S \subseteq A$, $S' \subseteq A'$ be sets of arguments and S^b the (possibly empty) set of bypass arguments generated by S in A'. The following holds:*

- *if S is self–supporting in ES, then $S \cup S^b$ is pd–acyclic in D^{ES}.*
- *if S is strongly self–supporting in ES, then $S \cup S^b$ is pd–acyclic conflict–free in D^{ES}.*
- *if S is a σ–extension of ES, where $\sigma \in \{$admissible, preferred, complete$\}$, then $S \cup S^b$ is an aa–σ–extension of D^{ES}.*
- *if S is grounded in ES, then $S \cup S^b$ is acyclic grounded in D^{ES}.*

- *if S is stable in ES, then $S \cup S^b$ is stable in D^{ES}.*
- *if S is pd–acyclic in D^{ES}, then $S' \cap A$ is self–supporting in ES.*
- *if S is pd–acyclic conflict–free in D^{ES}, then $S' \cap A$ is strongly self–supporting in ES.*
- *if S' is an aa–σ–extension of D^{ES}, then $S' \cap A$ is a σ–extension of ES.*
- *if S is acyclic grounded in D^{ES}, then $S' \cap A$ is grounded in ES.*
- *if S is stable in D^{ES}, then $S' \cap A$ is stable in ES.*

If we use Translation 16, then the following holds due to Theorems 4.58 and 11.21:

Theorem 11.23. *Let $ES = (A, R, E)$ be an EAS and $D^{ES} = (A', L, C)$ its corresponding ADF obtained by Translations 16 and 81. Let S^b the (possibly empty) set of bypass arguments generated by a set $S \subseteq A$ in A'. If a set of arguments S is self–supporting in ES, then $S \cup S^b$ is pd–acyclic in D^{ES}. If $S' \subseteq A'$ is pd–acyclic in D^{ES}, then $S' \cap A$ is self–supporting in ES. $S \subseteq A$ is strongly self–supporting in ES iff it is pd–acyclic conflict–free in D^{ES}. S is a σ–extension of ES, where $\sigma \in \{$admissible, complete, preferred$\}$ iff it is an aa–σ–extension of D^{ES}. S is grounded in ES iff it is acyclic grounded in D^{ES}. Every stable extension S of ES is stable in D^{ES} but not vice versa.*

We can now put the results into our system. The properties of the chained translations will come from the properties of their sub–translations and thus we will omit further analysis.

Redefinition of Translation 81: *Let Fr^{EAS} be the collection of all EASs on domain $\mathcal{U}$ and $BADF$ the collection of all bipolar ADFs on domain $\mathcal{U} \cup \mathcal{U}^b$. The translation b-$Tr^{EAS}_{ADF}$: $Fr^{EAS} \rightarrow BADF$ is defined as b-$Tr^{EAS}_{ADF}((A, R, E))$ = sc-Tr^{EAS}_{ADF}(bc-$Tr^{EAS}((A, R, E))$). The translation sa-Tr^{EAS}_{ADF} : $Fr^{EAS} \rightarrow BADF$ is defined as sa-$Tr^{EAS}_{ADF}((A, R, E))$ = sc-Tr^{EAS}_{ADF}(sa-$Tr^{EAS}((A, R, E))$).*

Redefinition of Theorem 11.22: *Let $\sigma^{EAS} \in \{$self–supporting, strongly self–supporting, admissible, preferred, complete, grounded, stable$\}$ be an EAS semantics and $\sigma^{ADF} \in \{$pd–acyclic, pd–acyclic conflict–free, aa–admissible, aa–preferred, aa–complete, acyclic grounded, stable$\}$ a similar ADF semantics. Let SC^{Tr}_{σ} the removal casting functions for σ defined as $SC^{X}_{\sigma}(S) = S \cap A$, where $X = (A, R, E) \in Fr^{EAS}$ and $S \in \sigma^{ADF}$(b-$Tr^{EAS}_{ADF}(X)$). The translation b-Tr^{EAS}_{ADF} is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is semantics bijective under the (aa–) complete, (aa–) preferred (acyclic) grounded and stable semantics and the defined casting functions.*

Redefinition of Theorem 11.23: *Let $\sigma^{EAS} \in \{$self–supporting, strongly self–supporting, admissible, preferred, complete, grounded$\}$ be an EAS semantics and $\sigma^{ADF} \in \{$pd–acyclic, pd–acyclic conflict–free, aa–admissible, aa–preferred, aa–complete, acyclic grounded$\}$ a similar ADF semantics. Let SC^{Tr}_{σ} be the identity casting functions for σ. The translation sa-Tr^{EAS}_{ADF} is strong under $(\sigma, SC^{Tr}_{\sigma})$. With the exception of self–supporting –*

pd–acyclic semantics, it is also semantics bijective. It is $\subseteq$–weak under the stable semantics and defined casting functions.

Analysis of Translation 81 with Translation 14 as intermediary: Under the (pd–acyclic) self–supporting, (pd–acyclic conflict–free) strongly self–supporting, (aa–) admissible, (aa–) preferred, (aa–) complete, (acyclic) grounded and stable semantics and identity casting functions, the translation $b\text{-}Tr^{EAS}_{ADF}$ is:

- full, target–subclass, overlapping
- weakly argument domain altering, argument introducing, relation introducing, relation removing
- generic and weakly semantics domain altering
- semi–structural

The translation is neither $\otimes$ nor $\oplus$–modular. It is faithful under the (aa–) complete, (aa–) preferred, (acyclic) grounded and stable semantics and the removal casting functions. The translation $b\text{-}Tr^{EAS}_{ADF}$ is classified as basic under the listed semantics and casting functions.

Analysis of Translation 81 with Translation 16 as intermediary: Under the (pd–acyclic) self–supporting, (pd–acyclic conflict–free) strongly self–supporting, (aa–) admissible, (aa–) preferred, (aa–) complete, (acyclic) grounded and stable semantics and identity casting functions, the translation $sa\text{-}Tr^{EAS}_{ADF}$ is:

- full, target–subclass, overlapping
- weakly argument domain altering, argument introducing, relation introducing, relation removing
- generic and semantics domain preserving
- semi–structural

The translation is neither $\otimes$ nor $\oplus$–modular. It is exact under the (pd–acyclic conflict–free) strongly self–supporting, (aa–) admissible, (aa–) complete, (aa–) preferred and (acyclic) grounded semantics and their identity casting functions. The translation $sa\text{-}Tr^{EAS}_{ADF}$ is classified as basic under the listed semantics and casting functions.

11.6 EAS as Other Frameworks

In this section we have presented the translations from EASs into AFs, SETAFs, BAFs, AFNs and ADFs. Our analysis has not included AFRAs and EAF(C)s. However, we do not believe that recursive attack is particularly suitable for modeling evidential support. As for the value of defense attacks, we refer the reader to Section 10.3, and propose to use chained translations for the time being.

11.7 Summary

The three (semantically speaking) strongest translations we have obtained are from EASs to BAFs, AFNs and ADFs. Unfortunately, in all these cases they were defined only for a subclass of evidential systems. The first one – and the most limited as well – worked for those frameworks that were strongly valid, attack binary and support singular. The next one assumed that the source frameworks were attack binary, which is a big improvement. Finally, the translation to ADFs permitted the use of the group attack, but had to assume strong consistency. Nevertheless, ADFs still emerge as a good target for EAS translations even if we work with the full framework domain, giving us, in total, five exact and one strong relation between the source and target extensions. The second best exact approach is the attack propagation SETAF method. Finally, concerning faithfulness, we again have another ADF translation and the full version of the AFN method. Our results can be seen in Table 14. What is worth observing is the fact that, similarly as for AFNs, the only two modular translations are also source–subclass, even though the results were created for different frameworks. These continuing difficulties show that modularity for support frameworks can be more challenging than for attack–based structures.

Table 14: Translations from EASs to other frameworks

Properties		AF	SETAF				BAF	AFN		ADF		
Translation		71	72	73	75	76	77	78	79	80	81 with 4.54	81 with 4.58
Strength	cf		$\subseteq$-weak	$\supseteq$-weak	strong	strong	exact	exact	$\subseteq$-weak			
	ssup						exact	exact	strong	exact	strong	strong
	str ssup	$\subseteq$-weak	strong	$\subseteq$-weak	$\subseteq$-weak	$\subseteq$-weak	exact	exact	$\subseteq$-weak	exact	strong	exact
	adm	strong	strong	$\subseteq$-weak	strong	strong	exact	exact	strong	exact	strong	exact
	comp	strong	strong	exact	faithful	faithful	exact	exact	faithful	exact	faithful	exact
	pref	strong	strong	exact	faithful	faithful	exact	exact	faithful	exact	faithful	exact
	grd	strong	strong	exact	faithful	faithful	exact	exact	faithful	exact	faithful	exact
	stb	strong	strong	exact	faithful	faithful	exact	exact	faithful	exact	faithful	$\subseteq$-weak
Functional		full	full	full	full	source–subclass	source–subclass	source–subclass	full	source–subclass	full	full
		target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass	target–subclass
		overlapping	overlapping	overlapping	overlapping	injective	injective	overlapping	overlapping	overlapping	overlapping	overlapping
Syntactical		argument domain altering	argument domain altering	argument domain preserving	weakly argument domain altering	weakly argument domain altering	argument domain preserving	argument domain preserving	argument domain altering	argument domain preserving	weakly argument domain altering	weakly argument domain altering
		argument removing	argument removing	argument removing	argument introducing	argument introducing	structure preserving	support introducing	argument introducing	relation introducing	argument introducing	argument introducing
		argument introducing	argument introducing	induced attack removing	induced attack introducing	induced attack introducing			support introducing		relation introducing	relation introducing
		attack removing	attack removing	attack introducing	support removing						relation removing	relation removing
		support removing	support removing	support removing								
Semantical		generic	generic	generic	generic	generic	generic	generic	generic	generic	generic	generic
		semantics domain altering	semantics domain altering	semantics domain preserving	weakly semantics domain altering	weakly semantics domain altering	semantics domain preserving	semantics domain preserving	semantics domain altering	semantics domain preserving	weakly semantics domain altering	semantics domain preserving
Computational		semantical	semantical	semantical	semantical	semi–structural	structural	semi–structural	semi–structural	semi–structural	semi–structural	semi–structural
						modular	modular					

12 Translating ADFs

We will now consider the last framework in our report - the abstract dialectical framework. Although we will consider only three target structures in this analysis, one has to bear in mind that ADFs have many families of semantics. The performed study needs to take into account the aa, ac, cc, ca_1/ca_2 and labeling–based approaches and due to the differences between them, each one of them needs a separate translation. We will primarily focus on creating approaches targeted at AFs, since due to the simplicity of these frameworks, the obtained results can be seen as the baseline that we can try to improve upon in the future. We will also present two SETAF transformations, which, chronologically speaking, precede the AF methods. In one of them, specialized for the aa–semantics, we obtain our first and only exact results for ADFs. Although the provided approach works only with weakly valid ADFs, it can be easily generalized by the use of Translation 9. Finally, we also introduce one ADF–AFN translation, also aimed at aa–semantics. The created method is not stronger than the ADF–AF one, however, it is also the only approach we have not managed to fit into our classification system.

12.1 ADF as AF

We will now present the approaches towards translating ADFs in AFs. They will follow the coalition approach, but with a slightly more advanced argument structure than before. The target arguments will now be constructed from evaluations, not e.g. pd–acyclic sets. Although the blocking set part of an evaluation does not impact a given extension, i.e. the extensions are built from pd–set and pd–sequences, it is important for the construction of the target framework. A given pd–set (sequence) is not unique to a given evaluation and aside for serving as a basis for conflict–derivation, preserving both pd–sets and blocking sets makes sure that the arguments representing different evaluations remain distinguishable. Two different evaluations that contribute the same arguments to a given extension can be attacked by different arguments and accidentally removing or merging them can affect both conflict–freeness and defense (decisiveness w.r.t. range). We can consider a rather trivial example of a SETAF–style ADF, i.e. one in which the acceptance conditions represent group attacks as in SETAFs (see Section 4.5.7):

Example 136. Let $(\{a, b, c\}, \{C_a = \neg c, C_b = \top, C_c = \neg a \vee \neg b\})$ be a simple SETAF–style ADF. We construct the following evaluations (minimal for every argument): $Ev^a = (\{a\}, \{c\})$, $Ev^b = (\{b\}, \emptyset)$, $Ev^c_1 = (\{c\}, \{a\})$ and $Ev^c_2 = (\{c\}, \{b\})$. The conflicts between them correspond to evaluation blocking, i.e. Ev^a attacks Ev^c_1, Ev^b attacks Ev^c_2, and both Ev^c_1 and Ev^c_2 attack Ev^a. We can observe that the two evaluations for c differ only by the blocking set. Let us now consider the AF constructed from the evaluations and the attacks between them. We can observe that $\{Ev^b, Ev^c_1\}$ (corresponding to $\{b, c\}$) is both a conflict–free and an admissible extension of our framework, which is the desired result.

We can now try to modify our approach and bring it down to the standard coalition method. By focusing only on the pd–sets of our evaluations, we obtain three arguments - $\{a\}$, $\{b\}$ and $\{c\}$, where $\{a\}$ and $\{b\}$ attack $\{c\}$ and $\{c\}$ attacks $\{a\}$. It is easy to see that $\{b, c\}$ is not even a conflict–free extension, let alone admissible. Thus, the data loss caused by taking only part of an evaluation into account "breaks" the behavior of the translation.

For this reason, even though the translations we are about to introduce will follow the coalition method, the used casting functions will have to be extractions, not just unions. Due to the number of semantics in ADFs, we will present more than just one transformation. We will start with the easiest ones, targeted at the aa and ac–semantics, and then continue with the cc–family, which requires the use of additional auxiliary arguments. Then we will remark on the issues concerning the ca_1 and ca_2 families, echoing the difficulties seen in the EAF translations (see Section 8.2), and close our analysis with a transformation for the labeling–based semantics.

12.1.1 AA Semantics

The translation for aa–semantics is the simplest one in ADFs. The same types of arguments are valid both on the inside and on the outside. Moreover, the acyclicity assumption grants us another very important property. The acceptance condition of the starting a_0 argument in every evaluation is in w.r.t. any type of a range of $\emptyset$. This means that defending an evaluation causes a chain reaction leading to decisiveness of the arguments inside the evaluation. As we will see in Section 12.1.3, this is not always the case for partially acyclic (and thus standard) evaluations and requires the use of auxiliary arguments.

Let us now present the translation which follows the idea we have presented before. We create new arguments from (acyclic) evaluations in the original framework and the attack relation is derived from the blocking set. In order to prevent the excessive creation of arguments, we will use minimal evaluations (see Definition 2.122).

Translation 82. *Let $D = (A, C)$ be an ADF. Its corresponding AF $F^D_{AA} = (A', R)$ is built the following way:*

- *$A' = \{(F, B) \mid (F, B)$ is a minimal acyclic positive dependency evaluations for an argument $a \in A$ on $A\}$, and*
- *an evaluation (F, B) attacks (F', B') iff $B' \cap F \neq \emptyset$.*

Fortunately, there is a strong relation between the source and target extensions. It is worth noting it is such even when we consider SETAF–style ADFs, even though the coalition–style SETAF–AF Translation 25 was weak under the conflict–free semantics. This shows another advantage of using evaluations instead of sets of arguments:

Theorem 12.1. *Let $D = (A, C)$ be an ADF and F^D_{AA} its corresponding AF obtained from Translation 82. If $S \subseteq A$ is a pd–acyclic conflict–free (aa–admissible, aa–complete, aa–preferred, stable, acyclic grounded) extension of D, then there exists a conflict–free (admissible, complete, preferred, stable, grounded) extension $S' = \{(F_1, B_1), ..., (F_n, B_n)\} \subseteq A'$ of F^D_{AA} s.t. $S = \bigcup_{i=1}^{n} F_i$. If $S' = \{(F_1, B_1), ..., (F_n, B_n)\} \subseteq A'$ is a conflict–free (admissible, complete, preferred, stable, grounded) extension of F^D_{AA}, then $S = \bigcup_{i=1}^{n} F_i$ is pd–acyclic conflict–free (aa–admissible, aa–complete, aa–preferred, stable, acyclic grounded) extension of D.*

We can now redefine our results and put them into our system. We can observe that the presented casting function will be in fact a two–step one; first, we perform extraction on the evaluation arguments to get the pd–sequences. We then use the union casting approach to merge the obtain collections into a standard, argument–based extension of the original ADF.

Redefinition of Translation 82: *Let Fr^{ADF} be the collection of all ADFs on domain $\mathcal{U}$ and Fr^{AF} the collection of all AFs on domain $AEV^{\mathcal{U}} = \{(F, B) \mid F$ is a nonempty sequence on $E \subseteq \mathcal{U}$, $B \subseteq \mathcal{U}\}$. The translation $aa\text{-}Tr_{AF}^{ADF} : Fr^{ADF} \to Fr^{AF}$ is defined as $aa\text{-}Tr_{AF}^{ADF}((A, C)) = (A', R)$ for a framework $(A, C) \in Fr^{ADF}$, where $A' = \{(F, B) \mid (F, B)$ is a minimal acyclic positive dependency evaluation for an argument $a \in A$ on $A\}$ and $R = \{((F, B), (F', B') \mid F \cap B' \neq \emptyset\}$.*

Redefinition of Theorem 12.1: *Let $\sigma^{ADF} \in \{$pd–acyclic conflict–free, aa–admissible, aa–complete, aa–preferred, stable, acyclic grounded$\}$ be an ADF semantics and $\sigma^{AF} \in \{$conflict–free, admissible, complete, preferred, stable, grounded$\}$ a similar AF semantics. Let SC_{σ}^{Tr} be the extraction–union hybrid semantics casting function for σ defined as $SC_{\sigma}^{X}(S) = \bigcup_{i=1}^{n} F_i$, where $X = (A, C) \in Fr^{ADF}$ and $S = (F_1, B_1), ..., (F_n, B_n) \in \sigma(aa\text{-}Tr_{AF}^{ADF}(X))$. The translation $aa\text{-}Tr_{AF}^{ADF}$ is strong under $(\sigma, SC_{\sigma}^{Tr})$. It is also semantics bijective under the (aa–) complete, (aa–) preferred, stable and (acyclic) grounded semantics and the defined casting functions.*

Analysis of Translation 82: Under the (pd–acyclic) conflict–free, (aa–) admissible, (aa–) complete, (aa–) preferred, (acyclic) grounded and stable semantics and the defined casting functions, the translation $aa\text{-}Tr_{AF}^{ADF}$ is:

- full, target–subclass and overlapping
- argument domain altering, argument removing, argument introducing, relation removing, (possibly) relation introducing
- generic and semantics domain altering
- semantical

The translation is neither $\oplus$ nor $\otimes$–modular. The translation $aa\text{-}Tr_{AF}^{ADF}$ is classified as coalition–style under the listed semantics and casting functions.

Explanation. Any ADF can undergo our translation, and thus we classify it as full. Structure–wise, any AF can be produced. Let $F = (A, R)$ be a Dung's framework and $D^F = (A, L, C)$ its corresponding ADF (see Translation 23). By translating D^F back to an AF using our translation, every original argument $a \in A$ in F is now (in this case, uniquely) associated with $((a), \{a\}^-)$, where $\{a\}^-$ represents the arguments attacking a. Thus, if we were to disregard the contents of the arguments and their naming, the original AF F and the one associated with D^F would be the same. However, our translation changes the domain, and not every possible target framework can be produced by our approach. First of all, given an evaluation argument and its pd–sequence, there have to be other evaluation arguments dedicated for the element in the sequence preceding the final one. For example, if $((a,b),\emptyset)$ is an evaluation argument, then so is $((a),\emptyset)$. Certain restrictions also arise from the minimality assumption in the construction. Finally, since the attack relation in the target AF is defined by the blocking sets, constructing a framework such as $(\{((a),\{b\}),((b),\emptyset)\},\emptyset)$ is not possible – the relevant conflicts are missing. For all these reasons we decide to classify our approach as target–subclass.

Due to the fact that only arguments possessing acyclic pd–evaluations are represented in the produced Dung's framework, it can happen that two different arguments are translated into a single framework. However, we need to note that while e.g. two AFNs or EASs connected by the weakly valid form would translate to the same structure, it is not necessarily the case in ADFs. This is due to the presence of the blocking set, which can contain arguments not appearing in the pd–sequences of any evaluation arguments. For example, the ADFs $D_1 = (\{a,b\},\{C_a = \neg b, C_b = b\})$ and $D_2 = (\{a,b\},\{C_a = \top, C_b = b\})$ would get translated to different AFs ($(\{((a),\{b\})\},\emptyset)$ and $(\{((a),\emptyset)\},\emptyset)$ respectively), even though they have the same weak validity form $D_3 = (\{a\},\{C_a = \top\})$. Nevertheless, the AF associated with D_3 would be the same as with D_2, and thus the translation is overlapping.

We can observe that the argument domain clearly undergoes a drastic change. For the reason mentioned above, the translation is argument removing – not in all cases the arguments not meeting validity requirements will appear in the target framework. It is also argument introducing due to the fact that a single argument can be represented by multiple evaluations, we classify this approach as argument introducing. Although argument removal leads to relation removal on its own, separate removal and (possibly) addition can occur due to the imprecision of evaluation arguments (see e.g. the analysis of Translations 57 and 61).

The changes done to the argument domain affect (in this case) the semantics domain as well. The amount of handled semantics classifies our approach as generic. The fact that we are using acyclic evaluations qualifies our approach as semantical, similarly as it was in the case of Translations 61 and 71. Unfortunately, just like these approaches, the $aa\text{-}Tr_{AF}^{ADF}$ translation is not modular. To show this, we can take examples where performing an $\oplus$ or $\otimes$ joining of two ADFs changes the validity status of given arguments. Let $D_1 = (\{a,b\},\{C_a = b, C_b = \top\})$ and $D_2 = (\{a,b\},\{C_a = \top, C_b = a\})$ be two ADFs. In both of them, arguments a and b possess acyclic evaluations that will appear in the target AFs (and thus in their union as well). However, $D_1 \otimes D_2 = (\{a,b\},\{C_a = b, C_b = a\})$ and neither a nor b have acyclic evaluations. Thus, the AF associated with this structure is basically empty. Therefore, our translation is not $\otimes$–modular. Let us now consider the ADFs $D_3 = (\{a\},\{C_a = a\})$ and $D_4 = (\{a\},\{C_a = \neg a\})$. The AF corresponding to D_3 is empty, while the one associated with D_4 is $(\{Ev_a\},\{(Ev_a,Ev_a)\})$, where $Ev_a = ((a),\{a\})$. However, $D_3 \oplus D_4$ is basically $(\{a\},\{C_a = \top\})$. The corresponding AF is $(\{Ev'_a\},\emptyset)$, where $Ev'_a = ((a),\emptyset)$, which is quite different from $(\{Ev_a\},\{(Ev_a,Ev_a)\})$. Therefore, our translation is neither $\oplus$ nor $\otimes$–modular. ■

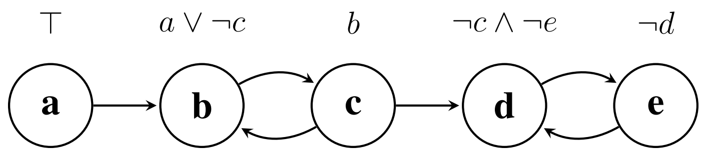
Example 137. Let us consider the framework $D = (\{a, b, c, d, e\}, \{C_a = \top, C_b = a \vee \neg c, C_c = b, C_d = \neg c \wedge \neg e, C_e = \neg d\})$ depicted in Figure 114a. The minimal decisively in interpretations for its arguments are $v_a = \emptyset$, $v_b^1 = \{a : \mathbf{t}\}$, $v_b^2 = \{c : \mathbf{f}\}$, $v_c = \{b : \mathbf{t}\}$, $v_d = \{c : \mathbf{f}, e : \mathbf{f}\}$, $v_e = \{d : \mathbf{f}\}$. Using them, we can produce the following minimal acyclic evaluations for our arguments: $ev_a = ((a), \emptyset)$ for a, $ev_b^1 = ((a, b), \emptyset)$ and $ev_b^2 = ((b), \{c\})$ for b, $ev_c^1 = ((a, b, c), \emptyset)$ and $ev_c^2 = ((b, c), \{c\})$ for c, $ev_d = ((d), \{c, e\})$ for d and finally $ev_e = ((e), \{d\})$ for e.

With this information, we can create the following pd–acyclic conflict–free extensions: $E_1 = \emptyset$, $E_2 = \{a\}$, $E_3 = \{b\}$, $E_4 = \{d\}$, $E_5 = \{e\}$, $E_6 = \{a, b\}$, $E_7 = \{a, d\}$, $E_8 = \{a, e\}$, $E_9 = \{b, d\}$, $E_{10} = \{b, e\}$, $E_{11} = \{a, b, c\}$, $E_{12} = \{a, b, d\}$, $E_{13} = \{a, b, e\}$ and $E_{14} = \{a, b, c, e\}$. Their associated acyclic discarded sets are $E_1^{a+} = E_2^{a+} = E_3^{a+} = E_6^{a+} = \emptyset$, $E_4^{a+} = E_7^{a+} = E_9^{a+} = E_{12}^{a+} = \{e\}$, and $E_5^{a+} = E_8^{a+} = E_{10}^{a+} = E_{11}^{a+} = E_{13}^{a+} = E_{14}^{a+} = \{d\}$. Therefore, we can show that E_1, E_2, E_5, E_6, E_8, E_{11}, E_{13} and E_{14} are aa–admissible in D. We can observe that only $E_{14} = \{a, b, c, e\}$ is aa–complete. The argument a is decisively in w.r.t. the acyclic ranges of E_1 and E_5, but is not accepted in these sets. In case of E_2 and E_8, the issue is with argument b, while for E_6 and E_{13} it is the absence of c that causes the loss of completeness. Finally, since d is in the acyclic discarded set of E_{11}, argument e should have been accepted and we need to discard this extension as well. It is easy to see that $E_{14} = \{a, b, c, e\}$ is the unique acyclic grounded, aa–preferred and stable extension of D.

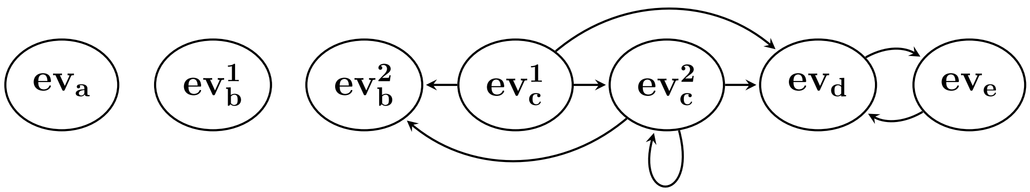
By looking at the evaluations and their blocking sets, we can now create the AF $F^D = (\{ev_a, ev_b^1, ev_b^2, ev_c^1, ev_c^2, ev_d, ev_e\}, \{(ev_c^1, ev_b^2), (ev_c^1, ev_c^2), (ev_c^1, ev_d), (ev_c^2, ev_b^2), (ev_c^2, ev_c^2), (ev_c^2, ev_d), (ev_d, ev_e), (ev_e, ev_d)\})$ that is associated with our ADF and is specialized for the aa–semantics. We can see it depicted in Figure 114b. There are in total 32 conflict–free extensions of this framework; therefore, we will start our analysis with the admissible semantics. The sets $\emptyset$, $\{ev_a\}$, $\{ev_b^1\}$, $\{ev_c^1\}$, $\{ev_e\}$, $\{ev_a, ev_b^1\}$, $\{ev_a, ev_c^1\}$, $\{ev_a, ev_e\}$, $\{ev_b^1, ev_c^1\}$, $\{ev_b^1, ev_e\}$, $\{ev_c^1, ev_e\}$, $\{ev_a, ev_b^1, ev_c^1\}$, $\{ev_a, ev_b^1, ev_e\}$, $\{ev_a, ev_c^1, ev_e\}$, $\{ev_b^1, ev_c^1, ev_e\}$ and $\{ev_a, ev_b^1, ev_c^1, ev_e\}$ are the admissible extensions of F^D. They correspond to the sets $\emptyset$, $\{a\}$, $\{a, b\}$, $\{a, b, c\}$, $\{e\}$, $\{a, e\}$, $\{a, b, e\}$ and $\{a, b, c, e\}$. We can therefore observe that all and only aa–admissible extensions of D are retrieved, though a given ADF extension may be produced by more than a single AF one. We can observe that evaluation arguments ev_a, ev_b^1 and ev_c^1 are unattacked in F^D and will therefore be contained in any complete extension. ev_c^1 additionally defends ev_e. All of the remaining arguments are attacked by our collection; we thus obtain the single grounded, complete, preferred and stable extension $\{ev_a, ev_b^1, ev_c^1, ev_e\}$ of F^D. It corresponds to the set $\{a, b, c, e\}$, which is the acyclic grounded, aa–complete, aa–preferred and stable extension of D. We thus retrieve all and only desired extensions of our ADF.

12.1.2 AC Semantics

The next translation we will consider is dedicated to the ac–family of semantics. Previously, only acyclic evaluations were taken into account both on the “inside” and “outside”.



(a) Sample ADF



(b) Associated AF for aa–semantics

Figure 114: Sample ADF and its associated AF for aa–semantics

Due to the fact that now the "outside" conditions are less specialized, all types of evaluations need to appear in the framework. However, due to the "inside" requirements, we do not accept an argument representing an evaluation that cannot be made acyclic. The "can attack, but cannot be accepted" restriction in the Dung's framework is easily addressed by the use of self–attackers. Thus, the non–acyclic evaluations are turned into such in our translation:

Translation 83. *Let $D = (A, C)$ be an ADF. The corresponding AF $F^D_{AC} = (A', R)$ is built the following way:*

- *let A^a be the set of all minimal acyclic positive dependency evaluations for all arguments $a \in A$ on A,*
- *let A^c be the set of all minimal standard evaluations for all $a \in A$ on A s.t. they cannot be made acyclic for a w.r.t. the pd–function they were created with,*
- *$A' = A^a \cup A^c$,*
- *an evaluation (E, B) in A' attacks (E', B') in A' if $B' \cap E \neq \emptyset$, and*
- *for every evaluation (E, B) in A^c, (E, B) attacks itself.*

We can now focus on the semantics. It can be observed that apart from the prefix change on the admissible, complete and preferred semantics, we now consider the (standard) grounded extensions rather than acyclic ones. Moreover, we do not consider the stable semantics anymore. As seen in Section 2.3.6, it is not part of the ac–family, and neither is the model semantics.

Theorem 12.2. *Let $D = (A, C)$ be an ADF and F^D_{AC} its corresponding AF obtained from Translation 83. If $S \subseteq A$ is a pd–acyclic conflict–free (ac–admissible, ac–complete, ac–preferred, grounded) extension of D, then there exists a conflict–free (admissible, complete, preferred, grounded) extension $S' = \{(E_1, B_1), ..., (E_n, B_n)\} \subseteq A'$ of F^D_{AC} s.t. $S = \bigcup_{i=1}^n E_i$. If $S' = \{(E_1, B_1), ..., (E_n, B_n)\} \subseteq A'$ is a conflict–free (admissible, complete, preferred, grounded) extension of F^D_{AC}, then $S = \bigcup_{i=1}^n E_i$ is pd–acyclic conflict–free (ac–admissible, ac–complete, ac–preferred, grounded) extension of D.*

We can now put the ac–translation into our system. We can observe that the properties of this approach will be almost the same as in the aa–version. Thus, most of the explanations will be omitted.

Redefinition of Translation 83: *Let Fr^{ADF} be the collection of all ADFs on domain $\mathcal{U}$ and Fr^{AF} the collection of all AFs on domain $AEV^{\mathcal{U}} \cup SEV^{\mathcal{U}}$, where $AEV^{\mathcal{U}} = \{(F, B) \mid F$ is a nonempty sequence on $E \subseteq \mathcal{U}$, $B \subseteq \mathcal{U}\}$ and $SEV^{\mathcal{U}} = \{(F, B) \mid F \subseteq \mathcal{U}, F \neq \emptyset, B \subseteq \mathcal{U}\}$. The translation ac-$Tr^{ADF}_{AF} : Fr^{ADF} \rightarrow Fr^{AF}$ is defined as ac-$Tr^{ADF}_{AF}((A, C)) = (A', R)$ for a framework $(A, C) \in Fr^{ADF}$, where $A' = A^a \cup A^c$ s.t. $A^a = \{(F, B) \mid (F, B)$ is a minimal acyclic positive dependency evaluation for an argument $a \in A\}$ and $A^c = \{(F, B) \mid (F, B)$ is a minimal standard evaluation for an argument $a \in A$ and cannot be made acyclic for any $b \in F\}$, and $R = \{((F, B), (F', B') \mid F \cap B' \neq \emptyset\} \cup \{((F, B), (F, B) \mid (F, B) \in A^c\}$.*

Redefinition of Theorem 12.2: *Let $\sigma^{ADF} \in \{$pd–acyclic conflict–free, ac–admissible, ac–complete, ac–preferred, grounded$\}$ be an ADF semantics and $\sigma^{AF} \in \{$conflict–free, admissible, complete, preferred, grounded$\}$ a similar AF semantics. Let SC^{Tr}_{σ} be the extraction–union hybrid semantics casting function for σ defined as $SC^X_{\sigma}(S) = \bigcup_{i=1}^n F_i$, where $X = (A, C) \in Fr^{ADF}$ and $S = (F_1, B_1), ..., (F_n, B_n) \in \sigma($ac-$Tr^{ADF}_{AF}(X))$. The translation ac-$Tr^{ADF}_{AF}$ is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is also semantics bijective under the (ac–) complete, (ac–) preferred and grounded semantics and the defined casting functions.*

Analysis of Translation 83: Under the (pd–acyclic) conflict–free, (ac–) admissible, (ac–) complete, (ac–) preferred and grounded semantics and the defined casting functions, the translation $ac\text{-}Tr^{ADF}_{AF}$ is:

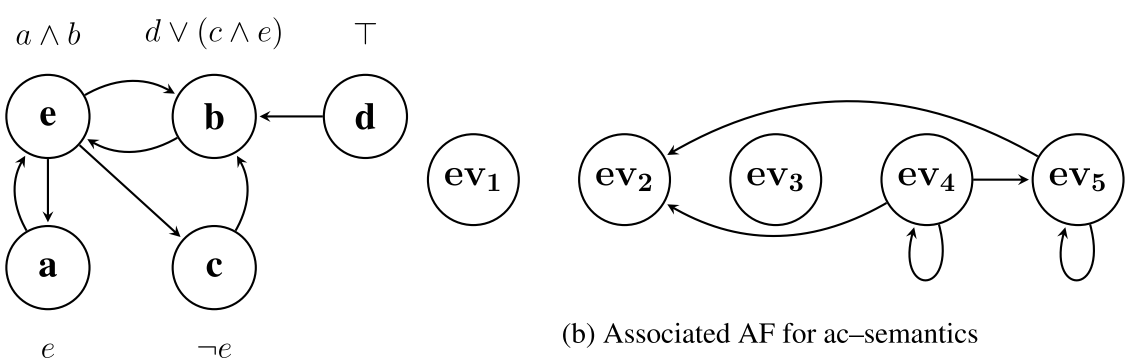
- full, target–subclass and overlapping
- argument domain altering, argument removing, argument introducing, relation removing, relation introducing
- generic and semantics domain altering
- semantical

The translation is neither $\oplus$ nor $\otimes$–modular. The translation $ac\text{-}Tr^{ADF}_{AF}$ is classified as coalition–style under the listed semantics and casting functions.

Explanation. Similarly as in Translation 82, our approach is full and target–subclass. It is also overlapping, though in this case we would repeat the analysis for cleansed form rather than for weakly valid form (see the analysis of Translation 82). Due to the fact that arguments not possessing any standard evaluations might not appear in the target framework, we classify our approach as argument removing. It is also argument introducing, as every ADF argument can be represented by multiple AF arguments. The removal of arguments leads to the removal of relations. Further relation removal (and possibly addition) can occur due to the imprecision of evaluation arguments (see e.g. the analysis of Translations 57 and 61). However, as the AF arguments in A^c are self–attacking even though the evaluations they represent might not necessarily be self–blocking, our translation is also relation introducing without any doubts as was in the case of Translation 82.

Let us now focus on modularity. Let $D_1 = (\{a, b\}, \{C_a = b, C_b = \top\})$ and $D_2 = (\{a, b\}, \{C_a = \top, C_b = a\})$ be two ADFs just like in the analysis of Translation 82. The union of their associated AFs is $(\{((a), \emptyset), ((a, b)), \emptyset), ((b), \emptyset), ((b, a)), \emptyset)\}, \emptyset)$. However, the AF corresponding to $D_1 \otimes D_2 = (\{a, b\}, \{C_a = b, C_b = a\})$ is $(\{(\{a, b\}, \emptyset)\}, \{((\{a, b\}, \emptyset), (\{a, b\}, \emptyset))\})$. Therefore, our translation is not $\otimes$–modular. Let us now consider the ADFs $D_3 = (\{a\}, \{C_a = a\})$ and $D_4 = (\{a\}, \{C_a = \neg a\})$. The AF corresponding to D_3 is $(\{(\{a\}, \emptyset)\}, \{((\{a\}, \emptyset), (\{a\}, \emptyset))\})$, while the one associated with D_4 is $(\{((a), \{a\})\}, \{(((a), \{a\}), ((a), \{a\}))\})$. If we were to join these frameworks and perform a translation on $D_3 \oplus D_4 = (\{a\}, \{C_a = \top\})$, we would obtain the framework $(\{((a), \emptyset)\}, \emptyset)$, which e.g. does not contain any elements from the AF created for D_4.

Further properties of the current method can be explained similarly as for Translation 82. ■



(a) Sample ADF

(b) Associated AF for ac–semantics

Figure 115: Sample ADF and its associated AF for ac–semantics

Example 138. Let us look at the ADF $D = (\{a, b, c, d, e\}, \{C_a = e, C_b = d \vee (c \wedge e), C_c = \neg e, C_d = \top, C_e = a \wedge b\})$ depicted in Figure 115a and previously analyzed in Example 28. As already explained before, the sets $\emptyset$, $\{c\}$, $\{d\}$, $\{b, d\}$, $\{c, d\}$ and $\{b, c, d\}$ are the pd–acyclic conflict–free extensions of D. From this, $\emptyset$, $\{d\}$ and $\{b, d\}$ are ac–admissible, with the last set being also the single ac–complete, ac–preferred and grounded extension.

The minimal decisively in interpretations in this framework are $v_a = \{e : \mathbf{t}\}$, $v_b^1 = \{d : \mathbf{t}\}$, $v_b^2 = \{c : \mathbf{t}, e : \mathbf{t}\}$, $v_c = \{e : \mathbf{f}\}$, $v_d = \emptyset$ and $v_e = \{a : \mathbf{t}\, b : \mathbf{t}\}$. With these interpretations, we can create the minimal acyclic evaluations $ev_1 = ((d, b), \emptyset)$, $ev_2 = ((c), \{e\})$ and $ev_3 = ((d), \emptyset)$ for arguments b, c and d respectively. However, we can also create standard evaluations that cannot be made acyclic; we have $ev_4 = (\{a, b, d, e\}, \emptyset)$ for a and e, and $ev_5 = (\{a, b, c, e\}, \{e\})$ for a, b and e.

Using the created evaluations and their types, we can now create the AF $F^D = (\{ev_1, ev_2, ev_3, ev_4, ev_5\}, \{(ev_4, ev_2), (ev_4, ev_4), (ev_4, ev_5), (ev_5, ev_2), (ev_5, ev_5)\})$ associated with our ADF and specialized for the ac–semantics. We can see it depicted in Figure 115b. Its admissible extensions are $\emptyset$, $\{ev_1\}$, $\{ev_3\}$ and $\{ev_1, ev_3\}$. They correspond to the sets $\emptyset$, $\{b, d\}$ and $\{d\}$, which are the desired extensions. We can observe that $\{b, d\}$ can be obtained both from $\{ev_3\}$ and $\{ev_1, ev_3\}$. From this, the set $\{ev_1, ev_3\}$ is complete, preferred and grounded. Its associated ADF set is $\{b, d\}$, which was the original ac–complete, ac–preferred and grounded extension of D.

12.1.3 CC Semantics

We now come to the cc–family of semantics, where an argument possessing any type of an evaluation is "good enough" for acceptance and needs to be defended from (i.e. is not falsified in the range by "default"). Although this family is "homogeneous" the same way the aa–semantics are, the translations are quite different. In order to show where the problem is, we will first consider a cc–transformation in the spirit of the aa–method, i.e. instead of admitting only acyclic pd–evaluations, we would work on standard ones. We will then explain why this approach will not work as intended.

Translation 84. ***Faulty CC translation*** *Let $D = (A, C)$ be an ADF. Its corresponding AF $F^D_{CC} = (A', R)$ is built the following way:*

- *A' is the set of all minimal standard evaluations for an argument $a \in A$, and*
- *an evaluation (E, B) attacks (E', B') if $B' \cap E \neq \emptyset$.*

Although such an approach would preserve conflict–free, cc–admissible and cc–preferred semantics, it would not work for cc–complete and grounded approaches. Let us look at an example:

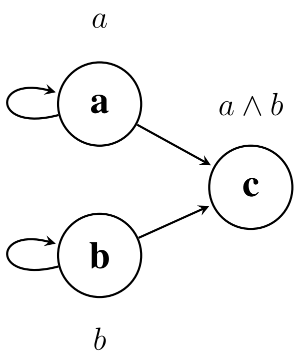
Example 139. Let $(\{a, b, c\}, \{C_a = b \wedge c, C_b = a, C_c = \mathbf{t}\}$ be a simple ADF. We can observe there is a cycle between a and b, i.e. one cannot be accepted without the other. Their (minimal) decisively in interpretations do not contain any $\mathbf{f}$ mappings. The two cc–complete extensions of our framework are $\{c\}$ and $\{a, b, c\}$. The corresponding AF is $(\{(\{c\}, \emptyset), (\{a, b, c\}, \emptyset)\}, \emptyset)$. We can observe there are no conflicts in this structure and thus both evaluation arguments are considered "initial". Thus, we only have a single complete extension, namely $\{(\{c\}, \emptyset), (\{a, b, c\}, \emptyset)\}$, which corresponds to $\{a, b, c\}$ in the original ADF. The existing conflicts are not enough to capture the positive dependency cycles in our ADF that cause the creation of two, not one cc–complete extension.

This difference in the produced extensions is a result of the fact that in the acyclic evaluations, at the start of an acyclic pd–evaluation is always an argument that has an empty positive part of a decisively in interpretation. Consequently, we only concern ourselves with the negative one – "attackers" – and proceed with building the evaluation. Thus, decisiveness of this starting argument was in correspondence with defense in the resulting AF. In a standard evaluation this might not be the case and thus decisiveness has to take into account both positive and negative parts, which makes defense derived from blocking insufficient.

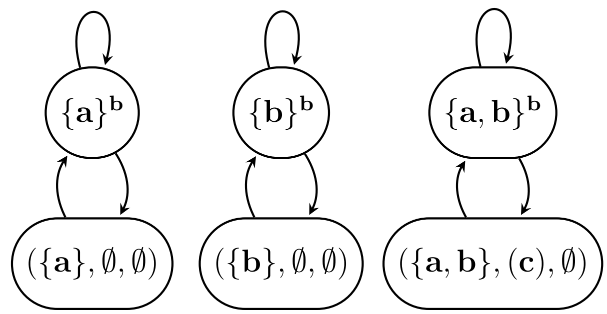
Our approach to address this issue is as follows: consider an argument a with a condition $C_a = a$. Its acceptance is self–dependent; for example, a is acceptable (decisively in) only w.r.t. sets that already contain it. A similar situation can be observed in case of the self–defending arguments in AFs: if we had arguments b and c mutually attacking each other, neither b nor c would be acceptable w.r.t. $\emptyset$. However, due to self–defense, both $\{b\}$ and $\{c\}$ were admissible sets. Consequently, our translation will make use of auxiliary "breaker" arguments attacking forcing self–defense of (parts of) evaluations that cannot be made acyclic. In a certain sense the breaker argument can be read as "you are unsupported", in a similar way as in the defender translations. However, there are certain differences in their purpose. Due to the fact that in the cc–semantics we cannot discard an argument just because it is not acyclic, the breaker arguments need to be self attackers – otherwise, they could appear in the extensions and serve as defenders against the evaluation arguments, which is against the design of the standard range. Moreover, unlike in the defender approaches, the breaker arguments might be related between each other or might be shared among arguments. Without this, the target AF can produce undesirable extensions:

Example 140. Let us consider an ADF $D_1 = (\{a, b, c\}, \{C_a = a, C_b = b, C_c = a \wedge b\})$ depicted in Figure 116a. We will represent our possible AF arguments with partially acyclic evaluations which show where the "cycles" are. We have one such evaluation per argument: $(\{a\}, \emptyset, \emptyset)$ for a, $(\{b\}, \emptyset, \emptyset)$ for b and $(\{a, b\}, (c), \emptyset)$ for c. The cc–complete extensions of D_1 are $\{a\}$, $\{b\}$ and $\{a, b, c\}$.

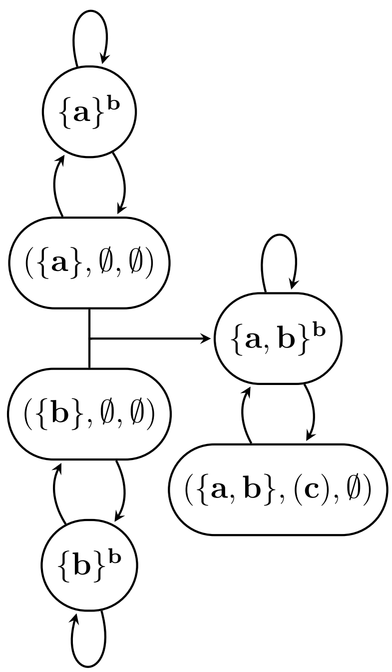
We will now consider different attempts at connecting breaker arguments to the evaluation arguments. Let us first look at the AF F_1 in Figure 116b. Every evaluation argument is attacked by the breaker argument associated with the pd–set of the evaluation. We can observe that the complete sets of F_1 correspond to $\{a\}$, $\{b\}$, $\{a, b\}$ and $\{a, b, c\}$. However, $\{a, b\}$ is not considered cc–complete in D_1 - this comes from the fact that the moment both a and b are present, we can accept c. This issue can now be addressed in several ways. First of all, we can consider adding a joint attack from $(\{a\}, \emptyset, \emptyset)$ and $(\{b\}, \emptyset, \emptyset)$ to the breaker of c, as done in SETAF SF_1 in Figure 116c. Although now the extensions are correct, we leave the domain of AFs and this is not the most desirable solution. Therefore, instead of allowing other arguments attack the breakers, we can let the breakers attack further elements. In F_2 presented in Figure 116d we remove the $\{a, b\}^b$ breaker attacking $(\{a, b\}, (c), \emptyset)$ and add the conflicts from $\{a\}^b$ and $\{b\}^b$ in its place. This only gives us the desired extensions, but brings us back to the domain of AFs.



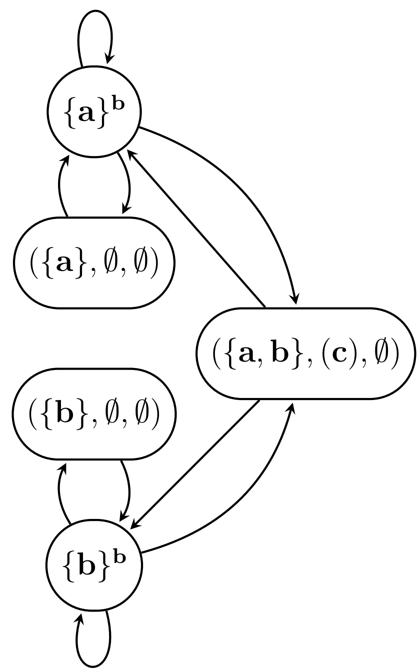
(a) D_1



(b) F_1



(c) SF_1



(d) F_2

Figure 116: Sample AFDF and handling of the breakers arguments

We are now ready to propose a translation from ADFs to AFs aimed at the cc–semantics. In what follows we will use the argument breakers approach from Figure 116d. In order to reduce the amount of required arguments, we will use minimal evaluations. However, please note that we are interested not in the minimality of an evaluation w.r.t. all of the possible evaluations, but only w.r.t. those that were created for the same argument, similarly as it was the case in Translation 61. Without this we would probably accidentally remove most of the arguments from the source framework.

Translation 85. *Let $D = (A, C)$ be an ADF. Its corresponding AF $F^D_{CC} = (A', R)$ is built* *the following way:*

- *let* $A^{ev} = \{(F, G, B) \mid (F, G, B)$ *is a minimal partially acyclic pd–evaluation for* $a \in A\}$,
- *let* $A^b = \{a^b \mid \exists (F, G, B) \in A^{ev}$ *s.t.* $a \in F\}$,
- $A' = A^{ev} \cup A^b$,
- *for every* $(F, G, B), (F', G', B') \in A^{ev}$, (F, G, B) *attacks* $(F,' G', B')$ *if* $(F \cup G) \cap B' \neq \emptyset$,
- *for every* $a^b \in A^b$, a^b *attacks* a^b,
- *for every* $(F, G, B) \in A^{ev}$, $a^b \in A^b$ *s.t.* $a \in F \cup G$, (F, G, B) *attacks* a^b, *and*
- *for every* $(F, G, B) \in A^{ev}$, $a^b \in A^b$ *s.t.* $a \in F$, a^b *attacks* (F, G, B).

Now we can show how the cc–family of the semantics behaves after our translation. Unlike in the previous translations in this section, we take into account the standard conflict–free, not pd–acyclic conflict–free semantics. However, just like in the ac–approach, we use the (standard) grounded semantics. Since neither stable nor model semantics belong to the cc–family (see Section 2.3.6), they will be excluded from this analysis. Please note that due to the fact that all arguments in A^b are self–attackers, the extensions of our produced AF will always be subsets of A^{ev}.

Theorem 12.3. *Let* $D = (A, C)$ *be an ADF and* F^D_{CC} *its corresponding AF obtained from Translation 85. If* $S \subseteq A$ *is a conflict–free (cc–admissible, cc–complete, cc–preferred, grounded) extension of* D*, then there exists a conflict–free (admissible, complete, preferred, grounded) extension* $S' = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ *of* F^D_{CC} *s.t.* $S = \bigcup_{i=1}^{n} F_i \cup G_i$*. If* $S' = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ *is a conflict–free (admissible, complete, preferred, grounded) extension of* F^D_{CC}*, then* $S = \bigcup_{i=1}^{n} F_i \cup G_i$ *is conflict–free (cc–admissible, cc–complete, cc–preferred, grounded) extension of* D*.*

We can now redefine our translation and analyze it. Although its properties will be technically the same as in the ac–translation, some of the reasons will be different.
Redefinition of Translation 85: *Let* Fr^{ADF} *be the collection of all ADFs on domain* $\mathcal{U}$ *and* Fr^{AF} *the collection of all AFs on domain* $PEV^{\mathcal{U}} \cup \mathcal{U}^b$ *where* $PEV^{\mathcal{U}} = \{(F, G, B) \mid F$ *is a sequence on* $E \subseteq \mathcal{U}$, $G \subseteq \mathcal{U}$, $F \cup G \neq \emptyset$, $B \subseteq \mathcal{U}\}$*. The translation* $cc\text{-}Tr^{ADF}_{AF}$: $Fr^{ADF} \rightarrow Fr^{AF}$ *is defined as* $cc\text{-}Tr^{ADF}_{AF}((A, C)) = (A', R)$ *for a framework* $(A, C) \in Fr^{ADF}$*, where* $A' = A^{ev} \cup A^b$ *s.t.* $A^{ev} = \{(F, G, B) \mid (F, G, B)$ *is a minimal partially acyclic pd–evaluation for* $a \in A\}$ *and* $A^b = \{a^b \mid a \in A \; s.t. \; \exists (F, G, B) \in A^{ev}, a \in F\}$*, and* $R = \{((F, G, B), (F', G', B')) \mid (F \cup G) \cap B' \neq \emptyset\} \cup \{(a^b, a^b) \mid a^b \in A^b\} \cup \{((F, G, B), a^b) \mid a \in F \cup G\} \cup \{(a^b, (F, G, B)) \mid a \in F\}$.

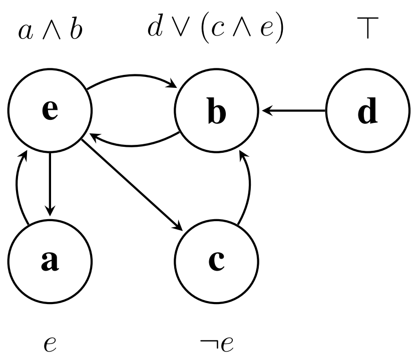
Redefinition of Theorem 12.3: *Let $\sigma^{ADF} \in \{$conflict–free, cc–admissible, cc–complete, cc–preferred, grounded$\}$ be an ADF semantics and $\sigma^{AF} \in \{$conflict–free, admissible, complete, preferred, grounded$\}$ a similar AF semantics. Let SC_{σ}^{Tr} be the extraction–union hybrid semantics casting function for σ defined as $SC_{\sigma}^{X}(S) = \bigcup_{i=1}^{n} F_i \cup G_i$, where $X = (A, C) \in Fr^{ADF}$ and $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \in \sigma(cc\text{-}Tr_{AF}^{ADF}(X))$. The translation $cc\text{-}Tr_{AF}^{ADF}$ is strong under $(\sigma, SC_{\sigma}^{Tr})$. It is also semantics bijective under the (cc–) complete, (cc–) preferred and grounded semantics and the defined casting functions.*
Analysis of Translation 85: Under the conflict–free, (cc–) admissible, (cc–) complete, (cc–) preferred and grounded semantics and the defined casting functions, the translation $cc\text{-}Tr_{AF}^{ADF}$ is:

- full, target–subclass and overlapping
- argument domain altering, argument removing, argument introducing, relation removing, (possibly induced) relation introducing
- generic and semantics domain altering
- semantical

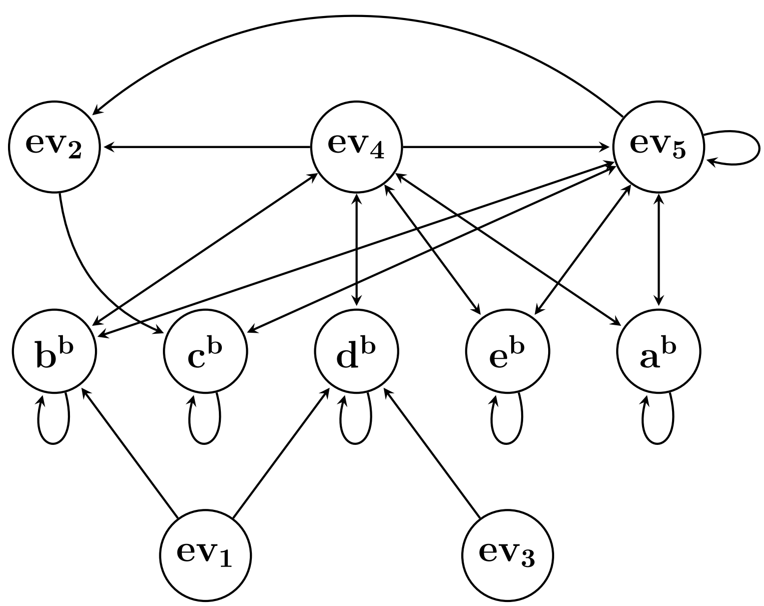
The translation is neither $\oplus$ nor $\otimes$–modular. The translation $cc\text{-}Tr_{AF}^{ADF}$ is classified as a hybrid coalition–defender style under the listed semantics and casting functions.

Explanation. The functional and semantical properties of our approach can be explained in the same manner as in the analysis of Translation 83. The fact that the argument domain undergoes a rather significant change can be easily seen from (the redefinition of) Theorem 12.3. Not every argument in a given ADF might possess a standard evaluation and thus is not necessarily represented in the target framework. Therefore, the translation is argument removing. However, for two reasons it is also argument introducing - first of all, an argument can appear in more than one evaluation, second of all, we use the breaker arguments for handling support cycles. The removal of arguments and imprecision of the evaluations in representing the original framework make our approach relation removing. New conflicts associated with the breaker arguments are added to the target AF, and thus the method is relation introducing as well. Due to the aforementioned imprecision, we can choose to see this addition as induced or not (see explanations in e.g. Translation 82).

Just like in the previous cases, our translation is neither $\oplus$ nor $\otimes$–modular. Let $D_1 = (\{a, b\}, \{C_a = b, C_b = \top\})$ and $D_2 = (\{a, b\}, \{C_a = \top, C_b = a\})$ be two ADFs just like in the analysis of Translation 82. The union of their associated AFs is $(\{(\emptyset, (a), \emptyset), (\emptyset, (a, b)), \emptyset), (\emptyset, (b), \emptyset), (\emptyset, (b, a)), \emptyset)\}, \emptyset)$. However, the AF corresponding to $D_1 \otimes D_2 = (\{a, b\}, \{C_a = b, C_b = a\})$ is $(\{(\{a, b\}, \emptyset, \emptyset), a^b, b^b\}, \{((\{a, b\}, \emptyset, \emptyset), a^b), ((\{a, b\}, \emptyset, \emptyset), b^b), (a^b, (\{a, b\}, \emptyset, \emptyset)),$ $(b^b, (\{a, b\}, \emptyset, \emptyset)), (a^b, a^b), (b^b, b^b)\})$. Therefore, our translation is not $\otimes$–modular. Let us now come back to the ADFs $D_3 = (\{a\}, \{C_a = a\})$ and $D_4 = (\{a\}, \{C_a = \neg a\})$. The AF corresponding to D_3 is $(\{(\{a\}, \emptyset, \emptyset), a^b\}, \{((\{a\}, \emptyset, \emptyset), a^b), (a^b, (\{a\}, \emptyset, \emptyset)), (a^b, a^b)\})$, while the one associated with D_4 is $(\{(\emptyset, (a), \{a\})\}, \{(\emptyset, ((a), \{a\}), (\emptyset, (a), \{a\}))\})$. However, if we were to join these frameworks and perform a translation on $D_3 \oplus D_4 = (\{a\}, \{C_a = \top\})$, we would obtain the framework $(\{(\emptyset, (a), \emptyset)\}, \emptyset)$. Hence, our approach is not $\oplus$–modular. ■



(a) Sample ADF



(b) Associated AF for cc–semantics

Figure 117: Sample ADF and its associated AF for cc–semantics

Example 141. Let us look at the ADF $D = (\{a, b, c, d, e\}, \{C_a = e, C_b = d \vee (c \wedge e), C_c = \neg e, C_d = \top, C_e = a \wedge b\})$ depicted in Figure 117a and previously analyzed in Examples 28 and 138. From Example 28, we know that the conflict–free extensions of D are $\emptyset$, $\{c\}$, $\{d\}$, $\{b, d\}$, $\{c, d\}$, $\{b, c, d\}$ and $\{a, b, d, e\}$. The cc–admissible sets are $\emptyset$, $\{d\}$, $\{b, d\}$ and $\{a, b, d, e\}$. Our cc–complete extensions are $\{b, d\}$ and $\{a, b, d, e\}$, with $\{b, d\}$ being also the grounded extensions and $\{a, b, d, e\}$ the single cc–preferred one.

Based on Example 138, the minimal decisively in interpretations in this framework are $v_a = \{e : \mathbf{t}\}$, $v_b^1 = \{d : \mathbf{t}\}$, $v_b^2 = \{c : \mathbf{t}, e : \mathbf{t}\}$, $v_c = \{e : \mathbf{f}\}$, $v_d = \emptyset$ and $v_e = \{a : \mathbf{t}\, b : \mathbf{t}\}$. Thus, we can create the following minimal partially acyclic evaluations: $ev_1 = (\emptyset, (d, b), \emptyset)$ for b, $ev_2 = (\emptyset, (c), \{e\})$ for c, $ev_3 = (\emptyset, (d), \emptyset)$ for d, $ev_4 = (\{a, b, d, e\}, \emptyset, \emptyset)$ for a and e, and $ev_5 = (\{a, b, c, e\}, \emptyset, \{e\})$ for a, b and e. We can observe that they are little more than partially acyclic representations of the acyclic and standard evaluations from Example 138[29].

We can now create the AF $F^D = (\{ev_1, ev_2, ev_3, ev_4, ev_5, a^b, b^b, c^b, d^b, e^b\}, \{(ev_4, ev_2), (ev_4, ev_5), (ev_5, ev_2), (ev_5, ev_5), (a^b, a^b), (b^b, b^b), (c^b, c^b), (d^b, d^b), (e^b, e^b), (ev_1, b^b), (ev_1, d^b), (ev_2, c^b), (ev_3, d^b), (ev_4, a^b), (ev_4, b^b), (ev_4, d^b), (ev_4, e^b), (ev_5, a^b), (ev_5, b^b), (ev_5, c^b), (ev_5, e^b), (a^b, ev_4), (b^b, ev_4), (d^b, ev_4), (e^b, ev_4), (a^b, ev_5), (b^b, ev_5), (c^b, ev_5), (e^b, ev_5)\})$ associated with our ADF and specialized for the cc–semantics. We can see it depicted in Figure 117b. Its admissible extensions are $\emptyset$, $\{ev_1\}$, $\{ev_3\}$, $\{ev_4\}$, $\{ev_1, ev_3\}$, $\{ev_1, ev_4\}$, $\{ev_3, ev_4\}$ and $\{ev_1, ev_3, ev_4\}$. The associated ADF sets are $\emptyset$, $\{b, d\}$, $\{d\}$ and $\{a, b, d, e\}$, which are exactly the cc–admissible extensions of D. The complete sets of F^D are $\{ev_1, ev_3\}$ and $\{ev_1, ev_3, ev_4\}$. They correspond to $\{b, d\}$ and $\{a, b, d, e\}$, which are the cc–complete extensions of D. We can now easily verify that the grounded and (cc–) preferred extensions of both frameworks are also in agreement.

12.1.4 CA Semantics

Out of all of our approaches, the translations dedicated to the ca–families of semantics are the most problematic ones. Although their unusual structure (see e.g. Theorem 2.158) made the ca$_2$–family a perfect match for the EAF semantics (see Section 8.6.2), it means we face issues similar to the ones from Section 8.2. In the previous case, the solution to this problem was to limit ourselves to the frameworks in which the behavior of the semantics is somewhat more "traditional". In our approach it would mean assuming we are working with AADF$^+$s. However, as seen in Theorem 2.172, our semantics classification "collapses" on such frameworks and thus we could use any of the already presented translations to obtain our extensions. Therefore, we will try to approach this issue differently. Even though we are not able to create a generic ca–translation, we may still point out where the issues are.

The semantics classification system that we have introduced is based on two parameters: the arguments we can accept or arguments that are valid (the "inside" restriction), and which arguments we need to defend from or which arguments are valid attackers (the "outside" restriction). The validity was specified by the types of positive dependency evaluations we wanted a given argument to possess. In a homogeneous approach, such as the aa–family, whatever was "valid" for acceptance was "valid" for attack and vice versa.

[29] Please note that despite the fact that argument c meets the requirements of an a_0 element of a sequence argument in an acyclic evaluation, it is contained in the pd–set. This is due to the fact that b cannot leave the pd–set and depends on c. Consequently, a partially acyclic evaluation should consist both of a front pd–sequence and an end pd–sequence and we will consider this improvement in the future. It would e.g. also address the unnecessary introduction of the breaker argument for c.

Thus, the translation only required the removal of arguments that did not possess any acyclic evaluations. In the ac–approach, an argument we could accept was also considered capable of attack, but not the other way around. Hence, we used the self–attacking approach to simulate this behavior for those arguments (evaluations) that did not meet the acyclicity restriction. We now reach to the ca–family, in which not everything we can accept is valid for attack, even though the other way around holds. We can try to simulate this behavior in AFs in various ways; the first one requires the removal of particular relations, the second focuses on the relation addition, and the last one on the introduction of auxiliary arguments that can serve as additional defenders. We will now describe the problems of all of these approaches.

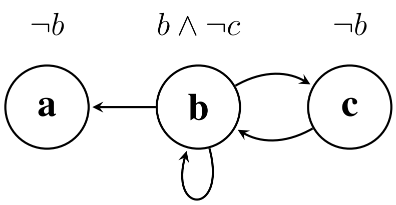
Let us first focus on the relation removal. The idea is rather simple; if a relation is not valid, we should remove it – this has already been done successfully in other frameworks for various validity forms (see Section 4.3). However, the problem with both ca_1 and ca_2 semantics is the fact that validity of attacks is in a certain sense conditional, which was not the case in any of the other families. Let us look at the framework $D_1 = (\{a, b, c\}, \{C_a = \neg b, C_b = b \wedge \neg c, C_c = \neg b\})$ depicted in Figure 118a. The framework F_1 (made of gray and black edges only) visible in Figure 118c is the AF we would obtain if we followed the translation for cc–semantics (Translation 85). We can observe that the argument b has one (minimal) partially acyclic evaluation $(\{b\}, \emptyset, \{c\})$ and no acyclic ones. Unless b is accepted in a ca_1 extension, it is automatically assigned $\mathbf{f}$ by the acyclic range. In other words, we do not need to concern ourselves with the possible conflicts it carries out, and thus the set $\{a\}$ is ca_1–admissible. However, the word "unless" is important here – if b is accepted in an extension, then the conflicts it carries out are valid (i.e. b is used in determining whether an evaluation is blocked or not). For this reason, b can block the evaluation for c and the set $\{b\}$ is ca_1–admissible. This inconsistency was visible e.g. in Proposition 2.150, Theorem 2.155 and Theorem 2.158. What it means for us is that we cannot use the relation removal method in our AF translation, as validity of a given relation changes depending on how a given extension looks like. The same situation appears in the ca_2 semantics, though this time it is from the definition of the partially acyclic range, which explicitly states that whether an argument is put into the discarded set "by default" depends on what has been accepted in a given extension. In the presented example, both $\{a\}$ and $\{b\}$ would be ca_2–admissible as well. If we were to "remove" the conflicts carried out by b (i.e. the grayed out conflicts in F_1 in Figure 118c), the resulting structure would be similar to the AF for the framework $D_2 = (\{a, b, c\}, \{C_a = \top, C_b = b \wedge \neg c, C_c = \top\})$ from Figure 118b. However, in this case the set $\{a, b\}$ is conflict–free and $\{b\}$ is not admissible, unlike in D_1. Therefore, the removal of relations affects both conflict–freeness and decisiveness (defense) in a way that is not desirable from our perspective and should not be pursued further.

In the next method, instead of removing invalid attacks, we add symmetric conflicts so that the attacked argument is able to defend itself. Thus, we work again with the AF F_1 depicted in Figure 118c, but this time take into account gray, black and red edges. This structure is similar to the one we would create for $D_3 = (\{a, b, c\}, \{C_a = \neg b, C_b =$

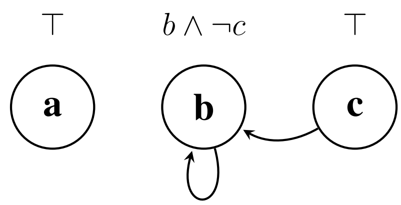
$b \wedge \neg c \wedge \neg a, C_c = \neg b\})$ visible in Figure 118d. Both D_1 and D_3 produce the same extensions; we have five ca$_1$ and ca$_2$ admissible sets – $\emptyset$, $\{a\}$, $\{b\}$, $\{c\}$ and $\{a, c\}$, two that are at the same time complete, preferred and model extensions – $\{b\}$ and $\{a, c\}$ – with the latter being the acyclic grounded extension. This basically means that the relation addition method is so far quite acceptable on the ADF level. However, this does not necessarily mean that it is adequate on the AF level. In fact, not all of the extensions we can get from our modified F_1 are "correct". The admissible and preferred extensions correspond to the sets that we have expected. However, the complete and grounded do not; F_1 produces one additional complete extension $\emptyset$ which is also grounded. Moreover, only one of our model extensions – $\{b\}$ – is retrieved as the single stable extension of F_1 $\{(\{b\}, \emptyset, \{c\})\}$.

The reason for the loss of one of the model extensions is the presence of the breaker argument that is attacked only by its associated evaluation. For this particular semantics, this can be addressed by not enforcing self–attacks on the breaker arguments, which will be described further in the next method. Unfortunately, the cause of the misbehaving of the complete and grounded semantics is much more difficult to resolve. The problem can be explained in two ways. First of all, it can be seen as the difference between an initial and a self–defending argument. Although both of them will appear in admissible extensions, the initial argument will be defended by any set (in particular, $\emptyset$), while the self–defending one might not – we can only be sure it is defended by a set that contains it in the first place. Therefore, while in our case a is decisively in w.r.t. the acyclic range of $\emptyset$ in D_1 and D_3 (thus acting as "initial" w.r.t. the assumed notion of validity), $(\emptyset, (a), \{b\})$ is not defended by $\emptyset$ in F_1. A different way to explain this issue is by looking at the discarded sets. The discarded set (of any kind) of the empty set is not necessarily empty itself in ADFs. However, it will always be such in AFs. Making sure that the two would eventually "come together" would require making appropriate evaluation arguments attacked by an auxiliary initial argument. Unfortunately, this of course prevents these evaluations from appearing in an admissible extension of any sort, thus damaging the behavior of semantics to even a larger extent. Therefore, we are not aware of any way we can repair the relation addition method in a way that would allow us handle the complete and grounded approaches.

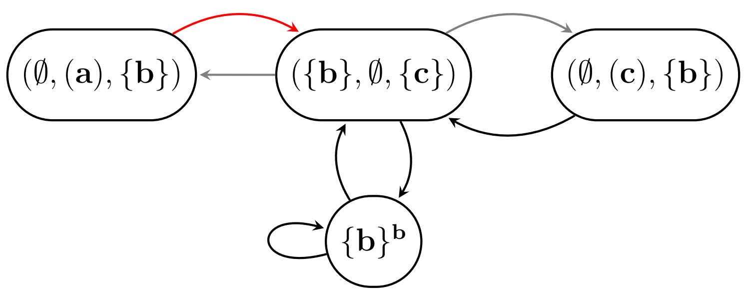
The auxiliary argument method is in fact quite similar to what has been done in Translation 85 dedicated to the cc–semantics. The difference is that in this case, we remove the self–attack restriction from the breaker arguments. Accepting such arguments in an extension corresponds to rejecting the original argument due to its lack of acyclicity. Let us again focus on F_1 depicted in Figure 118c and take into account all gray and black edges with the exception of the $(\{b\}^b, \{b\}^b)$ one. The resulting structure will produce admissible extensions corresponding to sets $\emptyset$, $\{a\}$ (the original extension contains the auxiliary argument), $\{b\}$, $\{c\}$ and $\{a, c\}$, which is the desired result. In this case, also the preferred and stable (model) sets are the ones we hoped for. Nevertheless, we still have one complete extension too many (namely, $\emptyset$), and the produced grounded one corresponds to the standard approach and not the acyclic one. Furthermore, when we consider more complex examples, the behavior of the preferred semantics "breaks". Let us consider the framework $(\{a, b\}, \{C_a = a, C_b = b\})$, previously studied in Example 29. It has a single (ca_1 and ca_2) preferred extension, namely $\{a, b\}$. The AF constructed for this framework with the current method would be $(\{(\{a\}, \emptyset, \emptyset), (\{b\}, \emptyset, \emptyset), \{a\}^b, \{b\}^b\}, \{((\{a\}, \emptyset, \emptyset), \{a\}^b), (\{a\}^b, (\{a\}, \emptyset, \emptyset)), ((\{b\}, \emptyset, \emptyset), \{b\}^b), (\{b\}^b, (\{b\}, \emptyset, \emptyset))\})$. It produces in total four preferred extensions – the first one containing the evaluations for a and b, next with evaluation for a and the breaker for b, then the breaker for a and evaluation for b, and finally only the breakers. This corresponds to extensions $\{a, b\}$, $\{a\}$, $\{b\}$ and $\emptyset$, and therefore produces multiple extensions we had not expected.. This mismatch is due to the fact that the breaker arguments are accepted independently of each other and thus the way they appear in an AF extension is not in the desired correspondence with how the (partially) acyclic range is built. Thus, this method, similarly to the approaches we had discussed before, does not handle the ca_1 and ca_2 semantics as successfully as we have managed in the other families.



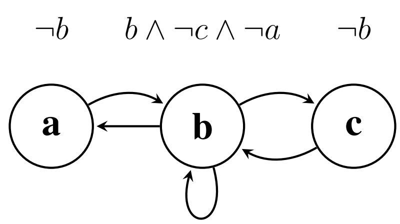
(a) D_1



(b) D_2



(c) F_1



(d) D_3

Figure 118: Sample methods for handling the ca_1 and ca_2 semantics

The analysis above has presented the difficulties in designing the translations aimed at the ca–families of ADF semantics. Consequently, what we will introduce now should only be treated as a partial solution for the ca$_2$–case and we hope that in the future we will be able to create a more satisfactory solution and manage to extend it to the ca$_1$–family. We will focus on the last of the described methods that works with the breaker arguments and removes the self–attack restrictions from them that was present in Translation 85. Although its usefulness in this case is limited, the independence of the breaker arguments that causes problems for the ca–semantics is precisely what we want for the labeling–based approaches. Consequently, this method will be the main topic of the next section.

Translation 86. *Let $D = (A, C)$ be an ADF. Its corresponding AF $F^D_{CA_2} = (A', R)$ is built the following way:*

- *let $A^{ev} = \{(F, G, B) \mid (F, G, B)$ is a minimal partially acyclic pd–evaluation for $a \in A\}$,*
- *let $A^b = \{a^b \mid \exists (F, G, B) \in A^{ev}$ s.t. $a \in F\}$,*
- *$A' = A^{ev} \cup A^b$,*
- *for every $(F, G, B), (F', G', B') \in A^{ev}$, (F, G, B) attacks $(F,' G', B')$ if $(F \cup G) \cap B' \neq \emptyset$,*
- *for every $(F, G, B) \in A^{ev}$, $a^b \in A^b$ s.t. $a \in F \cup G$, (F, G, B) attacks a^b, and*
- *for every $(F, G, B) \in A^{ev}$, $a^b \in A^b$ s.t. $a \in F$, a^b attacks (F, G, B).*

Theorem 12.4. *Let $D = (A, C)$ be an ADF and $F^D_{CA_2} = (A', R)$ its corresponding AF obtained from Translation 86. If $E \subseteq A$ is a conflict–free (ca$_2$–admissible, ca$_2$–complete, ca$_2$–preferred, model, grounded) extension of D, then there exists a conflict–free (admissible, complete, preferred, stable, grounded) extension $E' = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n), a^b_1, ..., a^b_m\} \subseteq A'$ of $F^D_{CA_2}$ s.t. $E = \bigcup^n_{i=1} F_i \cup G_i$. If $E' = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n), a^b_1, ..., a^b_m\} \subseteq A'$ is a conflict–free (admissible, stable, grounded) extension of $F^D_{CA_2}$, then $E = \bigcup^n_{i=1} F_i \cup G_i$ is conflict–free (ca$_2$–admissible, model, grounded) extension of D.*

We can now put the translation into our system. It can be observed that the analysis will be the same as in the case of the cc–semantics, which should not be surprising if we take into account that, structurally speaking, this approach is only a minor modification of Translation 85. Thus, further explanations will be omitted.

Redefinition of Translation 86: *Let Fr^{ADF} be the collection of all ADFs on domain $\mathcal{U}$ and Fr^{AF} the collection of all AFs on domain $PEV^{\mathcal{U}} \cup \mathcal{U}^b$ where $PEV^{\mathcal{U}} = \{(F, G, B) \mid F$ is a sequence on $E \subseteq \mathcal{U}$, $G \subseteq \mathcal{U}$, $F \cup G \neq \emptyset$, $B \subseteq \mathcal{U}\}$. The translation ca$_2$-$Tr^{ADF}_{AF}$: $Fr^{ADF} \rightarrow Fr^{AF}$ is defined as ca$_2$-$Tr^{ADF}_{AF}((A, C)) = (A', R)$ for a framework $(A, C) \in Fr^{ADF}$, where $A' = A^{ev} \cup A^b$ s.t. $A^{ev} = \{(F, G, B) \mid (F, G, B)$ is a minimal partially acyclic pd–evaluation for $a \in A\}$ and $A^b = \{a^b \mid a \in A \, s.t. \, \exists (F, G, B) \in A^{ev}, a \in F\}$,* *and* $R = \{((F,G,B),(F',G',B')) \mid (F \cup G) \cap B' \neq \emptyset\} \cup \cup\{((F,G,B),a^b) \mid a \in F \cup G\} \cup \{(a^b,(F,G,B)) \mid a \in F\}$.

Redefinition of Theorem 12.4: *Let $\sigma^{ADF} \in \{$conflict–free, ca$_2$–admissible, grounded, model$\}$, $\delta^{ADF} \in \{$ca$_2$–complete, ca$_2$–preferred$\}$ be ADF semantics and $\sigma^{AF} \in \{$conflict–free, admissible, grounded, stable$\}$, $\delta^{AF} \in \{$complete, preferred$\}$ similar AF semantics. Let SC^{Tr}_{σ} be the extraction–union hybrid semantics casting function for σ (and δ) defined as $SC^X_\sigma(S) = \bigcup_{i=1}^{n} F_i \cup G_i$, where $X = (A, C) \in Fr^{ADF}$ and $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n), a^b_1, ..., a^b_m\} \in \sigma(ca_2\text{-}Tr^{ADF}_{AF}(X))$. The translation $ca_2\text{-}Tr^{ADF}_{AF}$ is strong under (σ, SC^{Tr}_σ) and $\subseteq$–weak under (δ, SC^{Tr}_δ) It is also semantics bijective under the (stable) model and grounded semantics and the defined casting functions.*

Analysis of Translation 86: Under the conflict–free, (ca$_2$–) admissible, (ca$_2$–) complete, (ca$_2$–) preferred, (model) stable and grounded semantics and the defined casting functions, the translation $ca_2\text{-}Tr^{ADF}_{AF}$ is:

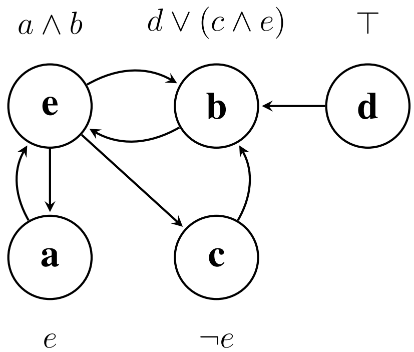
- full, target–subclass and overlapping
- argument domain altering, argument removing, argument introducing, relation removing, (possibly induced) relation introducing
- generic and semantics domain altering
- semantical

The translation is neither $\oplus$ nor $\otimes$–modular. The translation $ca_2\text{-}Tr^{ADF}_{AF}$ is classified as a hybrid coalition–defender style under the listed semantics and casting functions.

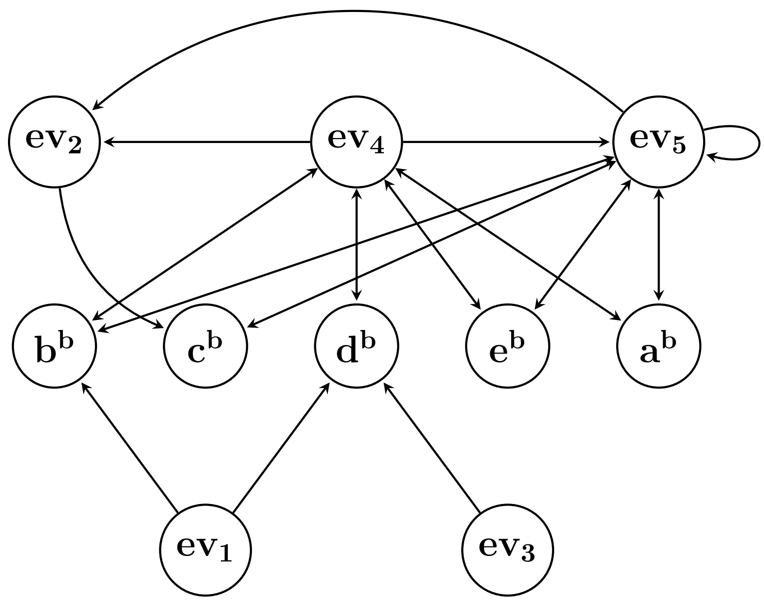
Example 142. Let us again look at the ADF $D = (\{a, b, c, d, e\}, \{C_a = e, C_b = d \vee (c \wedge e), C_c = \neg e, C_d = \top, C_e = a \wedge b\})$ depicted in Figure 119a and previously analyzed in Examples 28, 138 and 141.

The ca$_2$–admissible extensions of D are $\emptyset$, $\{c\}$, $\{d\}$, $\{b, d\}$, $\{c, d\}$, $\{b, c, d\}$ and $\{a, b, d, e\}$. The last two are also ca$_2$–complete, ca$_2$–preferred and models. $\{b, d\}$ is the grounded extension. Based on Example 141, we can create the following minimal partially acyclic evaluations: $ev_1 = (\emptyset, (d, b), \emptyset)$ for b, $ev_2 = (\emptyset, (c), \{e\})$ for c, $ev_3 = (\emptyset, (d), \emptyset)$ for d, $ev_4 = (\{a, b, d, e\}, \emptyset, \emptyset)$ for a and e, and $ev_5 = (\{a, b, c, e\}, \emptyset, \{e\})$ for a, b and e.

We can now create the AF $F^D = (\{ev_1, ev_2, ev_3, ev_4, ev_5, a^b, b^b, c^b, d^b, e^b\}, \{(ev_4, ev_2), (ev_4, ev_5), (ev_5, ev_2), (ev_5, ev_5), (ev_1, b^b), (ev_1, d^b), (ev_2, c^b), (ev_3, d^b), (ev_4, a^b), (ev_4, b^b), (ev_4, d^b), (ev_4, e^b), (ev_5, a^b), (ev_5, b^b), (ev_5, c^b), (ev_5, e^b), (a^b, ev_4), (b^b, ev_4), (d^b, ev_4), (e^b, ev_4), (a^b, ev_5), (b^b, ev_5), (c^b, ev_5), (e^b, ev_5)\})$ associated with our ADF and specialized for the ca$_2$–semantics. We can see it depicted in Figure 119b. The difference between this framework and the one from Example 141 is in the removal of self–attacks from the breaker arguments. The admissible extensions of F^D are $\emptyset$, $\{ev_1\}$, $\{ev_3\}$, $\{a^b\}$, $\{e^b\}$ and all of their combinations, then $\{ev_2, a^b\}$ and $\{ev_2, e^b\}$ and all of their combinations with the previous sets, and finally $\{ev_4\}$, $\{ev_1, ev_4\}$, $\{ev_3, ev_4\}$, $\{ev_4, c^b\}$, $\{ev_1, ev_3, ev_4\}$,



(a) Sample ADF



(b) Associated AF for the ca$_2$–semantics

Figure 119: Sample ADF and its associated AF for the ca$_2$–semantics

$\{ev_1, ev_4, c^b\}$, $\{ev_3, ev_4, c^b\}$, and $\{ev_1, ev_3, ev_4, c^b\}$. Therefore, the sets we can retrieve from them are $\emptyset$, $\{b, d\}$, $\{d\}$, $\{c\}$, $\{b, c, d\}$, $\{c, d\}$ and $\{a, b, d, e\}$. We can observe these are all and only ca$_2$–admissible extensions of D. The complete extensions of our AF are $\{ev_1, ev_3\}$, $\{ev_1, ev_3, ev_4, c^b\}$ and $\{ev_1, ev_2, ev_3, a^b, e^b\}$. They correspond to $\{b, d\}$, $\{a, b, d, e\}$ and $\{b, c, d\}$; unfortunately, $\{b, d\}$ is not a ca$_2$–complete extension of D. On this occasion, the preferred extensions of both frameworks agree, though please note it is not necessarily always the case. However, the grounded set $\{ev_1, ev_3\}$ is correctly retrieved as the grounded extension $\{b, d\}$ of D^F. Both $\{ev_1, ev_3, ev_4, c^b\}$ and $\{ev_1, ev_2, ev_3, a^b, e^b\}$ (i.e. $\{a, b, d, e\}$ and $\{b, c, d\}$) are the stable extensions of F^D. They coincide with the models of D.

12.1.5 Labeling–Based semantics

In the previous section we have described various attempts at translating ADFs to AFs w.r.t. the ca–families of extension–based semantics. The last method we have described (i.e. Translation 86) was a modification of Translation 85 that was aimed at the cc–semantics. The removal of self–attack from the breaker arguments allowed us to accept them and simulate the effect of discarding an argument due to its participation in a positive dependency cycle. Although this approach was not entirely sufficient for the ca–semantics, it is precisely what we need to model the labeling–based semantics of ADFs.

Before we proceed, we will introduce some notation first that will help us to write down the relations between source and target labelings more easily. Retrieving the original extensions from the translated ones was usually a matter of using or combining the casting functions listed in Definition 3.6. However, when we come to labelings, it is not necessarily that easy, particularly when a source argument can be represented by a number of target ones. Although assigning $\mathbf{t}$ to an argument is a matter of verifying if any evaluation containing it is accepted, retrieving the $\mathbf{f}$ mapping requires checking if all of the evaluations including the desired argument are rejected as well. Thus, we will introduce some functions meant for retrieving evaluations.

Let $E = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n), a_1^b, ..., a_m^b\}$ be a set of arguments F_{lab}^D; the set of D arguments that can be retrieved from E will be defined as $\uplus^{E}_{V}\ E = \bigcup_{i=1}^{n} F_i \cup G_i$. Given an argument $a \in A$ in D, the collection of arguments in A' in F_{lab}^D containing a will be denoted as $EV^p(a) = \{(F, G, B) \in A^{ev} \mid a \in F \cup G\}$. We can extend the EV^p function to sets of arguments as well. Finally, by $ALL(E) = \{a \in A \mid EV^p(a) \subseteq E\}$ we will understand the set of those arguments in D s.t. all of their partially acyclic evaluations are contained in the set of F_{lab}^D arguments E.

Theorem 12.5. *Let $D = (A, C)$ be an ADF and $F_{lab}^D = (A', R)$ its corresponding AF obtained through Translation 86. If v is an admissible labeling of D, then there exists an admissible labeling v' of F_{lab}^D s.t. $v^{\mathbf{t}} = \uplus^{E}_{V}\ in(v')$ and $EV^p(v^{\mathbf{f}}) \subseteq out(v')$. If v is a complete (preferred, grounded) labeling of D, then there exists a complete (preferred, grounded) labeling v' of F_{lab}^D s.t. $v^{\mathbf{t}} = \uplus^{E}_{V}\ in(v')$ and $v^{\mathbf{f}} = ALL(out(v'))$.*

If v' is an complete (preferred, grounded) labeling of F_{lab}^D, then a labeling v of D s.t. $v^{\mathbf{t}} = \uplus^{E}_{V}\ in(v')$ and $v^{\mathbf{f}} = ALL(out(v'))$ is complete (preferred, grounded) in D. This does not necessarily hold for admissible semantics.

What is perhaps somewhat surprising, taking into account the behaviour of the translations concerning the extension–based semantics for ADFs, is the weakness of the current approach w.r.t. the admissible labelings. Although the translation was strong w.r.t. the ca_2–admissible extensions and there is a particular correspondence between these two semantics (see Theorem 2.166), the issue lies with the arguments rejected in a labeling. Not every argument that can be assigned *out* has to be assigned *out* in the labeling–based semantics for AFs. This means that when translating back into an ADF, a given argument can be assigned $\mathbf{f}$, but another argument it depended on might not, which in turn can lead to the loss of decisiveness. Let us look at the following example:

Example 143. Let us consider a simple ADF $D = (\{a, b, c\}, \{C_a = \top, C_b = \neg a, C_c = b\})$. Every argument possesses precisely one minimal partially acyclic evaluation. The evaluation for a is $ev_a = (\emptyset, (a), \emptyset)$, for b we have $ev_b = (\emptyset, (b), \{a\})$ and $ev_c = (\emptyset, (b, c), \{a\})$ for c. We can observe that no breaker arguments will be created. The associated AF is now $(\{ev_a, ev_b, ev_c\}, \{(ev_a, ev_b), (ec_a, ev_c)\})$. We can observe that assigning *in* to ev_a, *undec* to ev_b and *out* to ev_c gives us an admissible labeling. The corresponding labeling in our ADF is $v = \{a : \mathbf{t}, b : \mathbf{u}, c : \mathbf{f}\}$. Unfortunately, v is not admissible in D. We can observe that the characteristic operator produces associates with v the labeling $\Gamma_D(v) = \{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{u}\}$, which is incomparable with v w.r.t. $\leq_i$. Only assigning *out* to both ev_b and ev_c would give us an admissible ADF labeling.

Taking into account the analysis done in the previous section, it is only the semantics theorem that needs to be redefined and put into our system. Although in this case we manage to preserve many more semantics than previously, both methods still qualify as generic, and thus the properties remain unchanged.

Redefinition of Theorem 12.5: *Let $\sigma^{ADF} \in \{$complete, preferred, grounded$\}$ be a labeling–based ADF semantics and $\sigma^{AF} \in \{$complete, preferred, grounded$\}$ a similar AF semantics. Let SC_σ^{Tr} be the semantics casting function for σ defined as $SC_\sigma^X(v') = v$ s.t. $v^{\mathbf{t}} = \biguplus^{E}_{V} v'^{\mathbf{t}}$, $v^{\mathbf{f}} = ALL(v'^{\mathbf{f}})$ and $v^{\mathbf{u}} = A \setminus (v^{\mathbf{t}} \cup v^{\mathbf{f}})$, where $X = (A, C) \in Fr^{ADF}$ and $v' \in \sigma(ca_2\text{-}Tr_{AF}^{ADF}(X))$. The translation $ca_2\text{-}Tr_{AF}^{ADF}$ is strong and semantics bijective under (σ, SC_σ^{Tr}). It is $\subseteq$–weak under the admissible semantics and the defined casting function.*

Example 144. Let us again look at the ADF $D = (\{a, b, c, d, e\}, \{C_a = e, C_b = d \vee (c \wedge e), C_c = \neg e, C_d = \top, C_e = a \wedge b\})$ depicted in Figure 119a and analyzed in Example 142. Its associated AF is $F^D = (\{ev_1, ev_2, ev_3, ev_4, ev_5, a^b, b^b, c^b, d^b, e^b\}, \{(ev_4, ev_2), (ev_4, ev_5), (ev_5, ev_2), (ev_5, ev_5), (ev_1, b^b), (ev_1, d^b), (ev_2, c^b), (ev_3, d^b), (ev_4, a^b), (ev_4, b^b), (ev_4, d^b), (ev_4, e^b), (ev_5, a^b), (ev_5, b^b), (ev_5, c^b), (ev_5, e^b), (a^b, ev_4), (b^b, ev_4), (d^b, ev_4), (e^b, ev_4), (a^b, ev_5), (b^b, ev_5), (c^b, ev_5), (e^b, ev_5)\})$, where $ev_1 = (\emptyset, (d, b), \emptyset)$, $ev_2 = (\emptyset, (c), \{e\})$, $ev_3 = (\emptyset, (d), \emptyset)$, $ev_4 = (\{a, b, d, e\}, \emptyset, \emptyset)$ and $ev_5 = (\{a, b, c, e\}, \emptyset, \{e\})$.

The complete labelings of D are $\{a : \mathbf{u}, b : \mathbf{t}, c : \mathbf{u}, d : \mathbf{t}, e : \mathbf{u}\}$, $\{a : \mathbf{t}, b : \mathbf{t}, c : \mathbf{f}, d : \mathbf{t}, e : \mathbf{t}\}$ and $\{a : \mathbf{f}, b : \mathbf{t}, c : \mathbf{t}, d : \mathbf{t}, e : \mathbf{f}\}$. The first one is grounded and the other two preferred.

The complete labelings of our AF are v_1, where $in(v_1) = \{ev_1, ev_3\}$, $out(v_1) = \{b^b, d^b\}$ and $undec(v_1) = \{ev_2, ev_4, ev_5, ev_6, a^b, c^b, e^b\}$, v_2, where $in(v_2) = \{ev_1, ev_3, ev_4, c^b\}$, $out(v_2) = \{ev_2, ev_5, a^b, b^b, d^b, e^b\}$ and $undec(v_2) = \emptyset$, and v_3, where $in(v_3) = \{ev_1, ev_2, ev_3, a^b, e^b\}$, $out(v_3) = \{ev_4, ev_5, b^b, c^b, d^b\}$ and $undec(v_3) = \emptyset$. v_1 is the grounded labeling of F^D, while v_2 and v_3 are preferred. The ADF labelings associated with the AF ones are $v_1' = \{a : \mathbf{u}, b : \mathbf{t}, c : \mathbf{u}, d : \mathbf{t}, e : \mathbf{u}\}$, $v_2' = \{a : \mathbf{t}, b : \mathbf{t}, c : \mathbf{f}, d : \mathbf{t}, e : \mathbf{t}\}$, and $v_3' = \{a : \mathbf{f}, b : \mathbf{t}, c : \mathbf{t}, d : \mathbf{t}, e : \mathbf{f}\}$. We can observe they coincide with the previously listed complete labelings. It is now easy to check that the grounded and preferred labelings are also retrieved.

12.1.6 Improvements

In this section we have introduced a variety of translations from ADFs to AFs, each of them meant to handle a different family of ADF semantics. We believe they will be improved in the future, particularly due to the fact that all of them follow the coalition pattern, which is in a certain sense weaker than the defender and attack propagation methods. However, one thing needs to be stated openly: with the exception of the grounded and acyclic grounded cases, there exist no exact translations from ADFs to AFs for the analyzed semantics, even when we limit ourselves to BADFs or AADF$^+$s:

Theorem 12.6. *Let Fr^{ADF} be the collection of all ADFs, $BADF$ the collection of all BADFs and $AADF^+$ of all AADF$^+$s, all on a domain $\mathcal{U}^{ADF}$. Let Fr^{AF} the collection of all AFs on a domain $\mathcal{U}^{AF}$. There exists no full (resp. source–subclass) translation from Fr^{ADF} (resp. $BADF$, $AADF^+$) to Fr^{AF} that is exact under conflict–free, pd–acyclic conflict–free, xy–admissible, xy–complete, xy–preferred, stable, model, three–valued model, labeling admissible, labeling complete and labeling preferred semantics and identity casting functions for them.*

Although it may sound impressive at first, this theorem is basically a result of the fact that ADFs handle SETAFs easily and AFs do not. We can repeat the analysis from Section 6.1.3 and represent the discussed SETAFs as ADFs using Translation 31. Since SETAFs also fall into the two mentioned subclasses of ADFs, it is not just the full exact translations that cannot be created, but also certain source–subclass ones.

Due to this gap between AFs and ADFs, the improvements we would like to consider in the future concern the labeling–based translation. In particular, we believe that the defender approach can be exploited in order to connect evaluations and their sub–evaluations in order to address the issues raised in Example 143. This would allow us to improve the strength of our approach w.r.t. the admissible semantics.

12.2 ADF as SETAF

12.2.1 CC Semantics

The aa and ac–families of ADF semantics can be translated into AFs relatively easily. The problems started appearing when we moved to the cc–family, where we needed additional breaker arguments to deal with cycles. Although the translation from ADFs to SETAFs w.r.t. this family of semantics we are about to present is not that different from the one we have presented for AFs, it is the approach we have created first and only later simplified after gaining some insight. The line of reasoning which has lead us to this solution was already presented in Example 140. To each evaluation that was not acyclic we have assigned a breaker argument representing the pd–set. The fact that some breakers might have arguments in common and thus be related was, in the first instance, grasped by the introduction of group attacks, and only later by changing the nature of the breakers. Consequently, even though we believe the AF solution to be more adequate, we would still like to present the original translation.

Translation 87. *Let $D = (A, C)$ be an ADF. Its corresponding SETAF $SF^D_{CC} = (A', R)$ is built the following way:*

- *let $A^{ev} = \{(F, G, B) \mid (F, G, B)$ is a minimal partially acyclic pd–evaluation for $a \in A\}$,*
- *let $A^b = \{F^b \mid (F, G, B) \in A^{ev}, F \neq \emptyset\}$,*
- *$A' = A^{ev} \cup A^b$,*
- *for every $(F, G, B), (F', G', B') \in A^{ev}$, $\{(F, G, B)\}$ attacks $(F,' G', B')$ if $(F \cup G) \cap B' \neq \emptyset$,*
- *for every $F^b \in A^b$, $\{F^b\}$ attacks F^b,*
- *for every $(F, G, B) \in A^{ev}$ s.t. $F \neq \emptyset$, $\{(F, G, B)\}$ attacks F^b and $\{F^b\}$ attacks (F, G, B), and*
- *given an argument $F^b \in A^b$, which represents the set $F \subseteq A$, and a minimal set of arguments $\{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ s.t. $F \subseteq \bigcup_{i=1}^n F_i \cup G_i$, $\{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\}$ attacks F^b.*

Due to the way the attack relation is constructed, we can observe that the resulting SETAF will be in minimal form:

Theorem 12.7. *Let $D = (A, C)$ be an ADF and $SF^D_{CC} = (A', R)$ its corresponding SETAF obtained through Translation 87. SF^D_{CC} is in minimal normal form.*

We can now proceed with analyzing the extensions produced by our source and target frameworks. Please note that every breaker argument in the produced SETAF will be a self attacker and as such, they will not appear in the created sets.

Theorem 12.8. *Let $D = (A, C)$ be an ADF and $SF^D_{CC} = (A', R)$ its corresponding SETAF obtained from Translation 87. If $S \subseteq A$ is a conflict–free (cc–admissible, cc–complete, cc–preferred, grounded) extension of D, then there exists a conflict–free (admissible, complete, preferred, grounded) extension $S' = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ of SF^D_{CC} s.t. $S = \bigcup_{i=1}^n F_i \cup G_i$. If $S' = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ is a conflict–free (admissible, complete, preferred, grounded) extension of SF^D_{CC}, then $S = \bigcup_{i=1}^n F_i \cup G_i$ is conflict–free (cc–admissible, cc–complete, cc–preferred, grounded) extension of D.*

The translation can be put into the system in the same way Translation 85 was. Thus, we will omit further explanations.

Redefinition of Translation 87: *Let Fr^{ADF} be the collection of all ADFs on domain $\mathcal{U}$ and Fr^{AF} the collection of all AFs on domain $PEV^{\mathcal{U}} \cup (2^{\mathcal{U}})^b$ where $PEV^{\mathcal{U}} = \{(F,G,B) \mid F$ is a sequence on $E \subseteq \mathcal{U}$, $G \subseteq \mathcal{U}$, $F \cup G \neq \emptyset$, $B \subseteq \mathcal{U}\}$. The translation cc-$Tr^{ADF}_{SETAF} : Fr^{ADF} \to Fr^{SETAF}$ is defined as cc-$Tr^{ADF}_{SETAF}((A,C)) = (A', R)$ for a framework $(A,C) \in Fr^{ADF}$, where $A' = A^{ev} \cup A^b$ s.t. $A^{ev} = \{(F,G,B) \mid (F,G,B)$ is a minimal partially acyclic pd–evaluation for $a \in A\}$ and $A^b = \{F \mid (F,G,B) \in A^{ev}, F \neq \emptyset\}$, and $R = \{(\{(F,G,B)\},(F',G',B')) \mid (F \cup G) \cap B' \neq \emptyset\} \cup \{(\{F^b\},F^b) \mid F^b \in A^b\} \cup \{(\{(F,G,B)\},F^b),(\{F^b\},(F,G,B)) \mid (F,G,B) \in A^{ev}, F \neq \emptyset\} \cup \{(\{(F_1,G_1,B_1),...,(F_n,G_n,B_n)\},F^b) \mid \{(F_1,G_1,B_1),...,(F_n,G_n,B_n)\} \subseteq A^{ev}$ is a minimal set of arguments s.t. $F^b \subseteq \bigcup_{i=1}^n F_i \cup G_i\}$.*

Redefinition of Theorem 12.8: *Let $\sigma^{ADF} \in \{$conflict–free, cc–admissible, cc–complete, cc–preferred, grounded$\}$ be an ADF semantics and $\sigma^{SETAF} \in \{$conflict–free, admissible, complete, preferred, grounded$\}$ a similar SETAF semantics. Let SC^{Tr}_{σ} be the extraction–union hybrid semantics casting function for σ defined as $SC^X_\sigma(S) = \bigcup_{i=1}^n F_i \cup G_i$, where $X = (A,C) \in Fr^{ADF}$ and $S = \{(F_1,G_1,B_1),...,(F_n,G_n,B_n)\} \in \sigma(cc\text{-}Tr^{ADF}_{SETAF}(X))$. The translation cc-$Tr^{ADF}_{SETAF}$ is strong under (σ, SC^{Tr}_σ). It is also semantics bijective under the (cc–) complete, (cc–) preferred and grounded semantics and the defined casting functions.*

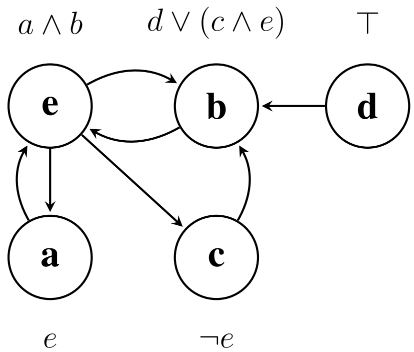
Analysis of Translation 87: Under the conflict–free, (cc–) admissible, (cc–) complete, (cc–) preferred and grounded semantics and the defined casting functions, the translation cc-Tr^{ADF}_{SETAF} is:

- full, target–subclass and overlapping
- argument domain altering, argument removing, argument introducing, relation removing, (possibly induced) relation introducing
- generic and semantics domain altering
- semantical

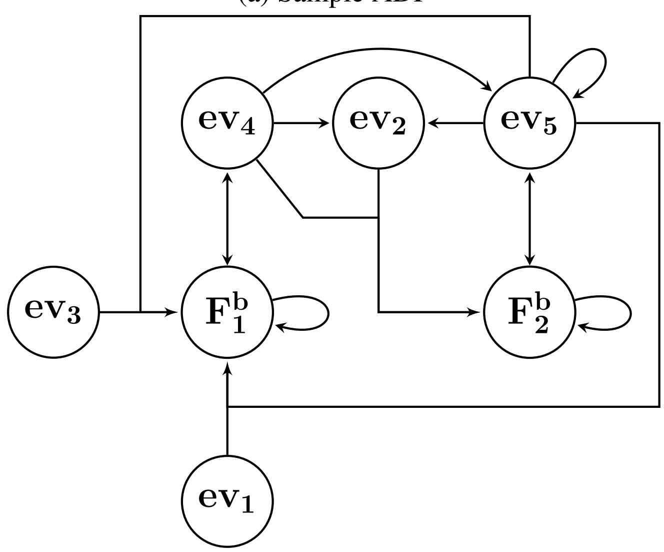
The translation is neither $\oplus$ nor $\otimes$–modular. The translation cc-Tr^{ADF}_{SETAF} is classified as hybrid coalition–defender style under the listed semantics and casting functions.

Example 145. Let us look at the ADF $D = (\{a,b,c,d,e\},\{C_a = e, C_b = d \vee (c \wedge e), C_c = \neg e, C_d = \top, C_e = a \wedge b\})$ depicted in Figure 120a and previously analyzed in Examples 28, 138, 141, 142 and 144. The conflict–free extensions of D are $\emptyset$, $\{c\}$, $\{d\}$, $\{b,d\}$, $\{c,d\}$, $\{b,c,d\}$ and $\{a,b,d,e\}$. The cc–admissible sets are $\emptyset$, $\{d\}$, $\{b,d\}$ and $\{a,b,d,e\}$. Our cc–complete extensions are $\{b,d\}$ and $\{a,b,d,e\}$, with $\{b,d\}$ being also the grounded extensions and $\{a,b,d,e\}$ the single cc–preferred one.

The minimal partially acyclic evaluations for our arguments are $ev_1 = (\emptyset,(d,b),\emptyset)$ for b, $ev_2 = (\emptyset,(c),\{e\})$ for c, $ev_3 = (\emptyset,(d),\emptyset)$ for d, $ev_4 = (\{a,b,d,e\},\emptyset,\emptyset)$ for a



(a) Sample ADF



(b) Associated SETAF for the cc–semantics

Figure 120: Sample ADF and its associated SETAF for the cc–semantics

and e, and $ev_5 = (\{a, b, c, e\}, \emptyset, \{e\})$ for a, b and e. We can observe that in two cases, the pd–sets are not empty and thus will lead to the creation of appropriate breaker arguments $F_1^b = \{a, b, d, e\}^b$ and $F_2^b = \{a, b, c, e\}^b$. We can now create the SETAF $SF^D = (\{ev_1, ev_2, ev_3, ev_4, ev_5, F_1^b, F_2^b\}, \{(\{ev_4\}, ev_2), (\{ev_4\}, ev_5), (\{ev_5\}, ev_2),$ $(\{ev_5\}, ev_5)$, $(\{F_1^b\}, F_1^b)$, $(\{F_2^b\}, F_2^b)$, $(\{F_1^b\}, ev_4)$, $(\{F_2^b\}, ev_5)$, $(\{ev_4\}, F_1^b)$, $(\{ev_5\}, F_2^b)$, $(\{ev_1, ev_5\}, F_1^b)$,$(\{ev_3, ev_5\}, F_1^b)$, $(\{ev_2, ev_4\}, F_2^b)\})$ associated with our ADF and specialized for the cc–semantics. We can see it depicted in Figure 120b.

The admissible extensions of SF^D are $\emptyset$, $\{ev_1\}$, $\{ev_3\}$, $\{ev_4\}$, $\{ev_1, ev_3\}$, $\{ev_1, ev_4\}$, $\{ev_3, ev_4\}$ and $\{ev_1, ev_3, ev_4\}$. By looking at the pd–sets and pd–sequences of our evaluations, we can see that these extensions correspond to the sets $\emptyset$, $\{d\}$, $\{b, d\}$, and $\{a, b, d, e\}$, which are the cc–admissible extensions of D. The complete extensions of SF^D are $\{ev_1, ev_3\}$ (associated with $\{b, d\}$) and $\{ev_1, ev_3, ev_4\}$ (associated with $\{a, b, d, e\}$), with the first one being grounded and the other preferred. These are indeed the desired answers and we retrieve all and only extensions of D.

12.2.2 AA Semantics

When we presented translations from ADFs to AFs, we have noted that out of all of the approaches, the aa–family is perhaps the simplest. This is also the family for which we can construct an attack propagation translation, resembling the ones we have created in case of AFNs and EASs (Translations 64 and 74). Just like in the previous cases we need to assume we are working with a weakly valid ADF. However, it can be easily chained with the weakly valid normal form translation (Translation 9) in order to lift this condition, and in this case we will omit further chain analysis.

Translation 88. *Let $D = (A, C)$ be a weakly valid ADF. Its corresponding SETAF is $SF^D_{AA} = (A, R)$, where $R = \{(E, a) \mid E \subseteq A$ is a minimal set of arguments s.t. for every acyclic pd–evaluation (F, B) for a on A, $E \cap B \neq \emptyset\}$.*

Due to the minimality assumption in the construction of the attack relation any target SETAF will be in minimal normal form:

Theorem 12.9. *Let $D = (A, C)$ be an ADF and $SF^D_{AA} = (A, R)$ its corresponding SETAF obtained through Translation 88. SF^D_{AA} is in minimal normal form.*

Just like in the case of AFNs and EASs, we can observe that the pd–acyclic conflict–free and admissible semantics are preserved only one way. Even though defending an argument implies defending at least one of its evaluations, it is only the complete semantics that forces the acceptance of the defended elements:

Theorem 12.10. *Let $D = (A, C)$ be a weakly valid ADF and $SF^D_{AA} = (A, R')$ its associated SETAF created through Translation 88. If $E \subseteq A$ is pd–acyclic conflict–free (aa–admissible) in D, then it is conflict–free (admissible) in SF^D_{AA}. Not every conflict–free (admissible) extension of SF^D_{AA} is pd–acyclic conflict–free (aa–admissible) in D. $E \subseteq A$ is an aa–complete (aa–preferred, acyclic grounded, stable) extension of D iff it is a complete (preferred, grounded, stable) extension of SF^D_{AA}.*

Redefinition of Translation 88: *Let WV^{ADF} be the collection of all weakly valid ADFs and Min^{SETAF} the collection of all SETAFs in minimal form, both on domain $\mathcal{U}$. The translation $aa\text{-}Tr^{ADF}_{SETAF} : WV^{AFN} \rightarrow Min^{SETAF}$ is defined as $aa\text{-}Tr^{ADF}_{SETAF}((A, C) = (A, R)$ for a framework $(A, C) \in WV^{AFN}$, where $R = \{(E, a) \mid E \subseteq A$ is a minimal set of arguments s.t. for every acyclic pd–evaluation (F, B) for a on A, $E \cap B \neq \emptyset\}$.*
Redefinition of Theorem 12.10: *Let $\sigma^{ADF} \in \{$aa–complete, aa–preferred, acyclic grounded, stable$\}$ be an ADF semantics, $\sigma^{AF} \in \{$complete, preferred, grounded, stable$\}$ a similar AF semantics and SC^{Tr}_{σ} the identity casting functions for σ. The translation*

$aa\text{-}Tr^{ADF}_{SETAF}$ is strong and semantics bijective under $(\sigma, SC^{Tr}_{\sigma})$. It is $\subseteq$–weak under the (pd–acyclic) conflict–free and (aa–) admissible semantics and identity casting functions.

Analysis of Translation 88: Under the (pd–acyclic) conflict–free, (aa–) admissible, (aa–) complete, (aa–) preferred, (acyclic) grounded and stable semantics and their identity casting functions, the translation $aa\text{-}Tr^{ADF}_{SETAF}$ is:

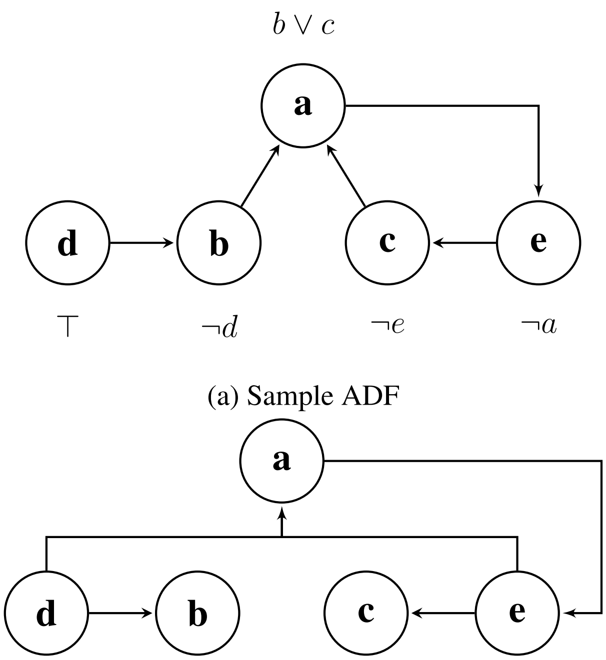
- source–subclass, target–subclass, overlapping
- argument domain preserving, argument set preserving, relation introducing and relation removing
- generic and semantics domain preserving
- semantical

Translation $aa\text{-}Tr^{ADF}_{SETAF}$ is neither $\oplus$ nor $\otimes$–modular. It is exact under the (aa–) complete, (aa–) preferred, (acyclic) grounded and stable semantics and the identity casting functions. We classify this approach as an attack propagation translation.

Explanation. Due to the fact that we work with weakly valid ADFs and produce only minimal SETAFs, our translation is both source and target–subclass. Moreover, it is overlapping. We can consider two simple frameworks $(\{a, b\}, \{C_a = \top, C_b = \top\})$ and $(\{a, b\}, \{C_a = \top, C_b = a\})$; in both cases, the associated SETAF will be simply $(\{a, b\}, \emptyset)$. This also exemplifies the relation removal occurring in the translation. However, it can also be seen as relation introducing. The sets of attacking arguments are based on the blocking sets of the evaluations. This means that an attacking argument is related to the attacked one in some way, but not necessarily in a direct manner. From the redefinitions of our translation and the semantics theorem we can easily observe that both types of domain are preserved. Moreover, as four semantics are handled in a strong manner, we can classify our method as generic. The use of acyclic pd–evaluations makes it semantical.

Let us now focus on $\oplus$–modularity and consider two ADFs $D_1 = (\{a, b, c\}, \{C_a = \top, C_b = \neg a, C_c = b\})$ and $D_2 = (\{d, e, c\}, \{C_d = \top, C_e = \neg d, C_c = e\})$. The associated SETAFs are $SF_1 = (\{a, b, c\}, \{(a, b), (a, c)\})$ and $SF_2 = (\{d, e, c\}, \{(d, e), (d, c)\})$. However, the SETAF associated with the framework $D_1 \oplus D_2 = (\{a, b, c, d, e\}, \{C_a = \top, C_b = \neg a, C_c = b \vee e, C_d = \top, C_e = \neg d\})$ is $(\{a, b, c, d, e\}, \{(a, b), (d, e), (\{a, d\}, c)\})$. We can observe that this structure is different from $SF_1 \cup SF_2$, thus making our approach not $\oplus$–modular.

Finally, let us analyze $\otimes$–modularity. Assume we have two frameworks $D_3 = (\{a, b, c\}, \{C_a = \top, C_b = \neg a, C_c = \top\})$ and $D_4 = (\{a, b, c\}, \{C_a = \top, C_b = \top, C_c = b\})$. The SETAFs created for them would be $SF_3 = (\{a, b, c\}, \{(a, b)\})$ and $SF_4 = (\{a, b, c\}, \emptyset)$. Unfortunately, the SETAF for the structure $D_3 \otimes D_4 = (\{a, b, c\}, \{C_a = \top, C_b = \neg a, C_c = b\})$ is $(\{a, b, c\}, \{(a, b), (a, c)\})$. We can easily observe it contains an attack which is present neither in SF_3 nor in SF_4. Consequently, our translation cannot be classified as $\otimes$–modular. ∎



(a) Sample ADF

(b) Associated SETAF for aa–semantics

Figure 121: Sample ADF and its associated SETAF for aa-semantics

Example 146. Let us consider the framework $D = (\{a, b, c, d, e\}, \{C_a = b \vee c, C_b = \neg d, C_c = \neg e, C_d = \top, C_e = \neg a\})$ depicted in Figure 121a. We can observe it is the weakly valid form of the ADF analyzed in Example 126. We can observe that every argument posseses a single minimal acyclic evaluation; we can create $((b, a), \{d\})$ and $((c, a), \{e\})$ for a, $((b), \{d\})$ for b, $((c), \{e\})$ for c, $((d), \emptyset)$ for d and $((e), \{a\})$ for e. The aa–admissible extensions of D are $\emptyset$, $\{d\}$, $\{a, c\}$, $\{d, e\}$ and $\{a, c, d\}$. Only $\emptyset$ and $\{a, c\}$ are not aa–complete. The acyclic grounded extension of our framework is $\{d\}$, while $\{d, e\}$ and $\{a, c, d\}$ are aa–preferred and stable.

Let us now construct the SETAF associated with our ADF. The set of arguments stays the same and it is the attack relation we need to focus on. By looking at the blocking sets, we can observe that argument d will not be attacked at all. In order to attack all of the evaluations of a, we need both d and e to be present. The other conflicts are quite straightforward as the arguments in question possess only one evaluation; d attacks b, e attacks c and a attacks e. We thus obtain the framework $SF^D = (\{a, b, c, d, e\}, \{(\{a\}, e)$, $(\{d\}, b)$, $(\{e\}, c)$, $(\{d, e\}, a)\})$ from Figure 121b.

The admissible extensions of SF^D are $\emptyset$, $\{a\}$, $\{d\}$, $\{a, c\}$, $\{a, d\}$, $\{d, e\}$, and $\{a, c, d\}$. Although every aa–admissible extension of D is admissible in SF^D, our target framework produces more sets than desired, as usual in the pure attack propagation approach. Nevertheless, when we reach the complete extensions – $\{d\}$, $\{d, e\}$ and $\{a, c, d\}$ – we obtain an exact correspondence between the answers produced by D and SF^D. We can observe that

$\{d\}$ is grounded SF^D, while $\{d, e\}$ and $\{a, c, d\}$ are preferred and stable. These answers are in agreement with the acyclic grounded, aa–preferred and stable sets of D.

12.2.3 Improvements

The results we have presented in this section were somewhat modest, particularly taking into account the amount of the ADF semantics and the previously created attack propagation and defender methods for frameworks with support which used SETAFs as the target frameworks. Although we would like to look further at these approaches for ADFs in the future, in this section we would like to ask ourselves one question. Namely, it was not possible to create an exact translation from ADFs to AFs under any of the known semantics - is it perhaps possible for SETAFs then?

Although we cannot give a definite answer due to the lack of study of semantics signatures in SETAFs, we can already make several observations. Let us first consider the conflict–free semantics; let $SF = (A, R)$ be a SETAF and $E \subseteq A$ a set of arguments. We can observe that if E is conflict–free in SF, then so is any subset of E. Consequently, even though we do not know the sufficient conditions for the SETAF conflict–free signature, being downward–closed is necessary:

Proposition 12.11. *Let $SF = (A, R)$ be a SETAF. The set of conflict–free extensions $cf(SF)$ of SF is downward–closed.*

Unfortunately, this is not the case in ADFs. Let us consider a simple framework $D_1 = (\{a, b, c, d\}, \{C_a = \neg b \vee \neg c \vee d, C_b = \neg a \vee \neg c \vee d, C_c = \neg a \vee \neg b \vee d, C_d = \top\})$. D_1 is an AADF$^+$ and a BADF and has the same conflict–free and pd–acyclic conflict–free extensions. Among them are $\{a, b\}$, $\{b, c\}$, $\{a, c\}$ and $\{a, b, c, d\}$, but not $\{a, b, c\}$. Thus, the downward closure is clearly violated.

Theorem 12.12. *Let Fr^{ADF} be the collection of all ADFs, $BADF$ the collection of all BADFs and $AADF^+$ of all AADF$^+$s, all on a domain $\mathcal{U}^{ADF}$. Let Fr^{SETAF} the collection of all SETAFs on a domain $\mathcal{U}^{AF}$. There exists no full (resp. source–subclass) translation from Fr^{ADF} (resp. $BADF$, $AADF^+$) to Fr^{SETAF} that is exact under conflict–free and pd–acyclic conflict–free semantics and identity casting functions for them.*

Let us now focus on admissibility. Although SETAF admissible extensions are not adm–closed in the sense of Definition 2.176, a somewhat similar property holds, though of course adapted to group conflict:

Proposition 12.13. *Let $SF = (A, R)$ be a SETAF and $E, E' \subseteq A$ two admissible extensions of SF. If there are no $b \in E, B' \subseteq E'$ s.t. $B'Rb$ and no $b' \in E', B \subseteq E$ s.t. BRb', then $E \cup E'$ is admissible in SF.*

With this property at hand, the most conclusive results can be given concerning the ca–admissible semantics of ADFs. Let us consider a simple modification of our previous framework which changes d into a self–supporter: $D_2 = (\{a, b, c, d\}, \{C_a = \neg b \vee \neg c \vee$

$d, C_b = \neg a \vee \neg c \vee d, C_c = \neg a \vee \neg b \vee d, C_d = d\})$. Its ca$_1$ and ca$_2$–admissible extensions are $\emptyset$, $\{d\}$, $\{a, d\}$, $\{b, d\}$, $\{c, d\}$, $\{a, b\}$, $\{a, c\}$, $\{b, c\}$, $\{a, b, d\}$, $\{b, c, d\}$, $\{a, c, d\}$ and $\{a, b, c, d\}$. Let us now think about the SETAF capable of producing such admissible sets. We can observe that $\{a, b, c\}$ is not in the collection, despite the fact that $\{a, c\}$, $\{a, b\}$ and $\{b, c\}$ are. However, due to the admissibility of $\{a, b, c, d\}$, it cannot be the case that there is any conflict between these sets. Consequently, $\{a, b, c\}$ would have to be admissible by the above proposition. This brings us to the following result (please note that D_2 is not an AADF$^+$):

Theorem 12.14. *Let Fr^{ADF} be the collection of all ADFs and $BADF$ the collection of all BADFs, both on a domain $\mathcal{U}^{ADF}$. Let Fr^{SETAF} the collection of all SETAFs on a domain $\mathcal{U}^{AF}$. There exists no full (resp. source–subclass) translation from Fr^{ADF} (resp. $BADF$) to Fr^{SETAF} that is exact under (ca$_1$, ca$_2$) admissible semantics and associated identity casting functions.*

Please note that due to the correspondence between ca$_2$–admissible extensions and admissible labelings (see Theorem 2.166), this result can be extended to the labelings as well.

The lack of exactness also holds for the ca$_1$ and ca$_2$–complete semantics. We can recall that SETAF complete extensions form a complete semilattice (Theorem 2.24), while the ADF ones might not (Theorem 2.158). A similar issue arose when we were considering translations from EAF(C)s to other argumentation frameworks. Due to the way how EAF(C)s are handled by ADFs, we can repeat the analysis done e.g. in Section 8.3.2 and show the following:

Theorem 12.15. *Let Fr^{ADF} be the collection of all ADFs and $BADF$ the collection of all BADFs, both on a domain $\mathcal{U}^{ADF}$. Let Fr^{SETAF} the collection of all SETAFs on a domain $\mathcal{U}^{AF}$. There exists no full (resp. source–subclass) translation from Fr^{ADF} (resp. $BADF$) to Fr^{SETAF} that is exact under (ca$_1$, ca$_2$) complete semantics and associated identity casting functions.*

Finally, the model extensions of ADFs cannot be handled by the stable ones in SETAFs, and the extensions associated with the preferred labelings can create collections that will not be produced by any SETAFs. The reason in both cases is the fact that stable and preferred semantics in SETAFs always produce incomparable sets, which is not the case in ADFs. As already seen in Examples 28 and 29, both model and labeling–induced preferred extensions might be comparable w.r.t. $\subseteq$.

Theorem 12.16. *Let Fr^{ADF} be the collection of all ADFs and $BADF$ the collection of all BADFs, both on a domain $\mathcal{U}^{ADF}$. Let Fr^{SETAF} the collection of all SETAFs on a domain $\mathcal{U}^{AF}$. There exists no full (resp. source–subclass) translation from Fr^{ADF} (resp. $BADF$) to Fr^{SETAF} that is exact under (model) stable and (labeling) preferred semantics and associated identity casting functions.*

Despite all these negative results, we did manage to obtain an exact translation for the aa–complete, aa–preferred and stable semantics with the attack propagation translation. This gives us a certain hope when considering the ac and cc approaches. However, the strategies used in the AF translation cannot be directly reused due to the fact that a given argument can, at the same time, posses an evaluation that cannot be made acyclic and one that can. In other words, while we can propagate the conflicts using standard evaluations, we need to know how an argument is derived in order to introduce an appropriate self–attack or a breaker argument. Therefore, we are forced to alter the domain and store such information, which pushes us away from exactness and brings us closer to the existing translations. Consequently, it may appear more worthwhile to first analyze the precise signatures of the semantics in question, so that we know whether it is even possible to improve our approaches.

The reason why we have not considered the defender transformations in the style of Translations 65 and 75 is the fact that in contrast with the powerful and evidential sequences, ADF evaluations carry additional information in the form of the blocking sets. Consequently, a given pd–sequence can appear in more than one evaluation, and the construction of appropriate attacks can turn out to be complicated. A possible way to address it by using the target argument domain as in the Translation 89 that we will present in the next section, though unfortunately this method again does not give us exact conversions. Therefore, we would like to analyze this issue further in the future.

12.3 ADF as AFN

Our previous translations followed the coalition approach, i.e. the aspects of ADFs that were not directly expressible within (SET)AFs were hidden away in the structure of argument. Additionally, this structure also held certain information relevant for the construction of the target framework, but not for extracting the source extensions. In this section we would like to see how AFNs handle ADFs, particularly that both of the frameworks handle, this way or the other, a certain notion of support between arguments.

Due to the design similarities w.r.t. the AFN semantics, our analysis will be focused on the aa–family as the source semantics (see e.g. Section 10.6). Among the things that ADFs can express are group attacks and a form of support we referred to as "overpowering", which corresponded to defense attacks from EAF(C)s (see Section 8.6.1). Unfortunately, due to the best of our knowledge, this is something AFNs cannot handle directly (see Section 8.5). Moreover, the interplay of various relations can create a (functional) condition from which it is somewhat more difficult to extract what is the exact nature of the incoming links. Consequently, what we propose to do is to create multiple representations of a single ADF argument, each associated with a minimal decisively in interpretation, from which the extraction of support and attack is straightforward. The additional advantage of this approach is the fact that building a powerful sequence on the AFN side produces a clear pd–function and associated evaluation on the ADF side. However, this assumes that the arguments we work with actually possess decisively in interpretations, which is not necessarily the case in ADFs. Thus, the ADFs we will work with need to be in cleansed form. However, please note that changing arguments with conditions equivalent to falsum into self–supporters would have also worked. The drawback of this solution is that, should we be able to retrieve the original ADF from such an approach, the framework would behave differently under e.g. ca, ac and cc families of semantics, which is not the case when we use the cleansed form (see Section 4.2). This brings us to the following approach:

Translation 89. *Let $D = (A, C)$ be an ADF in cleansed normal form. The corresponding AFN $FN^D_{AA} = (A', R', N')$ is created as follows:*

- *let a be an argument in A and $min\_dec(in, a)$ the set of its minimal decisively in interpretation. For any interpretation $v_a \in min\_dec(in, a)$ add the pair (a, v_a) to A',*
- *let (a, v_a) be an argument in A' and b an argument in A s.t. $v_a(b) = \mathbf{t}$. Let $B = \{(b, v_1), ..., (b, v_m)\}$ be the collection of all arguments in A' corresponding to b. Add $(B, (a, v_a))$ to N', and*
- *let (a, v_a) be an argument in A' and b an argument in A s.t. $v_a(b) = \mathbf{f}$. Let $B = \{(b, v_1), ..., (b, v_m)\}$ be the collection of all arguments in A' corresponding to b. For every (b, v_i), add $((b, v_i), (a, v_a))$ to R'.*

We can observe that the group nature of support in AFNs is used to handle the fact that a single source argument present in the $\mathbf{t}$ part of a given interpretation can be represented by multiple arguments in the target AFN and that the presence of any of them is sufficient. Thus, as such, it is not used to handle the actual group support that can be expressed in ADFs (more on it in Section 12.3.1).

Let us now see how the produced AFNs will look like. Due to the way the support and attack relations are constructed, it will always be in minimal and consistency norms, independently of the forms of the source ADF. Moreover, the translation will preserve the validity forms:

Theorem 12.17. *Let $D = (A, C)$ be a cleansed form ADF and $FN^D_{AA} = (A', R', N')$ its corresponding AFN obtained through Translation 89. Then, FN^D_{AA} is in minimal and (strongly) consistent normal forms. FN^D_{AA} might not be weakly valid if D is weakly valid. If D is relation valid, then FN^D_{AA} is weakly and relation valid. If D is strongly valid, then so is FN^D_{AA}.*

We can now observe that the aa–family of ADF semantics and the semantics of AFNs are closely related. We manage to retrieve all and only extensions of ADFs starting from pd–acyclic (coherent) extensions. However, due to the fact that every ADF argument can be represented by a number of AFN ones, the relations between the answers of both frameworks become one–to–one only starting with the complete semantics.

Theorem 12.18. *Let $D = (A, C)$ be a cleansed form ADF and $FN^D_{AA} = (A', R', N')$ its corresponding AFN obtained through Translation 89. If $S = \{(a_1, v_{a_1}), ..., (a_n, v_{a_n})\} \subseteq A'$ is a coherent (strongly coherent, admissible, preferred, complete, grounded, stable) extension of FN^D_{AA}, then $S' = \bigcup_{i=1}^{n}\{a_i\}$ is a pd–acyclic (pd–acyclic conflict–free, aa–admissible, aa–preferred, aa–complete, acyclic grounded, stable) extension of D.*

If $S' \subseteq A$ is a pd–acyclic (pd–acyclic conflict–free, aa–admissible, aa–preferred, aa–complete, acyclic grounded, stable) extension of D, then there exists a coherent (strongly coherent, admissible, preferred, complete, grounded, stable) extension $S = \{(a_1, v_{a_1}), ..., (a_n, v_{a_n})\} \subseteq A'$ of FN^D_{AA} s.t. $S' = \bigcup_{i=1}^{n}\{a_i\}$.

Let us now put this translation into our system. Although we can easily analyze our approach using the properties we have defined in Section 3.2, the methodology behind this translation escapes our classification system. The approach is not particularly difficult and therefore exhibits some of the basic behaviors. However, the severe domain changes and the amount of possible arguments that need to be introduced means that our approach is more likely a hybrid. Nevertheless, the idea behind this translation is neither coalition, nor attack propagation nor defender. Consequently, at this point we choose not to classify this translation and treat it as a reminder that our system can still be improved in the future.

Redefinition of Translation 89: *Let Cln^{ADF} be the collection of all cleansed ADFs on domain $\mathcal{U}$ and $Min^{AFN} \cap SCons^{AFN}$ the collection of all minimal and strongly consistent AFNs on domain $AI^{\mathcal{U}} = \{(a, v_a) \mid a \in \mathcal{U},\ v_a$ is an interpretation over $E \subseteq \mathcal{U}\}$. The translation $aa\text{-}Tr^{ADF}_{AFN} : Cln^{ADF} \rightarrow Min^{AFN} \cap SCons^{AFN}$ is defined as $aa\text{-}Tr^{ADF}_{AFN}((A, L, C)) = (A', R', N')$ for a framework $(A, L, C) \in Cln^{ADF}$, where $A' = \{(a, v_a) \mid a \in A, v_a \in min\_dec(in, a)\}$, $R' = \{((a, v_a), (b, v_b)) \mid a, b \in A, v_b(a) = \mathbf{f}\}$ and $N' = \{(B, (a, v_a)) \mid a, b \in A, v_a(b) = \mathbf{t}$ and $B = \{(b, v_1), ..., (b, v_m)\}$ is the collection of all arguments in A' for $b\}$.*

Redefinition of Theorem 12.18: *Let $\sigma^{ADF} \in \{$pd–acyclic, pd–acyclic conflict–free, aa–admissible, aa–complete, aa–preferred, acyclic grounded, stable$\}$ be an ADF semantics and $\sigma^{AFN} \in \{$coherent, strongly coherent, admissible, complete, preferred, grounded, stable$\}$ a similar AFN semantics. Let SC^{Tr}_{σ} be the extraction semantics casting function for σ defined as $SC^{X}_{\sigma}(S) = \bigcup_{i=1}^{n}\{a_i\}$, where $X = (A, C) \in Fr^{ADF}$ and $S = \{(a_1, v_1), ..., (a_n, v_n)\} \in \sigma(aa\text{-}Tr^{ADF}_{AFN}(X))$. The translation $aa\text{-}Tr^{ADF}_{AFN}$ is strong under $(\sigma, SC^{Tr}_{\sigma})$. It is also semantics bijective under the (aa–) complete, (aa–) preferred, (acyclic) grounded and stable semantics and the defined casting functions.*

Analysis of Translation 89: Under the (pd–acyclic) coherent, (pd–acyclic conflict–free) strongly coherent, (aa–) admissible, (aa–) complete, (aa–) preferred, (acyclic) grounded and stable semantics and the defined casting functions, the translation $aa\text{-}Tr^{ADF}_{AFN}$ is:

- source–subclass, target–subclass and overlapping
- argument domain altering, argument introducing, relation introducing, relation removing
- generic and semantics domain altering

- semi–structural

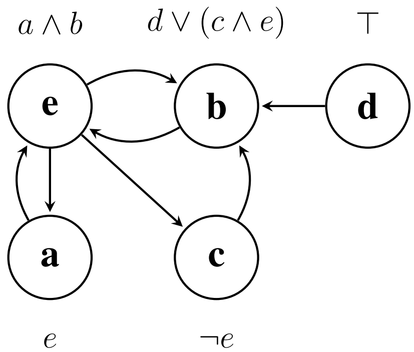
The translation is neither $\oplus$ nor $\otimes$–modular.

Explanation. Due to the fact that we work with the cleansed ADFs only, our translation is classified as source–subclass. We also choose to view it as target subclass due to the domain changes, i.e. not every combination of arguments from our domain represents a framework that can be obtained from an ADF. Our choice to use minimal decisively in interpretations in the construction also causes the conversion to be overlapping, i.e. two ADFs can produce the same interpretations without necessarily being identical, which is one of the aspects of the redundancy–free form (see Section 4.1.5). For this reason, the translation is also classified as relation removing. We also choose to see it as argument and relation introducing due to the multiple representations of a single ADF argument or link in the target AFN. We can easily observe that both semantics and argument domains are altered in our approach. Although it can be classified as structural or semi–structural, we are leaning towards the latter due to the minimality assumption on the chosen interpretations.

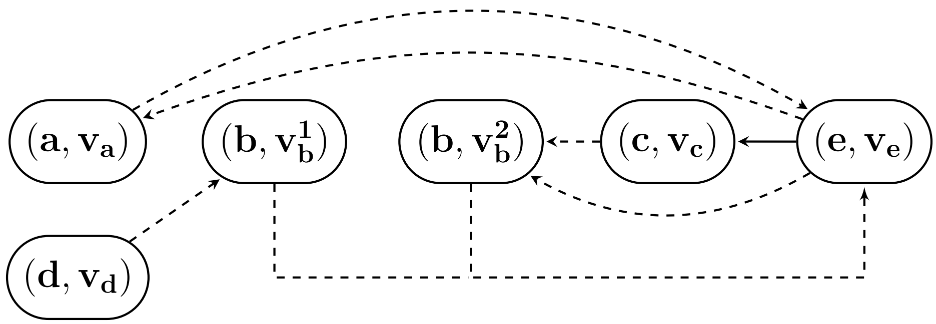
Despite the fact that the translation is not particularly complicated, it is not in any way modular, even if we consider only conflict–based ADFs. Let us consider two frameworks $D_1 = (\{a\}, \{C_a = \top, C_b = \top\})$ and $D_2 = (\{a, b\}, \{C_a = \neg a, C_b = \top\})$. Their associated AFNs are $FN_1 = (\{(a, v_1), (b, v_1)\}, \emptyset, \emptyset)$ and $FN_2 = (\{(a, v_2), (b, v_1)\}, \{((a, v_2), (a, v_2))\}, \emptyset)$, where v_1 is empty, and $v_2 = \{a : \mathbf{f}\}$. If we consider $\otimes$ modularity, the structure $D_1 \otimes D_2$ would be equivalent to D_2 and producing the AFN FN_2, not $FN_1 \cup FN_2$. If we consider $\oplus$–modularity, then we would obtain a framework $D_3 = (\{a\}, \{C_a = \top \vee \neg a, C_b = \top\})$ in which the (a, a) link is redundant. Thus, the associated AFN is FN_1, not $FN_1 \cup FN_2$. ■

Example 147. Let us look at the ADF $D = (\{a, b, c, d, e\}, \{C_a = e, C_b = d \vee (c \wedge e), C_c = \neg e, C_d = \top, C_e = a \wedge b\})$ depicted in Figure 122a and thoroughly analyzed in this section. The sets $\emptyset$, $\{c\}$, $\{d\}$, $\{b, d\}$, $\{c, d\}$ and $\{b, c, d\}$ are aa–admissible in D, with $\{b, c, d\}$ being the single aa–complete, aa–preferred, acyclic grounded and stable set. The analysis of this framework is interesting due to the cycles appearing in it.

The minimal decisively in interpretations in this framework are $v_a = \{e : \mathbf{t}\}$, $v_b^1 = \{d : \mathbf{t}\}$, $v_b^2 = \{c : \mathbf{t}, e : \mathbf{t}\}$, $v_c = \{e : \mathbf{f}\}$, $v_d = \emptyset$ and $v_e = \{a : \mathbf{t}, b : \mathbf{t}\}$. Therefore, we create the pairs (a, v_a), (b, v_b^1), (b, v_b^2), (c, v_c), (d, v_d) and (e, v_e) as the new arguments for our AFN. By looking at the $\mathbf{t}$ assignments, we can see that $\{(e, v_e)\}$ supports (a, v_a) and the other way around; the extra supporting set for (e, v_e) is $\{(b, v_b^1), (b, v_b^2)\}$. The pair (b, v_b^1) is supported only by $\{(d, v_d)\}$, while for (b, v_b^2) we have sets $\{(c, v_c)\}$ and $\{(e, v_e)\}$. Finally, neither (c, v_c) nor (d, v_d) requires any support. By looking at the $\mathbf{f}$ assignments in the interpretations, we can observe that (e, v_e) attacks (c, v_c). It is also the only attack in our framework. We therefore obtain the AFN $FN^D = (\{(a, v_a), (b, v_b^1), (b, v_b^2), (c, v_c), (d, v_d), (e, v_e)\}, \{((e, v_e), (c, v_c))\}, \{(\{(e, v_e)\}, (a, v_a)), (\{(a, v_a)\}, (e, v_e)), (\{(b, v_b^1), (b, v_b^2)\}, (e, v_e)), (\{(d, v_d)\}, (b, v_b^1)), (\{(c, v_c)\}, (b, v_b^2)), (\{(e, v_e)\}, (b, v_b^2))\})$, as visible in Figure 122b.



(a) Sample ADF



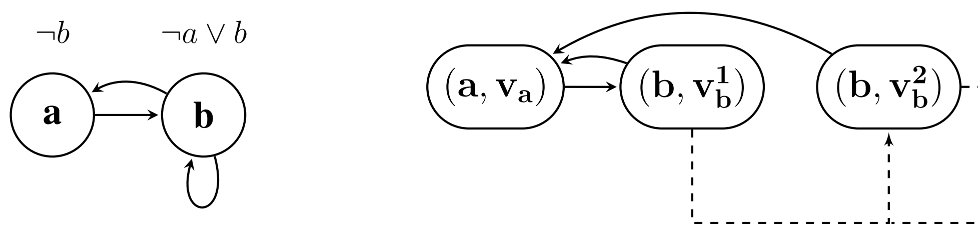
(b) Associated AFN

Figure 122: Sample ADF and its associated AFN

We only have three minimal powerful sequences in our AFN: $((c, v_c))$, $((d, v_d))$, $((d, v_d), (b, v_b^1))$. We can observe that the pairs representing a and e are correctly recognized as not possessing a powerful sequence at all. Due to this, no sequence for (b, v_b^2) can be produced either. Consequently, we can show that $\emptyset$, $\{(c, v_c)\}$, $\{(d, v_d)\}$, $\{(b, v_b^1), (d, v_d)\}$, $\{(c, v_c), (d, v_d)\}$ and $\{(b, v_b^1), (c, v_c), (d, v_d)\}$ are the admissible extensions of FN^D. We can observe that they indeed correspond to the aa–admissible sets of D. It is also easy to see that $\{(b, v_b^1), (c, v_c), (d, v_d)\}$ is the only complete, preferred, grounded and stable set of FN^D (please note that (e, v_e) possesses no powerful sequence). We can therefore retrieve all and only the extensions of D.

Example 148. In this example we would like to show that despite the relation between the two frameworks and their semantics, it can happen that an argument–interpretation pair accepted in the target AFN would, in fact, not be used by any of the acyclic pd–evaluations in the source ADF extension. Let us consider a simple ADF $D = (\{a, b\}, \{C_a = \neg b, C_b = \neg a \vee b\})$ visible in Figure 123a. We can observe that b has the power to override the attack from a through the use of a support cycle. The minimal decisively in interpretation for our arguments are $v_a = \{b : \mathbf{f}\}$, $v_b^1 = \{a : \mathbf{f}\}$ and $v_b^2 = \{b : \mathbf{t}\}$. The (minimal) acyclic evaluations for this framework are $((a), \{b\})$ and $((b), \{a\})$; we can observe that the interpretation v_b^2 will be not be used. Our aa–admissible extensions are $\emptyset$, $\{a\}$ and $\{b\}$; all of them are also aa–complete. $\emptyset$ is our acyclic grounded extension, while $\{a\}$ and $\{b\}$ are both aa–preferred and stable.

The AFN associated with our ADF is now $FN^D = (\{(a, v_a),\ (b, v_b^1),\ (b, v_b^2)\}, \{((a, v_a), (b, v_b^1)), ((b, v_b^1), (a, v_a)), ((b, v_b^2), (a, v_a))\}, \{(\{(b, v_b^1), (b, v_b^2)\}, (b, v_b^2))\})$, as depicted in Figure 123b. The minimal powerful sequences for our arguments are $((a, v_a))$, $((b, v_b^1))$ and $((b, v_b^1), (b, v_b^2))$. We can therefore observe that the argument paired with the undesirable interpretation can in fact be accepted. However, it is only because another representation of this argument is "correct", and the pair that can be considered incorrect has to come back to it. Therefore, despite this, behaviour, we can observe that the admissible extensions $\emptyset$, $(\{(a, v_a)\}$, $\{(b, v_b^1)\}$ and $\{(b, v_b^1), (b, v_b^2)\}$ of FN^D do correspond to the aa–admissible extensions of D. The complete extensions of our AFN are $\emptyset$, $(\{(a, v_a)\}$ and $\{(b, v_b^1), (b, v_b^2)\}$, with the first being grounded and the other two preferred and stable. We thus retrieve all and only desired extensions of D.



(a) Sample ADF

(b) Associated AFN

Figure 123: Sample ADF and its associated AFN

12.3.1 Improvements

In this section we would like to discuss two possible directions in enhancing our translation; one concerning its strength, and the other regarding the size of the produced AFN.

In the presented translation we have focused on the aa–family of ADF semantics. We would like to know whether the results can be improved and possibly extended to the other families. To the best of our knowledge, it is rather unlikely that an exact translation from ADFs to AFNs can be created for admissible and preferred semantics, for the same reasons as in the SETAF–AFN case (see Sections 6.3.1 and 10.1.1). Due to the fact that SETAF–style ADFs are both BADFs and AADF$^+$s (see Section 6.5), this particular result is quite strong and holds for any of the ADF families. However, we cannot say anything definite yet concerning the stable semantics, as their AFN signature goes beyond the AF one (see Theorem 10.7). For reasons similar as in Section 12.2.3, we can also state that an exact translation from ADFs to AFNs under the (ca$_1$, ca$_2$) complete semantics and (model) stable semantics is not possible.

Despite the amount of negative results, there are still some ways in which the translation can be improved in the terms of the size of the produced AFNs. For now, we have exploited the presence of group support in AFNs in order to handle the multiple representations of a given argument. However, we can also try to take it a step further; ADFs can in a certain sense represent group support, as was visible e.g. in Translations 69 and 80. Nevertheless, the current ADF–AFN approach does not reflect that. The idea is to associate an argument not with a single decisively in interpretations, but with a number of them. We only require that the $\mathbf{f}$ parts, corresponding to attacks in AFNs, are the same among all of the interpretations. Let us look at the following example:

Example 149. We can consider the ADF $(\{a, b, c, d, e, f, g\}, \{C_a = \neg c, C_b = f \vee \neg c, C_c = \neg a, C_d = \top, C_e = g \cap (a \vee b) \wedge (\neg c \vee \neg d), C_f = f, C_g = \top\})$ depicted in Figure 124. The minimal decisively in interpretations for our arguments are $v_a = \{c : \mathbf{f}\}$, $v_b^1 = \{f : \mathbf{t}\}$, $v_b^2 = \{c : \mathbf{f}\}$, $v_c = \{a : \mathbf{f}\}$, $v_d = v_g = \emptyset$, $v_e^1 = \{a : \mathbf{t}, c : \mathbf{f}\}$, $v_e^2 = \{b : \mathbf{t}, c : \mathbf{f}\}$, $v_e^3 = \{a : \mathbf{t}, d : \mathbf{f}\}$, $v_e^4 = \{b : \mathbf{t}, d : \mathbf{f}\}$ and finally $v_f = \{f : \mathbf{t}\}$. By following Translation 89, we would create as many representations of a given argument as many interpretations it possesses. The supporting sets for e.g. the pair (e, v_e^1) would be $\{(g, v_g)\}$ and $\{(a, v_a)\}$, while with (e, v_e^2) we would associate $\{(g, v_g)\}$ and $\{(b, v_b^1), (b, b_b^2)\}$. In both of these cases, the attacker would be (c, v_c). Consequently, we can choose to combine these two representations and obtain an argument $(e, \{v_e^1, v_e^2\})$, attacked by (c, v_c) and supported by the sets $\{(g, \{v_g\})\}$ and $\{(a, \{v_a\}), (b, \{v_b^1\}), (b, \{v_b^2\})\}$. We can observe that a representation of an ADF argument in the original AFN is accepted if and only if a representation of this argument is also accepted in the grouped AFN.

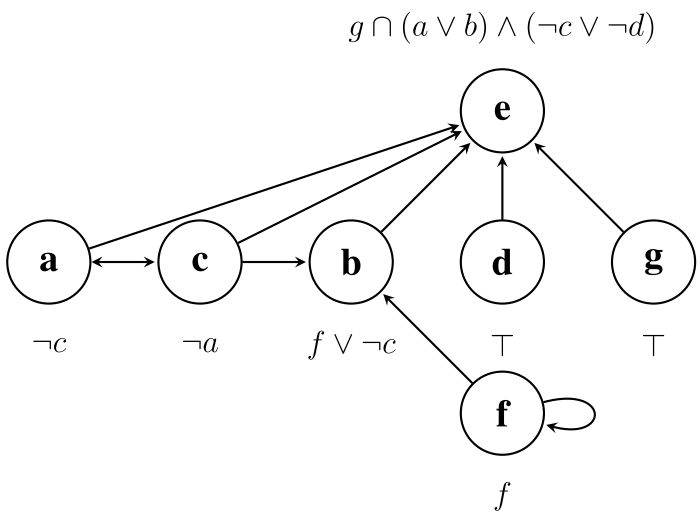


Figure 124: Sample ADF

The way the positive parts of a collection of decisively in interpretations are combined into supporting sets resembles the EAS–AFN Translation 78; the occurrences of the arguments are then replaced by their representations in the target AFN. Since we are interested in this method in particular due to the fact that it may produce smaller target frameworks than Translation 82, we will use the minimality assumption in the construction of our sets, particularly that we also assumed it in the choice of our decisively in interpretations. The correctness and the properties of this approach can be proved similarly as in the case of Translation 89. Thus, we close this section with the following proposal:

Translation 90. ***Sketch*** *Let $D = (A, C)$ be a cleansed form ADF. The corresponding grouped AFN $g - FN_{AA}^{D} = (A', R', N')$ is created as follows:*

- *let a be an argument in A and $min\_dec(in, a)$ the set of its minimal decisively in interpretation. Let $v' \subseteq min\_dec(in, a)$ be the collection of all and only minimal decisively in interpretation s.t. for every $v_1, v_2 \in v'$, $v_1^{\mathbf{f}} = v_2^{\mathbf{f}}$. For every such v', add (a, v') to A',*

- *let (a, v') be an argument in A' and $E \subseteq \bigcup_{v \in v'} v^{\mathbf{t}}$ a minimal set of ADF arguments s.t. $\forall v \in v', E \cap v^{\mathbf{t}} \neq \emptyset$. Let $B \subseteq A'$ be the collection of all arguments $(b, v_b) \in A'$ s.t. $b \in E$. Add $(B, (a, v'))$ to N', and*

- *let (a, v') be an argument in A' and b an argument in A s.t. $\exists v \in v', v(b) = \mathbf{f}$. Let $B = \{(b, v_1), ..., (b, v_m)\}$ be the collection of all arguments in A' corresponding to b. For every (b, v_i), add $((b, v_i), (a, v'))$ to R'.*

12.4 ADF as Other Frameworks

So far we have focused only on three frameworks, namely AFs, SETAFs and AFNs, thus omitting AFRAs, EAFs, BAFs and EASs. Although we are not convinced that the recursive attack can supersede the binary attack when it comes to handling ADFs, we believe that the defense attack holds certain potential. This is in particular due to the lack of monotonicity of the EAF semantics, which makes it more suitable for handling the ca_1 and ca_2–families than any of the other structures we have considered in this work. The nature of defense attacks might also help us to avoid the need for blocking sets or other false mappings to be stored within the target arguments, though we would still need to repeat a lot of the constructions from AFs and SETAFs in order to “sneak in” the support.

The translation from ADFs to BAFs would be, in a sense, more problematic than to AFNs. Although this is the only framework with support apart from ADFs that has semantics permitting the presence of support cycles, their handling is somewhat different. This is mostly due to the lack of the support ingredient in the definition of defense, which was more or less explicitly present in any other framework with support. Consequently, in a framework consisting only of two arguments a and b supporting each other, both a and b would be considered defended by the empty set in BAFs, but not decisively in w.r.t. the standard range that is used in the cc–semantics. Thus, we would have to limit ourselves to working with the strongly valid ADFs despite the fact that we would not be focusing on acyclic semantics.

Finally, we can consider the evidential systems, which due to the presence of group support and attack are, in a certain sense, closer to ADFs than AFNs. However, the current approach would be very similar to merging the ADF–AFN and AFN–EAS methods, i.e. we would still require the presence of decisively in interpretations in the arguments and connect the support from the evidence to those elements that have interpretations with empty $\mathbf{t}$ part. The possible improvements would employ the approach from Translation 90, though this time the grouping could be extended to $\mathbf{f}$ assignments as well as the $\mathbf{t}$ ones, thus providing little insight. The most important issue here is the fact that the signature of EAS semantics are not established. Since we believe them to be more admitting than in the case of AFNs, the knowledge about them would allow us to state what can or cannot be done, thus possibly pointing us to a more efficient approach. Therefore, we hope to investigate this translation more in the future.

12.5 Summary

With the exception of the ADF–SETAF Translation 88, in all of the approaches we have demonstrated the argument domain had to undergo quite significant modifications. Consequently, these methods did not go beyond the usual strength. However, the fact that they are often semantics bijective gives us hope for creating faithful approaches in the future. Unfortunately, in some of the cases we have already shown that an exact translation will not be possible.

Although the ADF–AF translations are the most developed ones and can handle the majority of the ADF semantics, the simplicity of the aa–family allowed us to create two interesting approaches. The attack propagation ADF–SETAF method turned out to be exact for most of the aa–semantics, though at the price of being semantical. Although it assumed that the source frameworks are weakly valid and is thus classified as source subclass, it is easy to see that an appropriate normal form translation could be used to address that. Although the ADF–AFN translation was only strong, it did fall into the semi–structural category. The results of our work can be seen in Table 15. Please observe that, with the exception of the ca_2–semantics that shares the same translation with the labeling–based approaches, the results we have reported concerned the family of semantics for which a given transformation was designed. This means that, if one wishes, the table could be further filled with weaker results by using the relations between the given families stated in Sections 2.3.6 and 2.3.7.

Table 15: Translations from ADFs to other frameworks

Properties		AF					SETAF		AFN
Translation		AA	AC	CC	CA_2	LAB	CC	AA	AA
		82	83	85	86		87	88	89
Strength	cf			strong	strong		strong		
	acy-cf	strong	strong					$\subseteq$-weak	strong
	adm	strong	strong	strong	strong	$\subseteq$-weak	strong	$\subseteq$-weak	strong
	comp	strong	strong	strong	$\subseteq$-weak	strong	strong	exact	strong
	pref	strong	strong	strong	$\subseteq$-weak	strong	strong	exact	strong
	grd		strong	strong	strong	strong	strong		
	acy-grd	strong						exact	strong
	stb	strong						exact	strong
	mod				strong				
Functional		full	full	full	full		full	source–subclass	source–subclass
		target–subclass	target–subclass	target–subclass	target–subclass		target–subclass	target–subclass	target–subclass
		overlapping	overlapping	overlapping	overlapping		overlapping	overlapping	overlapping
Syntactical		argument domain altering	argument domain altering	argument domain altering	argument domain altering		argument domain altering	argument domain preserving	argument domain altering
		argument removing	argument removing	argument removing	argument removing		argument removing	relation removing	argument introducing
		argument introducing	argument introducing	argument introducing	argument introducing		argument introducing	relation introducing	relation removing
		relation removing	relation removing	relation removing	relation removing		relation removing		relation introducing
		(possibly) relation introducing	relation introducing	(possibly induced) relation introducing	(possibly induced) relation introducing		(possibly induced) relation introducing		
Semantical		generic	generic	generic	generic		generic	generic	generic
		semantics domain altering	semantics domain altering	semantics domain altering	semantics domain altering		semantics domain altering	semantics domain preserving	semantics domain altering
Computational		semantical	semantical	semantical	semantical		semantical	semantical	semi–structural

13 Related Work

The research we have presented in this report consists both of new and the existing translations and their analysis. Consequently, the majority of the related works are not only referred to in the text, they are also studied and when necessary, completed or modified in order to address certain drawbacks. Our classification system provides an in–depth analysis and allows us to compare various approaches. Moreover, on many occasions we have discussed the possible improvements to the presented methods. Therefore, in order to see how a particular related work fits into our research, it suffices to go to the relevant part of this report. For these reasons, our related work section will be modest and will mostly provide pointers as to where a given paper is analyzed.

First, we would like to draw our attention to works that concern the properties of translations. Although our system is novel, we also wanted to include the attributes such as faithfulness, polynomiality or modularity in our analysis, which are commonly used to describe the transformations in nonmonotonic reasoning [52,55,57,58]. We have introduced them in Sections 3.2.3 and 3.2.4 as examples of semantical and computational properties. In particular, we have focused on the notion of faithfulness due to its varying interpretations available in the literature. As a result, we have adopted the distinction between the faithful and exact translations as presented in [42]. Please note that this work also contains the properties we have not considered in our system. In particular, some of them were defined precisely for situations when the source and target frameworks are of the same type and shifting them to a more general setting produced counterintuitive definitions. We have therefore focused more on introducing other properties in a similar spirit, particularly the syntactical ones in Section 3.2.2. Nevertheless, these results can be still relevant for the normal form translations, and we will consider extending our analysis in the future. Finally, we have also incorporated the notion of modularity for ADFs from [49] and proposed an equivalent definition aimed at ADFs in functional form.

The next lines of research we have drawn upon concern the types of translations. In Section 3.3 we have identified our four primary types; the basic, coalition, attack propagation and defender translations. To the best of our knowledge, no formal classifications of abstract argumentation translations have been introduced before. We are also not aware of any explicit comparisons between the motivation and patterns of the presented methods. However, they were already discussed separately; we have distinguished the coalition translations after the research in [28–30, 69, 73] and attack propagation after [30]. Although the defender translation was originally inspired by the discussion regarding the difference between support and defense done in [28], it has recently also appeared in [31]. Furthermore, we have come to realize that certain translations designed purely for attack–based frameworks, referred to as flattening or classified as meta–level argumentation, also fall into this category [18, 64]. For further discussion we refer the reader to Section 3.3.

Apart from the translations between different argumentation frameworks, we have distinguished the normal form transformations (see Section 4). The first form we have considered – the minimal normal form – comes from our previous work [77, 78]. We have also used the redundancy–free form for ADFs analyzed in [48]. The remaining results are, to the best of our knowledge, novel. The only other work we are aware of that also tried to tackle the normal forms is [35]. Nevertheless, it focuses solely on the Dung's framework and the presented results are relevant for computational and not translation purposes. We have therefore not recalled it in our report.

We finally come to the core of our report; the translations between different argumentation frameworks. We have drawn upon many existing works and tried to be thorough in their analysis. The first translation we have recalled concerns the AF–SETAF approach from [68] and its analysis can be found in Section 5.1. The next work we have focused on is [9]; it included the AF–AFRA and AFRA–AF methods (Sections 5.2 and 7.1) as well as the comparison between AFRAs and EAFs (Sections 7.2 and 8.4). Further works concerning these three frameworks are [18, 47, 64] and their analysis can be found in Sections 7.1.2 and 8.2. The work in [64] has also been discussed in Section 6.1.2 concerning one of the SETAF–AF translations.

The aforementioned works concerned the transformations between attack–based frameworks. However, in our research we wanted to go beyond these structures. Many of our approaches have been inspired by the works on BAFs [28–31], which themselves were analyzed in Sections 5.4, 9.1.1, 9.1.2 and 9.1.3. The results concerning the AF–AFN and AFN–AF translations, presented in [69], can be seen in Sections 5.5 and 10.1. In order to compare AFNs and EASs we have extended our previous research from [77, 78] (see Sections 10.5 and 11.4). Additionally, an interesting study on the relations between EASs, SETAFs and AFs can be found in [73]. Sections 5.6, 6.4 and 11.1 focus on presenting the results of this work.

Finally, we come to the translations concerning ADFs. Every transformation for this framework presented in this report, with the exception of the AF–ADF method from [21] (see Section 5.7), has been our own. There are, however, two works on translations for ADFs we have not analyzed here. The first one [20] presented various specialized translations, in particular for the stable, grounded and model semantics. Unfortunately, the stable semantics for ADFs have been later redefined and the approach for the model semantics has been shown to misbehave in certain cases [44]. An alternative method for translating ADF models into AF stable extensions has been proposed in [44], along with a possible way to transform ADFs into BADFs that was meant to preserve the conflict–free, model and grounded semantics. Both of these approaches were created for ADFs with propositional acceptance conditions only. Moreover, the construction relies purely on whether the condition is satisfied or not and not on how it was achieved. Consequentially, this approach does not deal with the issue of support cycles and cannot be straightforwardly extended to handle the families of the extension–based semantics we have introduced. Nevertheless, we hope that in the future we will improve this method in order to handle the functional ADFs as well as our new semantics.

14 Future Work & Conclusions

Abstract argumentation is a rich research area and a wide range of argumentation frameworks has been developed, from those that tackle different types of relations between arguments to structures that study components such as probabilities, preferences and strengths [22]. In our work, we have asked ourselves how different attack and support–

based argumentation frameworks are connected. We have thus established a number of translations between them in order to answer this question. However, the results we have obtained made us even more curious; we have observed certain patterns emerging and wanted to be able to compare them. We wanted to know what does it mean that a translation is "good", to be able to say which ones are and which are not, and to know if we can make them better.

All of these questions have lead to the creation of our in–depth compendium on the intertranslatability of argumentation frameworks consisting of almost ninety translations. We have proposed a number of new approaches as well as recalled and if necessary, extended, the existing ones. Furthermore, we have introduced a classification system for describing a given translation in terms of its functional, syntactical, semantical and computational properties. In this analysis we have also distinguished our four main types of transformations based on their underlying methodology – the basic, coalition, attack propagation and defender methods. The summary of our results for a given translation is visible in an appropriate table at the end of each section (Tables 7, 8, 9, 10, 11, 12, 13, 14 and 15). Finally, we have also studied the topic of normal form translations, which so far has received little attention. This has led to the introduction of various minimal, valid and consistent forms for both attack and support–based frameworks.

Whenever possible, we tried to show whether there is a chance for creating a, semantically speaking, stronger translations in the future. In particular, we have focused on the existence of exact and, if possible, full translations. We have obtained such a method in almost every case when we took AFs as the source frameworks. The only exceptions were the AF–AFRA and AF–EAS translations. Due to the nature of the AFRA semantics and the addition of evidence in EASs, it appears that we cannot create a generic exact method (see Sections 5.2.1 and 5.6.1).

In the case of translations from SETAFs to other structures, we have obtained only a single exact approach – the SETAF–ADF one (Translation 31). We have used the research on semantics signatures [37] to show that a full and exact SETAF–AF transformation is impossible (see Section 6.1.3). The same appears to be true in the SETAF–AFN case (Sections 6.3.1 and 10.1.1), though in this case we have limited ourselves to the admissible and preferred semantics only. Our inability to create an exact translation to EASs is again due to the addition of evidence (Section 5.6.1).

Fortunately, our results for the AFRA translations are somewhat more encouraging; we have a full, generic and exact translation to every other framework we have considered, i.e. AFs, BAFs and AFNs. We did not focus on ADFs in this analysis, however, by simply chaining the AFRA–AF and AF–ADF methods we can easily create a suitable exact translation.

We now come to EAFs and their generalization EAFCs. Due to their unique semantics (see Section 2.1.4 and in particular, 2.1.4.2), we were able to show that a full and exact translation is not possible from these frameworks to any other structure with the exception of ADFs (see Sections 8.2.2, 8.3.2, 8.4.1 and 8.5.2). Although the most general EAFC–ADF translation we have presented was faithful, it can be easily improved by considering the self–attacker consistency method from Section 4.4.2, not the pure bypass approach, as a basis for our transformation. However, as explained in Section 8.6.4, the approach for EAFs still requires further analysis due to the mismatch between the conflict–free semantics between our frameworks (see also Section 2.1.4.2).

The existence of exact translations from BAFs to other frameworks is, in a certain sense, a much more complicated question than in any other framework. This is due to the fact that in order to establish a given result we not only need to consider a given type of semantics, but also its parametrization. We have indeed obtained a full, generic and exact translation for the d–family of BAF semantics with the identical parametrization to AFs. Due to the fact that AFs can be exactly transformed into almost every other structure, this result propagates further. Nevertheless, in this particular case further research is required, and various semantics for BAFs still need to be properly defined (see Section 2.2.1).

We now come to AFNs. Owing to the research on semantics signatures, we have indeed managed to establish that improving the current AFN–AF method is possible (see Section 10.1.1) and that an exact translation for admissible and preferred semantics can still be created. We hope to find this method in the future. This result can propagate to other frameworks, though please note that the exact and full results for SETAFs and ADFs are already available. We are not yet sure whether an exact and full translation is possible for BAFs, but hopefully the future research will answer this question. Finally, an exact approach for EASs is not possible due to the evidence argument, similarly as it was with other methods that used them as target frameworks.

The methods we have used in the translations from EASs to other structures were similar as for the AFNs. Nevertheless, the presence of group attack in the framework made it impossible to create an exact translation to AFs (see Section 11.1.1). Moreover, as a result of the analysis performed in Sections 6.1.3 and 10.1.1, this is also true in the case of AFNs as far as conflict–free, admissible and preferred semantics are concerned. Fortunately, we have managed to establish generic, full and exact methods for translating EASs into SETAFs and ADFs.

Finally, we arrive at ADFs, which based on our results emerged as one of the most general tools for abstract argumentation, capable of handling even the extended argumentation framework. However, when we tried to translate from ADFs to other structures, we have obtained primarily strong translations. The single exact result we have managed to obtain concerned translating ADFs to SETAFs under the aa–semantics. Although it is technically speaking a source–subclass one, it can be easily extend to general ADFs by the use of weakly valid normal form translation. The analysis in Section 12.1.6 showed that in the case of AFs, an exact translation is not possible, even if we limit ourselves to the simpler types of ADFs such as BADFs or AADF$^+$s. A number of these results also propagated to other frameworks, particularly concerning the ca–families of ADF semantics. The only possible exception here is the extended argumentation framework and we would like to pursue this line of research in the future. Given the results for aa–semantics, we also hope that exact translations to SETAFs might be possible for other approaches. However, the severe modifications the argument domains need to undergo in the majority of the presented methods make it difficult to say with certainty how far we can go.

Along with a number of new results, we have also identified various tasks we would like to tackle in the future. First of all, we would like to refine and extend our translation classification system. In particular, we would like to consider the analysis performed in [42] and be able to categorize the ADF–AFN translation, which failed to conform to any of the patterns we have identified so far. Moreover, out of all of the groups of properties we have introduced, the computational attributes have received the least attention. No complexity analysis has been performed for our approaches, though it is in a great deal due to the fact that the complexity results for the semantics of various frameworks we have considered are not fully researched. We would also like to develop more advanced types of modularity for ADFs which would allow us to improve the quality of our translations. We believe that these results could help in developing efficient algorithms for the transformations we have presented in this work and in building software for abstract argumentation.

Another important task concerns the signatures and realizability of argumentation semantics. Whenever we could, we have tried to use the results for the Dung's framework in order to establish whether a given exact translation is possible or not, as we have explained in the previous paragraphs. Nevertheless, there are still more questions than answers concerning this particular topic. We believe it would also be beneficial to consider the introduction of slightly more relaxed, faithful signatures, which so far have been only considered in [43].

We would also like to continue our work on the ADF related translations. The ADF–AF approaches are primarily coalition–based and it would be interesting to consider other patterns, in particular the attack propagation and pure defender methods. We have also not found a satisfactory translation for the ca–family of ADF semantics; we believe that an ADF–EAF translation could address this particular issue. Moreover, we would like to extend our analysis with the results presented in [44] and strengthen the labeling–based ADF–AF transformation (see Section 12.1.5).

The two final tasks we would be interested in concern the research on normal forms and BAF semantics. We have already remarked upon possible other minimal normal forms (Section 4.1) and on the fact that we do not yet have any translations for strongly valid forms (Section 4.3.3). Concerning BAFs, we are interested in the development of their "missing" grounded and complete semantics (see Section 2.2.1). It would be also valuable to identify the subclasses of BAFs on which the semantics classification collapses, similarly as we have managed for ADFs with the introduction of AADF$^+$s. Although this particular task might seem inconspicuous at first, we believe that it might shed more light on the research on the bipolar argumentation altogether.

Acknowledgments

The author is a member of the Vienna PhD School of Informatics. This work has been funded by the project “Abstract Dialectical Frameworks: Advanced Tools for Formal Argumentation” under grant FWF I1102 and “GRAPPA: A Semantical Framework for Graph-Based Argument Processing” under grant FWF I2843. The author is currently funded by EPSRC Project EP/N008294/1 “Framework for Computational Persuasion”. We would like to thank Stefan Woltran for his immense help and great patience with this work. We also thank Piotr Jaroszek for proofreading this document, Thomas Linsbichler and Nir Oren for their assistance in developing some of the results, and Martin Riener for his constant support. Finally, we express our gratitude to numerous researchers whose work both inspired the author and made this study possible.

References

[1] Leila Amgoud and Jonathan Ben-Naim. Ranking-based semantics for argumentation frameworks. In Weiru Liu, V.S. Subrahmanian, and Jef Wijsen, editors, *Scalable Uncertainty Management*, volume 8078 of *Lecture Notes in Computer Science*, pages 134–147. Springer-Verlag Berlin Heidelberg, 2013.

[2] Leila Amgoud and Claudette Cayrol. On the acceptability of arguments in preference-based argumentation. In Gregory Cooper and Serafin Moral, editors, *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI 1998)*, pages 1–7. Morgan Kaufmann Publishers Inc., 1998.

[3] Leila Amgoud and Claudette Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning*, 29(2):125–169, 2002.

[4] Leila Amgoud, Yannis Dimopoulos, and Pavlos Moraitis. Making decisions through preference-based argumentation. In Gerhard Brewka and Jérôme Lang, editors, *Proceedings of the 11th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2008)*, pages 113–123. AAAI Press, 2008.

[5] Leila Amgoud and Srdjan Vesic. A new approach for preference-based argumentation frameworks. *Annals of Mathematics and Artificial Intelligence*, 63:149–183, 2011.

[6] Katie Atkinson, Trevor J. M. Bench-Capon, and Peter McBurney. PARMENIDES: Facilitating deliberation in democracies. *Artificial Intelligence and Law*, 14(4):261–275, 2006.

[7] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[8] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. Encompassing attacks to attacks in abstract argumentation frameworks. In Claudio Sossai and Gaetano Chemello, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 5590 of *Lecture Notes in Computer Science*, pages 83–94. Springer-Verlag Berlin Heidelberg, 2009.

[9] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. AFRA: Argumentation framework with recursive attacks. *International Journal of Approximate Reasoning*, 52(1):19–37, 2011.

[10] Pietro Baroni and Massimiliano Giacomin. Characterizing defeat graphs where argumentation semantics agree. In G. Simari and P. Torroni, editors, *Proceedings of 1st International Workshop on Argumentation and Non-Monotonic Reasoning (ARGNMR 2007)*, pages 33–48, 2007.

[11] Pietro Baroni and Massimiliano Giacomin. A systematic classification of argumentation frameworks where semantics agree. In Philippe Besnard, Sylvie Doutre, and Anthony Hunter, editors, *Proceedings of the 2nd Conference on Computational Models of Argument (COMMA 2008)*, volume 172 of *Frontiers in Artificial Intelligence and Applications*, pages 37–48. IOS Press, 2008.

[12] Pietro Baroni and Massimiliano Giacomin. Semantics of abstract argument systems. In Iyad Rahwan and Guillermo R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 25–44. Springer, 2009.

[13] Ringo Baumann, Wolfgang Dvorák, Thomas Linsbichler, Hannes Strass, and Stefan Woltran. Compact argumentation frameworks. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 69–74. IOS Press, 2014.

[14] Trevor J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[15] Trevor J. M. Bench-Capon and Sanjay Modgil. Case law in extended argumentation frameworks. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law (ICAIL 2009)*, pages 118–127, New York, NY, USA, 2009. ACM.

[16] Trevor J. M. Bench-Capon, Henry Prakken, and Giovanni Sartor. Argumentation in legal reasoning. In Iyad Rahwan and Guillermo R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 363–382. Springer, 2009.

[17] Philippe Besnard and Anthony Hunter. *Elements of Argumentation*. MIT Press, 2008.

[18] Guido Boella, Dov Gabbay, Leendert van der Torre, and Serena Villata. Meta-argumentation modelling I: Methodology and techniques. *Studia Logica*, 93(2-3):297–355, 2009.

[19] Guido Boella, Dov Gabbay, Leendert van der Torre, and Serena Villata. Support in abstract argumentation. In Massimiliano Giacomin Pietro Baroni, Federico Cerutti and Guillermo R. Simari, editors, *Proceedings of the 3rd Conference on Computational Models of Argument (COMMA 2010)*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 111–122, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[20] Gerhard Brewka, Paul E. Dunne, and Stefan Woltran. Relating the semantics of abstract dialectical frameworks and standard afs. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 780–785. AAAI Press, 2011.

[21] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes Peter Wallner, and Stefan Woltran. Abstract dialectical frameworks revisited. In Francesca Rossi, editor, *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, pages 803–809. AAAI Press, 2013.

[22] Gerhard Brewka, Sylwia Polberg, and Stefan Woltran. Generalizations of Dung frameworks and their role in formal argumentation. *IEEE Intelligent Systems*, 29(1):30–38, January 2014.

[23] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczyński, editors, *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010)*, pages 102–111. AAAI Press, 2010.

[24] Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5-6):286–310, 2007.

[25] Martin Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2):109–145, 2009.

[26] Martin Caminada and Yining Wu. On the limitations of abstract argumentation. In Patrick De Causmaecker, Joris Maervoet, Tommy Messelis, Katja Verbeeck, and Tim Vermeulen, editors, *Proceedings of the 23rd Benelux Conference on Artificial Intelligence (BNAIC 2011)*, pages 59–66, 2011.

[27] Martin W. A. Caminada, Walter A. Carnielli, and Paul E. Dunne. Semi-stable semantics. *Journal of Logic and Computation*, 22(5):1207–1254, October 2012.

[28] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Bipolar abstract argumentation systems. In Iyad Rahwan and Guillermo R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 65–84. Springer, 2009.

[29] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Coalitions of arguments: A tool for handling bipolar argumentation frameworks. *International Journal of Intelligent Systems*, 25(1):83–109, 2010.

[30] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Bipolarity in argumentation graphs: Towards a better understanding. *International Journal of Approximate Reasoning*, 54(7):876–899, 2013.

[31] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. An axiomatic approach to support in argumentation. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Proceedings of the 3rd Third International Workshop on Theory and Applications of Formal Argumentation (TAFA 2015)*, volume 9524 of *Lecture Notes in Computer Science*, pages 74–91. Springer International Publishing, 2015.

[32] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Inference from controversial arguments. In Geoff Sutcliffe and Andrei Voronkov, editors, *Proceedings of the 12th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2005)*, volume 3835 of *Lecture Notes in Computer Science*, pages 606–620. Springer-Verlag Berlin Heidelberg, 2005.

[33] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Prudent semantics for argumentation frameworks. In Andrew Lim, editor, *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2005)*, pages 568–572, Washington, DC, USA, 2005. IEEE Computer Society.

[34] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Symmetric argumentation frameworks. In Lluis Godo, editor, *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, volume 3571 of *Lecture Notes in Computer Science*, pages 317–328. Springer Berlin Heidelberg, 2005.

[35] Cosmina Croitoru and Timo Kötzing. A normal form for argumentation frameworks. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Theory and Applications of Formal Argumentation*, volume 8306 of *Lecture Notes in Computer Science*, pages 32–45. Springer-Verlag Berlin Heidelberg, 2014.

[36] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[37] Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. *Artificial Intelligence*, 228:153 – 178, 2015.

[38] Paul E. Dunne, Thomas Linsbichler, Christof Spanring, and Stefan Woltran. Investigating the relationship between argumentation semantics via signatures. In Subbarao Kambhampati, editor, *Proceedings of the 25th International Joint Conference on Artificial Intelligence, (IJCAI 2016)*, pages 1051–1057. AAAI Press, 2016.

[39] Paul E. Dunne, Sanjay Modgil, and Trevor J. M. Bench-Capon. Computation in Extended Argumentation Frameworks. In Rudi Studer Helder Coelho and Michael Wooldridge, editors, *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 119–124. IOS Press, 2010.

[40] Wolfgang Dvorák and Christof Spanring. Comparing the expressiveness of argumentation semantics. In Stefan Szeider Bart Verheij and Stefan Woltran, editors, *Proceedings of the 4th Conference on Computational Models of Argument (COMMA*

2012), volume 245 of *Frontiers in Artificial Intelligence and Applications*, pages 261–272. IOS Press, 2012.

[41] Wolfgang Dvořák, Sarah Alice Gaggl, Thomas Linsbichler, and Johannes Peter Wallner. Reduction-based approaches to implement modgils extended argumentation frameworks. In Thomas Eiter, Hannes Strass, Mirosław Truszczyński, and Stefan Woltran, editors, *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation*, volume 9060 of *Lecture Notes in Artificial Intelligence*, pages 249–264. Springer International Publishing, 2015.

[42] Wolfgang Dvořák and Stefan Woltran. On the intertranslatability of argumentation semantics. *Journal of Artificial Intelligence Research*, 41(2):445–475, 2011.

[43] Sjur Kristoffer Dyrkolbotn. How to argue for anything: Enforcing arbitrary sets of labellings using afs. In Giuseppe De Giacomo Chitta Baral and Thomas Eiter, editors, *Proceedings of the 14th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 626–629. AAAI Press, 2014.

[44] Stefan Ellmauthaler. Abstract dialectical frameworks: properties, complexity, and implementation. Master's thesis, Faculty of Informatics, Institute of Information Systems, Vienna University of Technology, 2012.

[45] Stefan Ellmauthaler and Hannes Strass. The DIAMOND system for computing with abstract dialectical frameworks. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 233–240. IOS Press, 2014.

[46] John Fox, David Glasspool, Vivek Patkar, Mark Austin, Liz Black, Matthew South, Dave Robertson, and Charles Vincent. Delivering clinical decision support services: There is nothing as practical as a good theory. *Journal of Biomedical Informatics*, 43(5):831–843, 2010.

[47] Dov M. Gabbay. Semantics for higher level attacks in extended argumentation frames part 1: Overview. *Studia Logica*, 93(2-3):357–381, 2009.

[48] Sarah Alice Gaggl, Sebastian Rudolph, and Hannes Strass. On the computational complexity of naive-based semantics for abstract dialectical frameworks. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2985–2991. AAAI Press, July 2015.

[49] Sarah Alice Gaggl and Hannes Strass. Decomposing abstract dialectical frameworks. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings*

of the 5th Conference on Computational Models of Argument (COMMA 2014), volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 281–292. IOS Press, 2014.

[50] Massimiliano Giacomin, Thomas Linsbichler, and Stefan Woltran. On the functional completeness of argumentation semantics. In Chitta Baral, James P. Delgrande, and Frank Wolter, editors, *Proceedings of the 15th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2016)*, pages 43–52. AAAI Press, 2016.

[51] Nikos Gorogiannis and Anthony Hunter. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. *Artificial Intelligence*, 175(9-10):1479–1497, 2011.

[52] Georg Gottlob. Translating default logic into standard autoepistemic logic. *Journal of the ACM*, 42(4):711–740, July 1995.

[53] Anthony Hunter. A probabilistic approach to modelling uncertain logical arguments. *International Journal of Approximate Reasoning*, 54(1):47 – 81, 2013.

[54] Anthony Hunter and Matthew Williams. Aggregating evidence about the positive and negative effects of treatments. *Artificial Intelligence in Medicine*, 56(3):173–190, 2012.

[55] Tomasz Imielinski. Results on translating defaults to circumscription. *Artificial Intelligence*, 32(1):131–146, 1987.

[56] Hadassa Jakobovits and Dirk Vermeir. Robust semantics for argumentation frameworks. *Journal of Logic and Computation*, 9(2):215–261, 1999.

[57] Tomi Janhunen. On the intertranslatability of non–monotonic logics. *Annals of Mathematics and Artificial Intelligence*, 27(1-4):79–128, 1999.

[58] Paolo Liberatore. Bijective faithful translations among default logics. *Journal of Logic and Computation*, 24(4):763–807, 2014.

[59] Thomas Linsbichler, Jörg Pührer, and Hannes Strass. A uniform account of realizability in abstract argumentation. In Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen, editors, *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016), Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 252–260. IOS Press, 2016.

[60] Peter McBurney and Simon Parsons. Dialogue games for agent argumentation. In Iyad Rahwan and Guillermo R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 261–280. Springer, 2009.

[61] Sanjay Modgil. An argumentation based semantics for agent reasoning. In Joao Leite Mehdi Dastani, Amal El Fallah Seghrouchni and Paolo Torroni, editors, *Proceedings of the 1st International Workshop on Languages, Methodologies and Development Tools for Multi-Agent Systems (LADS 2007)*, Lecture Notes in Artificial Intelligence, pages 37–53. Springer-Verlag Berlin Heidelberg, 2007.

[62] Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, 173(9-10):901–934, 2009.

[63] Sanjay Modgil. Revisiting abstract argumentation frameworks. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Theory and Applications of Formal Argumentation*, volume 8306 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag Berlin Heidelberg, 2014.

[64] Sanjay Modgil and Trevor J. M. Bench-Capon. Metalevel argumentation. *Journal of Logic and Computation*, 21(6):959–1003, 2011.

[65] Sanjay Modgil and Michael Luck. Argumentation based resolution of conflicts between desires and normative goals. In Iyad Rahwan and Pavlos Moraitis, editors, *Proceedings of the 5th International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2008)*, volume 4946 of *Lecture Notes in Artificial Intelligence*, pages 19–36. Springer-Verlag Berlin Heidelberg, 2009.

[66] Sanjay Modgil and Henry Prakken. Reasoning about preferences in structured extended argumentation frameworks. In Massimiliano Giacomin Pietro Baroni, Federico Cerutti and Guillermo R. Simari, editors, *Proceedings of the 3rd Conference on Computational Models of Argument (COMMA 2010)*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 347–358. IOS Press, 2010.

[67] Sanjay Modgil and Henry Prakken. A general account of argumentation with preferences. *Artificial Intelligence*, 195:361–397, 2013.

[68] Søren Nielsen and Simon Parsons. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In Nicolas Maudet, Simon Parsons, and Iyad Rahwan, editors, *Proceedings of the 3rd International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006)*, volume 4766 of *Lecture Notes in Computer Science*, pages 54–73. Springer Berlin Heidelberg, 2007.

[69] Farid Nouioua. AFs with necessities: Further semantics and labelling characterization. In Weiru Liu, V.S. Subrahmanian, and Jef Wijsen, editors, *Proceedings of the 7th International Conference on Scalable Uncertainty Management (SUM 2013)*, volume 8078 of *Lecture Notes in Computer Science*, pages 120–133. Springer-Verlag Berlin Heidelberg, 2013.

[70] Farid Nouioua and Vincent Risch. Argumentation frameworks with necessities. In Salem Benferhat and John Grant, editors, *Proceedings of the 5th International Conference on Scalable Uncertainty Management (SUM 2011)*, volume 6929 of *Lecture Notes in Artificial Intelligence*, pages 163–176. Springer-Verlag Berlin Heidelberg, 2011.

[71] Nir Oren. *An Argumentation Framework Supporting Evidential Reasoning with Applications to Contract Monitoring*. Phd thesis, University of Aberdeen, Aberdeen, Scotland, 2007.

[72] Nir Oren and Timothy J. Norman. Semantics for evidence-based argumentation. In Philippe Besnard, Sylvie Doutre, and Anthony Hunter, editors, *Proceedings of the 2nd Conference on Computational Models of Argument (COMMA 2008)*, volume 172 of *Frontiers in Artificial Intelligence and Applications*, pages 276–284. IOS Press, 2008.

[73] Nir Oren, Chris Reed, and Michael Luck. Moving between argumentation frameworks. In Massimiliano Giacomin Pietro Baroni, Federico Cerutti and Guillermo R. Simari, editors, *Proceedings of the 3rd Conference on Computational Models of Argument (COMMA 2010)*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 379–390, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[74] Sylwia Polberg. Extension–based semantics of abstract dialectical frameworks. Technical Report DBAI-TR-2014-85, Institute for Information Systems, Technical University of Vienna, 2014.

[75] Sylwia Polberg. Revisiting extension–based semantics of abstract dialectical frameworks. Technical Report DBAI-TR-2015-88, Institute for Information Systems, Technical University of Vienna, 2015.

[76] Sylwia Polberg. Understanding the Abstract Dialectical Framework. In Loizos Michael and Antonis Kakas, editors, *Proceedings of the 15th European Conference on Logics in Artificial Intelligence (JELIA 2016)*, volume 10021 of *Lecture Notes in Artificial Intelligence*, pages 430–446. Springer International Publishing, 2016.

[77] Sylwia Polberg and Nir Oren. Revisiting support in abstract argumentation frameworks. Technical Report DBAI-TR-2014-87, Institute for Information Systems, Technical University of Vienna, 2014.

[78] Sylwia Polberg and Nir Oren. Revisiting support in abstract argumentation systems. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 369–376. IOS Press, 2014.

[79] Sylwia Polberg, Johannes Peter Wallner, and Stefan Woltran. Admissibility in the abstract dialectical framework. In Paolo Torroni Leon van der Torre Joao Leite, Tran Cao Son and Stefan Woltran, editors, *Proceedings of the 14th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA 2013)*, volume 8143 of *Lecture Notes in Artificial Intelligence*, pages 102–118. Springer-Verlag Berlin Heidelberg, 2013.

[80] Henry Prakken. Models of persuasion dialogue. In Iyad Rahwan and Guillermo R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 281–301. Springer, 2009.

[81] Henry Prakken. On support relations in abstract argumentation as abstractions of inferential relations. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 735–740. IOS Press, 2014.

[82] Jörg Pührer. Realizability of three-valued semantics for abstract dialectical frameworks. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 3171–3177. AAAI Press, 2015.

[83] Iyad Rahwan and Guillermo R. Simari. *Argumentation in Artificial Intelligence*. Springer, 1st edition, 2009.

[84] Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence*, 205:39 – 70, 2013.

[85] Hannes Strass. Instantiating knowledge bases in abstract dialectical frameworks. In Paolo Torroni Leon van der Torre Joao Leite, Tran Cao Son and Stefan Woltran, editors, *Proceedings of the 14th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA 2013)*, volume 8143 of *Lecture Notes in Artificial Intelligence*, pages 86–101. Springer-Verlag Berlin Heidelberg, 2013.

[86] Hannes Strass. The relative expressiveness of abstract argumentation and logic programming. In Sven Koenig and Blai Bonet, editors, *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, pages 1625–1631, Austin, TX, USA, January 2015. AAAI Press.

[87] Hannes Strass and Johannes Peter Wallner. Analyzing the Computational Complexity of Abstract Dialectical Frameworks via Approximation Fixpoint Theory. In Giuseppe De Giacomo Chitta Baral and Thomas Eiter, editors, *Proceedings of the 14th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 101–110, Vienna, Austria, July 2014. AAAI Press.

[88] Serena Villata, Guido Boella, and Leendert Van Der Torre. Attack semantics for abstract argumentation. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 406–413. AAAI Press, 2011.

15 Proof Appendix

15.1 Background: Proof Appendix

Theorem 2.46. *Let $EF = (A, R, D)$ be a finite EAF and $E \subseteq A$ a conflict–free extension of EF. If an argument $a \in E$ defeats$_E$ an argument $b \in A$, then there is no reinstatement set for this defeat$_E$ on E, iff there exists a sequence $((z_1, (x_1, y_1)), ..., (z_n, (x_n, y_n)))$ of distinct defense attacks from D s.t.*

- *there is an argument $g \in A$ s.t. $x_n = a$, $y_n = b$ and $z_n = g$,*
- *no two pairs (x_i, y_i) and (x_j, y_j) are the same for $i \neq j$,*
- *for every $(z_i, (x_i, y_i))$ where $1 < i \leq n$, either no argument h in E defeats$_E$ z_i or for every such defeat, there exists an argument $l \in A$ s.t. $(l, (h, z_i)) \in \{(z_1, (x_1, y_1)), ..., (z_{i-1}, (x_{i-1}, y_{i-1}))\}$, and*
- *no argument in E defeats$_E$ z_1.*

Proof. Let $(x, y) \in R$. By $datt(x, y)$ we denote the set of arguments that carry out defense attacks on (x, y), i.e. $datt(x, y) = \{c \mid (c, (x, y)) \in D\}$.

Let us first show that if there is no reinstatement set for the (a, b) defeat$_E$ on E, then a suitable sequence $((z_1, (x_1, y_1)), ..., (z_n, (x_n, y_n)))$ exists. Due to the fact that no reinstatement set exists, then $\{(a, b)\}$ is not a reinstatement set for the the defeat$_E$ of a on b. Hence, $datt(a, b)$ is not empty and there exists at least one element in it different from b – otherwise, $\{(a, b)\}$ would have been a reinstatement set. Moreover, there is at least one element in $datt(a, b)$ that is not defeated$_E$ by E or none of the defeats$_E$ have a reinstatement set – otherwise, we could have joined these sets and added (a, b) to obtain a reinstatement set for the a–b defeat$_E$. Again, it has to be different from b, as we know that a defeats$_E$ b and that $\{(a, b)\}$ is not a reinstatement set. Let us denote elements meeting these requirements with $d^1_1, ..., d^1_k$. If it is the case that any of d^1_j is not defeated$_E$ by E, then $(d^1_j, (a, b))$ is a valid sequence and we are done.

Let us therefore assume that for every d^1_j we can find an argument $e \in E$ s.t. e defeats$_E$ d^1_j. Again, none of such defeats can have a reinstatement set on E – otherwise, we would have been able to construct a reinstatement set for (a, b). For the same reasons as above, this means that $datt(e, d^1_j)$ is not empty. Moreover, $datt(e, d^1_j)$ has to contain an argument that is different from d^1_j and either is not defeated$_E$ by E or no such defeat has a reinstatement set. However, we can also observe that if $e = a$, then we can choose such a d^1_j and $d^2_{mj} \in datt(e, d^1_j)$ for $1 < m < |datt(e, d^1_j)|$ s.t. that d^2_{mj} meets our requirements and is different from b. If it were not possible, then $\{(a, b), (a, d^1_1), ..., (a, d^1_k)\}$ would have been a reinstatement set for $\{(a, b)\}$. Thus, we can filter our first and second level d's and continue our analysis. If it is the case that any of d^2_{mj} is not defeated$_E$ by E, then again $(d^2_{mj}, (e, d^1_j))$ is a satisfactory sequence for the e–d^1_j defeat. By appending such sequences for the remaining defeats on d_j^1 and including the $(d_j^1, (a, b))$ defeat, we can receive the desired sequence for $(d_j^1, (a, b))$.

We can therefore focus again on the case that for no defeat$_E$ by any argument $f \in E$ on any $d_m j^2$ there is a reinstatement set on E. We can continue the analysis in the similar manner, each time showing that a sequence with unique conflicts can be built and that for each defense attacks in the sequence is "protected" by the attacks lower in the sequence. Since the amount of conflicts in our framework is finite, we are bound to reach defense attacks by arguments that are not defeated$_E$ by E. This concludes this part of the proof.

Let now $((z_1, (x_1, y_1)), ..., (z_n, (x_n, y_n)))$ be a defense attack sequence satisfying our requirements. There is no argument $d \in E$ s.t. d defeats$_E$ z_1. Therefore, there cannot be a reinstatement set for (x_1, y_1). If there exists an argument in E defeating$_E$ z_2, then by the construction of the sequence it holds that this conflict is defense attacked by z_1. Consequently, there cannot be a reinstatement set for this conflict on E. We can repeat this procedure till we reach z_n. As there is no defeat$_E$ on z_n that can be reinstated, there is no reinstatement set for (x_n, y_n). This concludes the proof. $\square$

Theorem 2.54. *Let $EF = (A, R, D)$ be a finitary EAF. The following holds:*

- *every preferred extension is complete, but not vice versa.*
- *every stable extension is complete, but not vice versa.*
- *the grounded extension is a minimal complete extension, but not necessarily the least one.*

Proof. Let $E \subseteq A$ be a preferred extension of EF. Assume it is not complete; as E is admissible, this means that there is an argument $a \in A \setminus E$ that is defended by E. Let us consider the extension $E' = E \cup \{a\}$. Due to defense, it cannot be the case that a defeats$_E$ any argument in E and vice versa. Furthermore, a cannot be defeating itself w.r.t. E either. This means that either there are no relevant conflicts in R to start with, or they are already defense attacked by elements in E. In both cases this leads to the conclusion that E' is conflict–free. We now need to show it is admissible. Let us consider an arbitrary defeat$_E$ by $b \in E$ on $c \in A$ that has a reinstatement set $\{(x_1, y_1), ..., (x_n, y_n)\}$ on E. As E does not defeat$_E$ a, it cannot be the case that there is a pair (x_i, y_i) in the reinstatement set s.t. $(a, (x_i, y_i)) \in D$. Therefore, if E defeats an argument $c \in A$ with reinstatement on E, then so does E'. We can also observe that if an argument $c \in A$ did not defeat$_E$ any argument in E, then it does not defeat$_{E'}$ any argument in E' either. This brings us to the result that E' has to be admissible. This however means that E could not have been a maximal admissible extension – we can observe that $E \subset E'$ – and thus we contradict the assumption it is preferred. Hence, we can conclude that if E is preferred, then it is complete. The fact that not every complete extension is preferred can be observed in Example 6.

Let $E \subseteq A$ be a stable extension of EF. We can observe it is also admissible in EF. Every argument outside of E is defeated$_E$ by E and the collection of all defeats$_E$

carried out by elements of E is a simple reinstatement set for any of them. Therefore, every argument $a \in E$ is defended by E, and due to conflict–freeness of E it cannot be the case that at the same time, E defeats$_E$ and defends an argument $b \notin E$. Therefore, E is complete in EF. The fact that not every complete extension is stable can be observed in Example 6.

In order to show that the grounded extension is a minimal complete one, we will use the operator iteration approach. Assume E is the grounded extension and there exists a smaller complete extension $E' \subset E$. Let $G = \emptyset$. We can observe that only those arguments that are not attacked in R at all can be acceptable w.r.t. $\emptyset$ – there is no argument in G that would prevent an attack turning into a defeat. Therefore, if an argument $b \in A$ is acceptable w.r.t. $\emptyset$, then it is acceptable w.r.t. any other set of arguments. Thus, we can add the arguments produced by $\mathcal{F}_{EF}(\emptyset)$ to G and observe that $G \subseteq E' \subset E$ due to the completeness of E'.

Let us now apply the operator again and let $a \in A$ be an argument acceptable w.r.t. G. Assume it is not acceptable w.r.t. E'. This means there is an argument $b \in A$ that defeats$_{E'}$ a and is not in turn defeated$_{E'}$ by any argument $c \in E'$ with a reinstatement set. We can observe that if b defeats$_{E'}$ a, then due to the fact that $G \subseteq E'$, b defeats$_G$ a as well. Therefore, G has to defeat$_G$ b with a reinstatement set on G, even though it is not the case for E'. Let $c \in G$ be an argument carrying out the reinstated defeat on b in G and let $\{(x_1, y_1), ..., (x_n, y_n)\}$ be the relevant reinstatement set. We will show that $G' = \mathcal{F}_{EF}(G)$ also defeats$_{G'}$ b with the same reinstatement. We can observe that every argument defense attacking any of the defeats listed in the reinstatement set is defeated$_G$ by G. Therefore, it cannot be acceptable w.r.t. G and will not appear in G'. This means that any pair in the reinstatement set that was a defeat$_G$ is also a defeat$_{G'}$. We can therefore show that if G defeats$_G$ b with a reinstatement, then so does the grounded extension of EF (which in this case, is E). Now, if c does not defeat$_{E'}$ b, then there is an argument $d \in E'$ s.t. $(d, (c, b)) \in D$. Consequently, d has to be defeated$_G$ by G with a reinstatement, which based on the previous explanations means that d cannot be in the grounded extension. Therefore, E' cannot be a subset of E and we reach a contradiction. This brings us to the conclusion that a has to be acceptable w.r.t. E' and by completeness of E', it holds that $G \subseteq E' \subset E$ where G is extended by the arguments in $\mathcal{F}_{EF}(G)$.

We can continue this line of reasoning till our grounded extension is computed and conclude that $G \subseteq E' \subset E = G$. We thus reach a contradiction with the assumption that $E' \subset E$ and can therefore conclude that E has to be a minimal complete extension of EF. The fact it is not necessarily the least can be observed in Example 8. $\square$

Theorem 2.59. *Let $bh - EF = (A, R, D)$ be bounded hierarchical EAF. Every stable extension of $bh - EF$ of is preferred, but not vice versa.*

Proof. Assume $E \subseteq A$ is a stable extension of EF, but is not preferred. As it is complete by Theorem 2.54, this means there exists an admissible extension $E' \subseteq A$ s.t. $E \subset E'$. We can observe that the arguments in the set $F = E' \setminus E$ are defeated$_E$ by E. Consequently, for every defeat$_E$ by an argument $c \in E$ on an argument $d \in F$ there must exist a corresponding defense attack by an argument in $e \in F$. Therefore, $(c, d) \in R$ and $(e, (c, d)) \in D$, which means that c and d belong to the same partition of $bh - EF$, while e is a level higher. As e is also defeated$_E$ by E (say, by $f \in E$), then again there must be a relevant defense attacking argument $g \in F$. Since we are dealing with a bounded hierarchical framework, g is again a level higher. We can continue in this manner until we reach a defeat$_E$ in E without a relevant defense attacking argument in F, and as we are dealing with a bounded hierarchical framework, we will reach this point sooner or later. Therefore, the conflict behind the defeat$_E$ cannot be defense attacked by E' and will become a defeat w.r.t. E'. We breach the conflict–freeness of E'. Thus, if E is a stable extension of EF, then it is also preferred.

To show the other way around, it suffices to consider an AF–style EAF $(\{a, b, c\}, \{(a, b), (b, a), (b, c), (c, c)\}, \emptyset)$ with a symmetric attack between a and b and a self–attacker c. Both $\{a\}$ and $\{b\}$ are preferred extensions, however, only the latter also attacks c. Consequently, only $\{b\}$ is stable. □

Lemma 2.60. *Let $EF = (A, R, D)$ be a bounded hierarchical EAF or an EAF s.t. there are no arguments $a, b \in A$ for which $(a, b), (b, a) \in R$. A set $E \subseteq A$ is a conflict–free extension of EF iff there are no defeats$_E$ in E.*

Proof. Let us first consider the bounded hierarchical case. Assume that even though there are no defeats$_E$ in E, the set is not conflict–free. Therefore, this means that there are $a, b \in E$ s.t. $(a, b), (b, a) \in R$. However, since these conflicts do not become defeats, then it means that there are $d_1, d_2 \in E$ s.t. $(d_1, (a, b)), (d_2, (b, a)) \in D$. First of all, we can observe that d_1 and d_2 have to be a level higher than a and b in the partition of EF. Moreover, due to the restrictions in the definition of EAFs, $(d_1, d_2), (d_2, d_1) \in R$. As these conflicts do not become defeats in E, then again there are some arguments d_3 and d_4 a level higher in the partition that defense attack these attacks. Furthermore, just like in the d_1 and d_2 case, there are symmetric attacks between d_3 and d_4. They also need to be defense attacked and we can continue this analysis until we reach symmetric attacks in which at least one conflict cannot be defense attacked further, which is warrantied by the bounded hierarchical nature of EF. Therefore, this conflict becomes a defeat, and we reach a contradiction with the assumptions on E. Hence, if there are no defeats$_E$ in E, then E is a conflict–free extension of EF. The fact that if a set is conflict–free, then it contains no defeats, comes from Proposition 2.43.

Let us now consider the frameworks without symmetric attacks. This means that for every $(a, b) \in R$, $(b, a) \notin R$. Thus, the definition of conflict–freeness is reduced to requiring that for every $a, b \in E$, if $(a, b) \in R$, $\exists c \in E$ s.t. $(c, (a, b)) \in D$. Therefore, E is conflict–free if there are no $a, b \in E$ s.t. a defeats$_E$ b. Along with Proposition 2.43, this gives us the final result. □

Lemma 2.72. ***BAF Fundamental Lemma*** *Let $BF = (A, R, S)$ be a BAF, $R' \subseteq R^{ind}$, $R'' \subseteq R^{ind}$ two collections of indirect attacks in BF, $E \subseteq A$ a d–admissible extension w.r.t. (R', R'') and $a, b \in A$ arguments defended by E w.r.t. R'' in BF. If $R' = R''$, then $E' = E \cup \{a\}$ is d–admissible w.r.t. (R', R'') and b is defended by E w.r.t. R'' in BF.*

Proof. Assume $E' = E \cup \{a\}$ is not +conflict–free w.r.t. R'. As E itself is, then it has to be the case that there is an argument $c \in E$ s.t. $cR'a$ or $aR'c$, or $aR'a$. If $cR'a$, then due to the fact that $R' \subseteq R''$, a has to be defended from c by E. Thus, there is an argument $d \in E$ s.t. $dR''c$. Since $R'' \subseteq R'$, we breach the +conflict–freeness of E. If $aR'c$, then due to the fact that $R' \subseteq R''$ and E is d–admissible, there is an argument $d \in E$ s.t. $dR''a$. Thus, as E defends a, there is an argument $e \in E$ s.t. $eR''d$. Since $R'' \subseteq R'$, we breach the +conflict–freeness of E. Similar analysis can be performed for the $aR'a$ case. We can therefore conclude that E' is +conflict–free.

Let us now assume that E' is not d–admissible. As it is +conflict–free, it has to be the case there is an argument $c \in E'$ that is not defended by E' w.r.t. R''. Therefore, there exists an argument $d \in A$ s.t. $dR''c$ and no argument $e \in E'$ s.t. $eR''d$. If there is no such defending argument in E', then there is none in E either. Thus, if $c \in E$, we breach the d–admissibility of E. If $c = a$, then a could not have been defended by E in the first place. We can therefore conclude that E' is d–admissible.

Finally, let us show that if an argument $b \in A$ is defended by E, then it is also defended by E' (w.r.t. R''). Assume it is not the case, i.e. b is no longer defended by E'. This means there is an argument $c \in A$ s.t. $cR''b$ and no argument $d \in E'$ s.t. $dR''c$. If there is no such argument in E', then there could not have been one in E. Therefore, E could not have defended b and we reach a contradiction. Hence, if b is defended by E, then it is also defended by E'. □

Lemma 2.76. *Let $BF = (A, R, S)$ be a BAF and $R' \subseteq R^{ind}$ a collection of indirect attacks in BF. If a set $E \subseteq A$ is +conflict–free w.r.t. R', then so is $d - \mathcal{F}_{BF}^{R'}(E)$.*

Proof. Let $E' = d - \mathcal{F}_{BF}^{R'}(E)$. Assume it is not +conflict–free w.r.t. E' and there are arguments $a, b \in E'$ s.t. $(a, b) \in R \cup \bigcup R'$. Due to defense of b, it has to be the case there is $c \in E$ s.t. $(c, a) \in R \cup \bigcup R'$. However, due to defense of a, it also has to be the case that there is $d \in E$ s.t. $(d, c) \in R \cup \bigcup R'$. We thus reach the contradiction with the +conflict–freeness of E. Hence, E' is conflict–free. □

Theorem 2.80. *Let $BF = (A, R, S)$ be BAF and $R' \subseteq R^{ind}$ the collections of indirect attacks in BF. The following holds:*

- *every d–preferred extension of BF w.r.t. (R', R') is a d–complete extension of BF w.r.t. (R', R'), but not vice versa.*
- *the d–grounded extension of BF w.r.t. R' is the least w.r.t. set inclusion d–complete extension of BF w.r.t. (R', R').*
- *every stable extension of BF w.r.t. R' is a d–preferred extension w.r.t. (R', R'), but not vice versa.*

Proof. Please note that for the sake of simplicity, we will not explicitly state parametrization in the proof.

- If a given d–preferred extension E is not d–complete, then there is an argument $a \in A \setminus E$ defended by E. By Lemma 2.72, $E \cup \{a\}$ is d–admissible and clearly $E \subset (E \cup \{a\})$. Thus, E could not have been d–preferred in the first place and we reach a contradiction. Thus, every d–preferred extension is d–complete. In order to show the reverse does not hold, we can adapt Example 1.

- Follows from Lemma 2.77 and Proposition 2.79.

- It can be easily shown that every stable extension is d–admissible. If it is not d–preferred, then there exists another d–admissible extension containing it. However, we can observe that from the properties of the stable semantics, such an extension cannot be +conflict–free. We reach a contradiction. Thus, every stable extension is d–preferred. In order to show the reverse does not hold, we can adapt Example 1.

□

Lemma 2.92. *Let $FN = (A, R, N)$ be an AFN and $E \subseteq A$ be a strongly coherent set. Then $E^{att} \subseteq E^{+}$.*

Proof. Let us assume this is not the case, i.e. an argument $a \in A$ is in E^{att}, but $\nexists e \in E, eRa$ and $\forall C \subseteq A$ s.t. CNa, $C \cap E \neq \emptyset$. It is easy to see that since sufficient support is provided and E is coherent, then $E \cup \{a\}$ would have to be coherent as well. Since $a \in E^{att}$, every coherent set containing a is attacked by E. As E is also conflict–free, it can thus only be the case that $\exists e \in E$ s.t. eRa. We reach a contradiction. Hence, whatever is in E^{att}, is also in E^{+}. □

Lemma 2.94. *Let $FN = (A, R, N)$ be an AFN. A set $E \subseteq A$ is a stable in FN iff it is strongly coherent and $E^{att} = A \setminus E$.*

Proof. Let us show that if E is strongly coherent and $E^{att} = A \setminus E$, then E is stable. By Lemma 2.92 we know that $E^{att} \subseteq E^{+}$. Thus, it suffices to show that E is complete. Since E is strongly coherent, $E \cap E^{att} = \emptyset$. Moreover, from Lemma 2.93 and the fact that $E^{att} = A \setminus E$ it follows that E is at least admissible. Now assume there is an argument $a \notin E$ that is defended by E. Since $a \in E^{att}$, E could not have been conflict–free in the first place. Thus, there cannot be a defended argument not in E. Hence, the set is complete and as a result, also stable.

Let us now show the other way. Since E is complete, it is at least strongly coherent. What remains to be shown is that in this case, whatever is in E^{+} is in E^{att}. Let us assume it is not the case, i.e. there is an argument in $a \in E^{+}$ s.t. E does not attack all coherent sets containing a. Let $(a_0, ..., a_n)$ be a powerful sequence for a that is not attacked by E. Assume that none of the elements of the sequence belong to E. This means that a_0 is in E^{+}, and as it requires no support due to the powerful sequence conditions, it has to be the case that E attacks it. Consequently, the powerful sequence for a would also be attacked by E and we would reach a contradiction. Thus, let us assume that at least a_0

is in E. If a_1 is not there, then by the fact it is supported by a_0 and thus by E we again would reach a conclusion that it can only be the case that E attacks a_1. Consequently, the sequence would again be attacked and we reach a contradiction. We will come to the same conclusion when we assume that a_1 is in E, but a_2 is not. We can continue until we reach $a_n = a$ and it is easy to see that it could not have been the case that a was in E^+, but not in E^{att}. Hence, E is strongly coherent and $E^{att} = A \setminus E$. □

Lemma 2.162. *Let $D = (A, C)$ be an ADF. Every xy–preferred extension of D is a maximal w.r.t. $\subseteq$ xy–complete extension of D for $x, y \in \{a, c\}$.*

Proof. By Theorem 2.158 we know that every xy–preferred extension is xy–complete. If it is not the maximal one, then it means there exists an xy–complete extension containing it. As any xy–complete extension is also xy–admissible, we reach a contradiction with the definition of xy–preferred semantics. □

15.2 Framework Normal Forms & Subclasses: Proof Appendix

Theorem 4.5. *Let $SF = (A, R)$ be a SETAF and $SF^{min} = (A, R')$ its minimal form. A set of arguments $E \subseteq A$ is a σ–extension of SF, where $\sigma \in$ {conflict–free, admissible, preferred, complete, grounded, stable}, iff it is a σ–extension of SF^{min}.*

Proof. First of all, we prove that a set of arguments S attacks an argument a in SF iff it attacks it in SF^{min}. Based on the way R' is obtained, it holds that $\forall (X, a) \in R$, $\exists (X', a) \in R'$ s.t. $X' \subseteq X$. This means that if there exists $S' \subseteq S$ s.t. $S'Ra$, then there exists $S'' \subseteq S'$ s.t. $S''R'a$. Thus, an argument attacked by S in SF also has to be attacked by S in SF^{min}. Since $R' \subseteq R$, it trivially follows that an argument attacked by S in SF^{min} is also attacked by S in SF.

From this analysis it clearly follows that a set S is conflict–free in SF iff it is conflict–free in SF^{min}. It is also easy to prove that a set S is stable in SF iff it is such in SF^{min} and that an argument a is acceptable w.r.t. S in SF iff it is acceptable in SF^{min}. Consequently, admissible, complete and preferred semantics produce the same extensions in both frameworks. Based on completeness and Theorem 2.24 , the same can be shown for the grounded semantics. □

Theorem 4.13. *Let $D = (A, L, C)$ be an ADF and $D^r = (A, L^r, C^r)$ its redundancy–free form. A set $E \subseteq A$ is a σ–extension of D, where $\sigma \in$ {conflict–free, pd–acyclic conflict–free, model, stable, grounded, acyclic grounded, xy–admissible, xy–complete, xy–preferred} and $x, y \in \{a, c\}$ iff it is a σ–extension of D^r. A three–valued interpretation on A is a δ–labeling of D, where $\delta \in$ {three–valued model, admissible, preferred, complete, grounded} iff it is a δ–labeling of D^r.*

Proof Sketch. Let $(b, a) \in L$ be a redundant link, C_a the condition of a and $X \subseteq par(a)$ an arbitrary subset of parents of a. Since $(b, a) \in L$ is supporting, then it cannot be the case that $C_a(X) = in$ and $C_a(X \cup \{b\}) = out$. Since $(b, a) \in L$ is also attacking, then it cannot be the case that $C_a(X) = out$ and $C_a(X \cup \{b\}) = in$. Thus, $C_a(X) = C_a(X \cup \{b\})$. Similarly, $C_a(X) = C_a(X \setminus \{b\})$. Therefore, the presence (or lack of it) of b in no way affects the outcome of the acceptance condition and for any set $E \subseteq A$, $C_a(E \cap par(a)) = C_a^r(E \cap par^r(a))$, where $par^r(a)$ are the parents of a w.r.t. L^r. Consequently, an addition of a redundant link does not alter the behavior of the conditions. Hence, we can observe that the conflict–free extensions, decisive interpretations and evaluations remain the same in both frameworks. Consequently, the extensions under any semantics remain the same. Based on the relation between labelings and decisiveness, as noted for example in Theorems 2.148 and 2.149, it is also easy to see that the labelings remain unaffected.

Theorem 4.15. *Let $D = (A, L, C)$ be an ADF and $D^c = (A', L', C')$ its cleansed form. A set $E \subseteq A$ is a σ–extension of D, where $\sigma \in \{$conflict–free, pd–acyclic conflict–free, model, stable, grounded, acyclic grounded, xy–admissible, xy–complete, xy–preferred$\}$ and $x, y \in \{a, c\}$, iff it is a σ–extension of D^c.*

Proof. Please note that an argument that does not possess a standard evaluation will not appear in any extension. Consequently, every extension E of D is a subset of A'.

We will start with the analysis of decisiveness and different types of evaluations. Let $a \in A$ be an argument that possesses a standard evaluation (F, B) on A in D. Clearly, $F \subseteq A'$. Let v_a be the minimal decisively in interpretation for a used in the construction of (F, B). We will show that v_a and its limitation to A' are decisively in interpretations for a in D^c, though not necessarily minimal ones. We can observe that $v_a^{\mathbf{t}} \subseteq A'$ and $v_a^{\mathbf{t}} \subseteq par(a)$ in D. Consequently, $v_a^{\mathbf{t}} \subseteq par(a)$ in D^c as well. From this, the Definition 4.3 of the reduct and the fact that $C_a(v_a^{\mathbf{t}}) = in$ in D, it holds that $C'_a(v_a^{\mathbf{t}}) = in$ in D^c. Assume v_a is not decisively in for a in D^c; this means there is a set of arguments E where $v_a^{\mathbf{t}} \subseteq E \subseteq A' \setminus v_a^{\mathbf{f}}$ s.t. $C'_a(E \cap par(a)) = out$. Since $E \subseteq A'$, then $E \cap par(a)$ in D is the same as $E \cap par(a)$ in D^c. Consequently, from the definition of the reduct it holds that $C'_a(E \cap par(a)) = C_a(E \cap par(a))$. From the properties of E and the definition of decisiveness it now follows that v_a could not have been decisively in for a in D and we reach a contradiction. Consequently, if v_a is decisively in for a in D and is used in the construction of a standard evaluation for a in A in D, then v_a (or more specifically, its subinterpretation limited to A') is decisively in for a in D^c. Depending on the initial framework, the condition of a can be reduced to an equivalent of a tautology, even if some parents show up as redundant ones. For example, a framework $(\{a, b\}, \{C_a = a \vee \neg b, C_b = \bot\})$ is cleansed into $(\{a\}, \{C_a = a \vee \top\})$. As a result, neither v_a nor its limitation to A' have to be minimally decisively in for a in D^c. Nevertheless, a minimal interpretation can be "extracted", and we can thus conclude that a will have a standard evaluation (F, B') on A' in D^c, where $B' \subseteq B \cap A'$ depends on the "extraction". Since our proof did not in fact depend on F being a set or a sequence, the same analysis can be done in the case of acyclic pd–evaluations. However, the situation is slightly different for the partially acyclic evaluations due to the requirement that the arguments that are in the pd–set are there only if they "really have to" and that the new minimal decisively in interpretations can be subinterpretations of the original ones. Consequently, it is both the $\mathbf{t}$ and $\mathbf{f}$ parts that can be "smaller" and cause a shift between the pd–set and pd–sequence. Thus, given an original partially acyclic evaluation (F, G, B) for an argument $a \in A$, we will obtain a new evaluation (F', G', B') s.t. $F \cup G = F' \cup G'$, $F' \subseteq F$ and $B' \subseteq B \cap A'$. It does not have to be the case that $F = F'$ and $G = G'$. Basically speaking, although the standard evaluations can be adapted from one framework to the other, the pd–functions will look differently and thus the partially acyclic evaluations representing the standard ones can separate pd–set and pd–sequence differently.

Let now $a \in A'$ be an argument that possesses a standard evaluation (F, B) on A in D^c and let v_a be the minimal decisively in interpretation for a used in the construction of (F, B). Assume v_a is defined for a set $E \subseteq A'$; let now z_a be the $\mathbf{f}$–completion of v_a to $E \cup (A \setminus A')$. We will show that z_a is decisively in for a in D. From the definition of the reduct and the fact that v_a and z_a have the same $\mathbf{t}$ mappings it follows that if $C'_a(v_a^{\mathbf{t}}) = in$ in D^c, then $C_a(z_a^{\mathbf{t}}) = in$ in D^c. Now assume that there is some set of arguments E where $z_a^{\mathbf{t}} \subseteq E \subseteq A \setminus z_a^{\mathbf{f}}$ s.t. $C_a(E \cap par(a)) = out$. From the construction of z_a it means that $v_a^{\mathbf{t}} \subseteq E \subseteq A \setminus v_a^{\mathbf{f}}$. Again, from the definition of reduct, if such a set outs the condition of a in D, then it does so in D^c as well and thus we contradict the decisiveness of v_a. Therefore, z_a is decisively in for a in D, even if not minimally. Nevertheless, a minimal interpretation can be "extracted", and we can conclude that a will have a standard evaluation (F, B') on A in D, where $B \subseteq B' \subseteq (B \cup A \setminus A')$ depends on the "extraction". Again, as the analysis does not really depend on whether F is a set or a sequence, the same analysis can be done in the case of acyclic pd–evaluations. Finally, when it comes to the partially acyclic evaluations, we can transform (F, G, B) into (F', G', B') s.t. $B \subseteq B' \subseteq (B \cup A \setminus A')$ and $F \cup G = F' \cup G'$. Moreover, from the presented analysis it follows that in this case, also $F = F'$ and $G = G'$ holds.

Let $E \subseteq A$ be a conflict–free extension of D. E can be seen as a standard evaluation (E, B) in D s.t. $B \subseteq A$ and $E \cap B = \emptyset$. Consequently, $E \subseteq A'$. From the previous parts of this proof it also follows that there is a standard evaluation (E, B') s.t. $B' \subseteq B$ in D^c. Since $E \cap B = \emptyset$, then $E \cap B' = \emptyset$ and the evaluation is not self–blocking. Thus, it can again be seen as a conflict–free set and E is conflict–free in D^c. The same analysis holds for pd–acyclic conflict–free extensions, but with acyclic pd–evaluations (see discussion in Section 2.3.5).

Let $E \subseteq A'$ be a conflict–free extension of D^c. Again, E can be seen as a standard evaluation (E, B) in D s.t. $B \subseteq A'$ and $E \cap B = \emptyset$. From the previous parts of this proof it also follows that there is a standard evaluation (E, B') in D s.t. $B \subseteq B'$ and $(B' \setminus B) \subseteq (A \setminus A')$. Consequently, as $E \cap B = \emptyset$ and $E \cap (A \setminus A') = \emptyset$, then $E \cap B' = \emptyset$ and the evaluation is not self–blocking. Thus, it can again be seen as a conflict–free set, and we can conclude that E is conflict–free in D. The same analysis holds for the pd–acyclic conflict–free extensions, but with acyclic pd–evaluations (see discussion in Section 2.3.5).

Let $E \subseteq A'$ be a conflict–free extension of D^c and D. We will show how the discarded sets between the frameworks are related. First of all, let E_c^+ and E^+ and be the standard discarded sets of E in D^c and D respectively. We will show that $E^+ = E_c^+ \cup (A \setminus A')$.

An argument $a \in (A \setminus A')$ by construction does not possess a standard evaluation and thus is trivially in E^+. Let now $a \in E_c^{a+}$; from the proofs above we could have observed that every standard evaluation (F, B) in D^c can be "extended" through the blocking set to a standard evaluation for a in D and that every standard evaluation (F, B') for a in D could have been "trimmed" through the blocking set to an evaluation for a in D^c. Since a is in the discarded set, then by Lemma 2.125 all of its standard evaluations are blocked through the blocking set by E in D^c, and by the explanation above it has to be the case that all standard evaluations for a in D are blocked through the blocking set by E. Consequently, $a \in E^+$. Let now $a \in E^+ \cap A'$. Assume that a is not in E_c^+. By Lemma 2.125 it means it possesses a standard evaluation (F, B) on A' s.t. $B \cap E = \emptyset$. By the proofs above, this evaluation can be extended to a standard evaluation (F, B') on A in D s.t. $B \subseteq B' \subseteq (B \cup A \setminus A')$. As $E \subseteq A'$ and $E \cap B = \emptyset$, then $B' \cap E = \emptyset$ and (F, B') cannot be blocked by E in D. Consequently, a could not have been in E^+ in the first place and we reach a contradiction. This brings us to the conclusion that $E^+ = E_c^+ \cup (A \setminus A')$. Exactly the same analysis can be done for acyclic pd–evaluations and thus the acyclic discarded sets between D and D^c are also related: $E^{a+} = E_c^{a+} \cup (A \setminus A')$.

Let us now focus on the partially acyclic discarded sets and show that $E^{p+} = E_c^{p+} \cup (A \setminus A')$ as well. An argument $a \in (A \setminus A')$ by construction does not possess a standard evaluation. Therefore, it does not possess a partially acyclic one as well and thus is trivially in E^{p+}. Let now $a \in E_c^{p+}$; this means that there is no unblocked partially acyclic evaluation for a s.t. its pd–set is in E. From the previous parts of the proof we could have observe that all partially acyclic evaluations for a in D^c can be extended through the blocking set to evaluations for a in D and every partially acyclic evaluation for a in D can be trimmed to one in D^c in a way that the resulting pd–set is a subset of the original one. Consequently, if no unblocked partially acyclic evaluation for a in D^c has a pd–set in E, then neither has one in D. Therefore, if $a \in E_c^{p+}$, then $a \in E^{p+}$. Let now $a \in E^{p+}$. Assume that a has a partially acyclic evaluation (F, G, B) on A' in D^c s.t. $B \cap E = \emptyset$ and $F \subseteq E$. This means it also has a partially acyclic evaluation (F, G, B') on A in D s.t. $B \subseteq B' \subseteq (B \cup A \setminus A')$. Since $(A \setminus A') \cap E = \emptyset$, then $B' \cap E = \emptyset$. Thus, (F, G, B') is not blocked by E in D and has a pd–set in E. This means a could not have been in E^{p+} in the first place and we reach a contradiction. Therefore, $E^{p+} = E_c^{p+} \cup (A \setminus A')$.

Let us move on to admissibility. Let us start with the cc case; assume that $E \subseteq A'$ is admissible in D, but not in D^c. Since E is conflict–free, this means that there is an argument $a \in E$ that is not decisively in w.r.t. the standard range interpretation. Thus, there exists a set E' in D^c s.t. $E \subseteq E' \subseteq (A' \setminus E_c^+)$ for which $C'_a(E' \cap par(a)) = out$ in D^c. However, from the previous parts of this proof it can be observed that this means that $C_a(E' \cap par(a)) = out$ in D. Due to the relation between the discarded sets and the fact that $E \subseteq E' \subseteq (A' \setminus E_c^+)$, it has to be the case that $E' \cap E^+ = \emptyset$. Consequently, a could not have been decisively in w.r.t. the range interpretation of E in D and thus E cannot be cc–admissible in D in the first place. Hence, we reach a contradiction. In a similar manner we can prove ca$_1$ and ca$_2$ admissibility. Let us now assume that E is an aa–admissible extension of D. Assume it is not aa–admissible in D^c;

since it is already pd–acyclic conflict–free, it means that there is an argument $a \in E$ s.t. none of its acyclic pd–evaluations (F, B) on E in D^c has a blocking set contained in the acyclic discarded set. However, based on the presented relations between the evaluations and the acyclic discarded sets in D and D^c, we can observe that if no evaluation for a meets admissibility requirements in D^c, then no evaluation can meet the admissibility requirements in D either. Thus, E could not have been aa–admissible in D in the first place and we reach a contradiction. Therefore, if a set of arguments is aa–admissible in D, then it is admissible in D^c. The same analysis can be repeated for ac–admissibility.

Let now E be cc–admissible in D^c; assume it is not cc–admissible in D. This means that there is an argument $a \in E$ that is not decisively in w.r.t. the standard range interpretation. Consequently, there exists a set E' s.t. $E \subseteq E' \subseteq (A \setminus E^+)$ for which $C_a(E \cap par(a)) = out$ in D. From the relation between discarded sets of D and D^c and the fact that any argument in $A \setminus A'$ is trivially in E^+, it holds that $A \setminus E^+ = A' \setminus E_c^+$. Moreover, as $C'_a(E \cap par(a)) = out$ in D^c, then a could not have been decisively in w.r.t. standard range interpretation of E in D^c and the set could not have been cc–admissible. We reach a contradiction. The same analysis can be performed for ca_1 and ca_2 semantics. Let now E be aa–admissible in D^c. From the previous parts of this proof we could observe that any acyclic pd–evaluation for $a \in E$ in D^c can be extended into one in D by adding some of the $A \setminus A'$ elements to the blocking set. Since these elements are disjoint from E and trivially in E^{a+}, we can observe that if an acyclic pd–evaluation for a satisfies admissibility requirements in D^c, then so does its "extension". Consequently, E is aa–admissible in D^c. Similar analysis can be done for the ac–admissible semantics.

We have shown that the cc–, ac–, aa–, ca_1 and ca_2–admissible extensions between our two frameworks coincide. We now need to show that it also holds for the respective complete semantics. Let $E \subseteq A'$ be a cc–complete extension of D. If it is not cc–complete in D^c, it means there is an argument $a \in A' \setminus E$ that is decisively in w.r.t. the standard range interpretation v_E of E in D^c. We can observe that since a is decisively in w.r.t. v_E and E can be seen as a standard evaluation, then a possesses a standard evaluation in D. Thus, we can use the previously done analysis to show that v_E extended with **f** mappings on $A \setminus A'$ is decisively in for a in D. Since this extended interpretation is also the standard range interpretation of E in D then E could not have been cc–complete in D in the first place. Similar analysis can be done for ac–, aa and ca_2–complete semantics. Since $A \setminus (E \cup E^{a+}) = A' \setminus (E \cup E_c^{a+})$, the proof can also be repeated for ca_1–complete semantics.

Let now E be cc–complete in D^c. Assume it is not complete in D; this means there is an argument $a \in A \setminus E$ that is decisively in w.r.t. standard range interpretation v_E of E in D. Since an argument not possessing a standard evaluation will be decisively out w.r.t. v_E (see Lemma 2.125 and Proposition 2.150), then only $a \in A' \setminus E$ are the possible candidates. Now, as a is decisively in w.r.t. the range interpretation, then we can remove the $A \setminus A'$ assignments from this interpretation and obtain one that is decisively in for a. Moreover, it will also be the standard range of E in D^c. Consequently, a would have been decisively in w.r.t. the standard range of E of D^c and we reach a contradiction with the completeness of E. Therefore, if E is cc–complete in D^c, then it is complete in D. The proof for the remaining complete semantics is similar; we only need to notice that an argument not possessing a standard evaluation will be decisively out w.r.t. remaining types of range interpretations as well (see Lemma 2.132).

We now move to the model and stable semantics. Let $E \subseteq A$ be a model extension of D. It is conflict–free in D^c and by Lemma 2.139 and the relation between the partially acyclic discarded sets, it holds that $E_c^{p+} = A' \setminus E$. Thus, by Proposition 2.150, it holds that for every argument $a \in A' \setminus E$, $C'_a(E \cap par(a)) = out$. Consequently, E is a model of D^c. The proof that every model of D^c is a model of D follows similarly. Since the model and pd–acyclic conflict–free extensions between D and D^c coincide, so do the stable ones by Theorem 2.138.

The fact that $E \subseteq A'$ is a preferred extension of D iff it is one in D^c follows straightforwardly from the relation between the admissible extensions. The coincidence of the acyclic grounded and grounded extensions between the two frameworks is a result of the relation between complete extensions and Theorem 2.158. □

Theorem 4.7. *Let $EFC = (A, R, D)$ be an EAFC and $EFC^{min} = (A, R, D)$ its minimal form. A set of arguments $E \subseteq A$ is a σ–extension of EFC, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, iff it is a σ–extension of EFC^{min}.*

Proof. First of all, we prove that a set of arguments E defeats$_E$ an argument a in EFC iff it defeats$_E$ it in EFC^{min}. We can observe that the attack relation is the same in both frameworks. Therefore, if an argument $b \in E$ attacks a in R and there is no defense attack for it in D, then there is no defense attack for it in D' and a is defeated$_E$ by b both in EFC and EFC^{min}. If there is a defense attack for the (b, a) conflict in D, then it cannot be contained in E. Thus, none of the minimal defense attacks is included either, and therefore no set of arguments defense attacking (b, a) in D' is in E. Hence, if E defeats$_E$ a in EFC, then it defeats$_E$ in EFC^{min} as well. Let us now focus on the other direction. If there is no defense attack at all in D' for the (b, a) conflict, then by the construction of D', there could not have been one in D either. Therefore, the defeat occurs in both frameworks. If there is a defense attack by a set $C \subseteq A$ on (b, a) in D', but $C \not\subseteq E$, then clearly no set C' s.t. $C \subseteq C'$ can be contained in E either. Consequently, from the construction of D' we can observe that if no defense attack for (b, a) was present in E in EFC^{min}, then no defense attack is present in E in EFC either. Hence, if an argument is defeated$_E$ in EFC^{min}, it is defeated$_E$ in EFC as well.

Due to the relation between the defeats in both frameworks, we can observe that the reinstatement sets of the defeats will also be the same. Let $\{(x_1, y_1), ..., (x_n, y_n)\}$ be a reinstatement set on E for the x_n-y_n defeat in EFC This means that for every set $C \subseteq A$ of arguments defense attacking any of the (x_i, y_i) defeats in the set in D, there is an (x_j, y_j) in the set s.t. $y_j \in C$. As this holds for every such C, then also for the minimal ones. Thus, from the fact that the defeats are the same both in EFC and EFC^{min}, $\{(x_1, y_1), ..., (x_n, y_n)\}$ will be a reinstatement set on E for x_n-y_n defeat in EFC^{min} as well.

Let now $\{(x_1, y_1), ..., (x_n, y_n)\}$ be a reinstatement set on E for the x_n-y_n defeat in EFC^{min}. This means that for every set $C \subseteq A$ of arguments defense attacking any of the (x_i, y_i) defeats in the set in D', there is an (x_j, y_j) in the set s.t. $y_j \in C$. We can observe that by construction, every defense attack in D has a "smaller" defense attack in D' and every defense attack in D' has a "bigger" one in D. Therefore, if every relevant defense attack is defeated by the reinstatement set in EFC^{min}, then every defense attack will be defeated by the same set in EFC as well.

Using the correspondence between the defeats and reinstatement sets it can be easily shown that the extensions under conflict–free, admissible, preferred, complete, grounded and stable semantics coincide between EFC and EFC^{min}. $\square$

Theorem 4.16. *Let $D = (A, L, C)$ be an ADF and $D^c = (A', L', C')$ its cleansed form. If v is σ–labeling of D, where $\sigma \in \{$three–valued model, admissible, preferred, complete, grounded$\}$, then $v|_{A'}$ is a σ–labeling of D^c* [30]*. If v is a σ–labeling of D^c, then the* **f***–completion of v to A is a σ–labeling of D.*

Proof. Let v be a three–valued model of D and $v' = v|_{A'}$ its restriction to A'. From the Definition 4.3 and the analysis done in the proof of Theorem 4.15 we can observe that $v^{\mathbf{t}} \subseteq A'$ and that if $C_a(v^{\mathbf{t}} \cap par(a)) = in$ then $C'_a(v^{\mathbf{t}} \cap par(a)) = in$ and if $C_a(v^{\mathbf{t}} \cap par(a)) = out$ then $C_a(v^{\mathbf{t}} \cap par(a)) = out$ for $a \in A'$. Consequently, v' will be a three valued model of D^c.

Let now v be a three–valued model of D^c. Again, from the previous analysis and the nature of $v^{\mathbf{t}}$ we can observe that if $C'_a(v^{\mathbf{t}} \cap par(a)) = in$ then $C_a(v^{\mathbf{t}} \cap par(a)) = in$ and if $C'_a(v^{\mathbf{t}} \cap par(a)) = out$ then $C'_a(v^{\mathbf{t}} \cap par(a)) = out$. If the $\mathbf{f}$–completion of v to A is not a three–valued model, this means that there is an argument $a \in A \setminus A'$ s.t. $C_a(v^{\mathbf{t}} \cap par(a)) = in$. Thus, a possesses a minimal decisively in interpretation s.t. its $\mathbf{t}$ part is a subset of $v^{\mathbf{t}}$. Since $v^{\mathbf{t}}$ is a conflict–free extension of D by Theorem 2.165, it can be represented with a standard evaluation (see Section 2.3.5). This evaluation can be extended with the minimal decisively interpretation for a we have described and will form a new standard evaluation for a. Consequently, a could not have been in $A \setminus A'$ in the first place and we reach a contradiction. Thus, the $\mathbf{f}$–completion of v to A is a three–valued model of D.

Let v be an admissible labeling of D and $v' = v|_{A'}$ its restriction to A'. Assume v' is not admissible in D^c; by Theorem 2.148 this means there is an argument $a \in v'^{\mathbf{t}}$ that is not decisively in w.r.t. the two–valued subinterpretation of v' or an argument $b \in v'^{\mathbf{f}}$ that is not decisively out w.r.t. the two–valued subinterpretation of v'. If it is the first case, then there exists a two–valued interpretation $z \in [v']_2$ s.t. $C'_a(z^{\mathbf{t}} \cap par(a)) = out$. Since $z^{\mathbf{t}} \subseteq A'$, then from the definition of the reduct it means that $C_a(z^{\mathbf{t}} \cap par(a)) = out$ and as the $\mathbf{f}$–completion of z to A is in $[v]_2$, then a could not have been decisively in w.r.t. the two–valued subinterpretation of v and by Theorem 2.148 v could not have been admissible in D. Similar analysis can be done for b. We can thus conclude that v' is admissible in D^c.

[30] Recall that $v|_A$ stands for the subinterpretation of v defined over A.

Let now v be an admissible labeling of D^c and let v' be its $\mathbf{f}$–completion to A. Assume that v' is not admissible in D. By Theorem 2.148 this means there is an argument $a \in v'^{\mathbf{t}}$ that is not decisively in w.r.t. the two–valued subinterpretation of v' or an argument $b \in v'^{\mathbf{f}}$ that is not decisively out w.r.t. the two–valued subinterpretation of v'. If it is the first case, then $a \in A'$ and there exists a two–valued interpretation $z \in [v']_2$ s.t. $C_a(z^{\mathbf{t}} \cap par(a)) = out$ in D. Since all arguments in $A \setminus A'$ are assigned $\mathbf{f}$, this means that $z^{\mathbf{t}} \subseteq A'$. Consequently, the restriction of z to A' is in $[v]_2$ and $C'_a(z^{\mathbf{t}} \cap par(a)) = out$ in D^c. Therefore, v could not have been admissible in D^c. Let us now focus on b. If $b \in A'$, then the proof follows similarly as in the case of a. Let us thus assume that $b \in A \setminus A'$ and let z be the two–valued interpretation in $[v']_2$ s.t. $C_b(z^{\mathbf{t}} \cap par(b)) = in$. We can observe that $z^{\mathbf{t}} \subseteq A'$. Consequently, from z we can extract a minimal decisively in interpretation for b that has a $\mathbf{t}$ part in A'. Since all elements in A' possess standard evaluations and A' itself can be represented as one, we can obtain a standard evaluation for b by extending this evaluation with the interpretation extracted from z. Thus, b has a standard evaluation and thus cannot be in $A \setminus A'$ and we reach a contradiction. Therefore, v' is an admissible labeling of D.

Let v be a complete labeling of D. Before we continue, we will first show that for every $a \in A \setminus A'$, $v(a) = \mathbf{f}$. Since v is a complete labeling, it is also a three–valued model and thus $E = v^{\mathbf{t}}$ is conflict–free in D. We will show that the standard range interpretation of E is a subinterpretation of v. Let us start with an interpretation z consisting only of $\mathbf{t}$ mappings on E and go through the original definition of range (Definition 2.124). Clearly, z is a subinterpretation of v. Let $B_1 \subseteq A \setminus E$ be the set of arguments decisively out w.r.t. z. From the definition of decisiveness it follows that they need to be decisively out w.r.t. the two–valued subinterpretation of v as well. Thus, by the completeness of v and Theorem 2.149, $\forall b \in B_1, v(b) = \mathbf{f}$. Extend z with $\mathbf{f}$ mappings for the elements in B_1. Again, z is a subinterpretation of v. Let $B_2 \subseteq A \setminus (E \cup B_1)$ be the set of arguments decisively out w.r.t. the new z. By decisiveness, they are also decisively out w.r.t. the two–valued subinterpretation of v and again all elements of B_2 have to be mapped to $\mathbf{f}$ by v. We can now extend z with $\mathbf{f}$ mappings for B_2; again, z will be a subinterpretation of v. We can continue reasoning in this manner till the set of decisively out arguments is empty. In this way we have computed the standard range interpretation of E and shown that it is a subinterpretation of v. From Lemma 2.125 we can see that every argument not possessing a standard evaluation is in the standard range. Consequently, for every $a \in A \setminus A'$, $v(a) = \mathbf{f}$.

Let now v be a complete labeling of D and $v' = v|_{A'}$ its restriction to A'. We will show that v' is complete in D^c. By Theorem 2.149, an argument in $v^{\mathbf{u}}$ can neither be decisively in nor decisively out. Thus, for every $a \in v^{\mathbf{u}}$, there is a set of arguments E s.t. $v^{\mathbf{t}} \subseteq E \subseteq A \setminus v^{\mathbf{f}}$ for which it holds that $C_a(v^{\mathbf{t}} \cap par(a)) \neq C_a(E \cap par(a))$. Since $(A \setminus A') \subseteq v^{\mathbf{f}}$ by the proof above, then $E \subseteq A'$. Therefore, from the previous parts of the proof it follows that $C'_a(v^{\mathbf{t}} \cap par(a)) \neq C'_a(E \cap par(a))$ in D^c. Consequently, there is no argument in $v'^{\mathbf{u}}$ that is decisively in or decisively out w.r.t. the two–valued subinterpretation of v'. Thus, by Theorem 2.149, v' is complete in D^c. Showing that the

$\mathbf{f}$–completion of a complete labeling of D^c is a complete labeling of D follows similarly.

From the bijective nature of the relation between the complete extensions of D and D^c and the fact they only differ by $\mathbf{f}$ mappings on arguments in $A \setminus A'$, it should be clear that the maximal and least complete labelings are related in the same manner. Thus, by Theorem 2.147, a labeling v is preferred or grounded in D if its restriction to A' is preferred or grounded in D^c and that the $\mathbf{f}$–completion to A of a labeling v' that is preferred or grounded in D^c, is preferred or grounded in D. □

Theorem 4.50. *Let $FN = (A, R, N)$ be an AFN and $FN^{sc} = (A', R, N')$ its corresponding strongly consistent framework obtained through Translation 13. Let $E \subseteq A$, $E' \subseteq A'$ be sets of arguments and E^b the (possibly empty) set of bypass arguments generated by E in A'. If E is a σ–extension of FN, where $\sigma \in$ {conflict–free, coherent, admissible, preferred, complete, grounded, stable}, then $E \cup E^b$ is a σ–extension of FN^{sc}. If E' is a σ–extension of FN^{sc}, then $E' \setminus E^b$ is a σ–extension of FN.*

Proof. It is easy to see that conflict–freeness is preserved in both ways due to the fact that the attack relation R remains unchanged; the bypass arguments can appear randomly in any conflict–free extensions as they do not take part in any attacks.

Let us now focus on analyzing the coherent semantics. Let $a \in A$ be an argument and let us assume that there exists a powerful sequence $(a_0, ..., a_n)$ for a on a set $B \subseteq A$. We create a new sequence on $B \cup B^b$ by inserting bypass arguments just after the arguments for which they were created for, with the exception of the bypass for a_n. For example, given an argument a_i for which a bypass a_i^b was created, $(a_0, ..., a_i, a_{i+1}, .., a_n)$ becomes $(a_0, ..., a_i, a_i^b, a_{i+1}, .., a_n)$. We will show that the resulting bypass sequence is still a powerful sequence for a. Since a_0 required no support in N in the first place, it was a strongly consistent argument by default and thus is not affected by the translation. Consequently, it needs no support in N' and meets the first powerful criterion. The sequence ends with $a_n = a$, thus meeting the endpoint requirement as well. Let a_i be a non–zero argument in the sequence. If a_i is a bypass argument, then naturally the powerful condition is satisfied - a_i is supported only by its creator, which by our method appears before it in the sequence. If it is not a bypass argument, then in the original sequence it was the case that $\forall C \subseteq A$ s.t. CNa_i it holds that $C \cap \{a_0, ..., a_{i-1}\} \neq \emptyset$. Should such C sets be transformed by replacing the arguments with their bypasses in the translation, then it is easy to see that since bypasses appear just after their “owners” and before a_i, the requirement is again satisfied. Thus, if a was coherent in B, then it is also coherent in $B \cup B^b$. By a similar reasoning we can show that if $a \in A$ is coherent on $B' \subseteq A'$ in FN^{sc}, then it is coherent on $B' \cap A$ in FN. Moreover, it also holds that if a is coherent, then so is its bypass argument if it exists – we can just extend the bypass sequence of a by putting a^b on the a_{n+1} position.

We will now show that if an argument $a \in A$ is defended by E in FN, then it is also defended by $E \cup E^b$ in FN^{sc}. The coherence requirement of the defense follows straightforwardly from the previous parts of this proof. Moreover, it is also easy to see that we cannot create a powerful sequence in FN^{sc} that would not be present in FN after the removal of bypass arguments – every bypass needs to be supported by its original argument. Since the attack relation R remains unchanged and bypass arguments do not participate in conflicts, we can observe that a coherent set C attacked in FN is attacked both in C and $C \cup C^b$ forms in FN^{sc} and vice versa. Consequently, if we can defend an argument in FN, then we can also defend it in FN^{sc} after including the required bypass arguments. We can also observe that if an argument a is defended by E in FN, then so is a^b by $E \cup E^b \cup \{a\}$ in FN^{sc}. The coherence requirement follows from the explanation above, and since a^b is not attacked in FN^{sc}, it is defended by $E \cup E^b \cup \{a\}$.

Let us now focus on the other way around; i.e. showing that if an argument $a \in A$ is defended by $E \subseteq A'$ in FN^{sc}, then it is also defended by $E \cap A$ in FN. By the coherence analysis above, we can see that if $E \cup \{a\}$ is coherent in FN^{sc}, then so will be $(E \cap A) \cup \{a\}$ in FN – we simply remove the bypass arguments. Again, due to the fact that the attack relation is the same in the two frameworks, then E in FN^{sc} carries out and suffers from the same attacks as $E \cap A$ in FN. Thus, we can observe that the defense from attacks in FN^{sc} implies defense from attacks in FN. We can therefore conclude that if a (non–bypass) argument is defended by a set in one framework, then it is also defended by this set without bypass arguments in the other structure.

Due to the coherence, conflict–freeness and defense proofs presented above, we can observe that if a set of arguments $E \subseteq A$ is admissible (complete) in FN, then $E \cup E^b$ is admissible (complete) in FN^{sc} and that if a set of arguments $E' \subseteq A'$ is admissible (complete) in FN^{sc}, then so is $E' \cap A$ in FN. Moreover, since the defense of an argument implies the defense of its bypass and a bypass will not appear in an admissible extension without its original argument, it follows that any complete extension in FN^{sc} satisfies the following property: $E = (E \cap A) \cup (E \cap A)^b$. In other words, it is precisely of the form $E \cup E^b$ for some $E \subseteq A$ and we obtain a one–to–one correspondence between the target and source extensions. What remains to be shown is the correspondence between preferred, grounded and stable semantics.

Let us assume that E is preferred in FN, but $E \cup E^b$ is not preferred in FN^{sc}. Since it is already complete, it means there is another admissible extension $E' \subseteq A'$ that contains at least one argument more than $E \cup E^b$. We can observe that it cannot be the case that $E = E' \cap A$. This would mean that E' contains a bypass argument not present in $E \cup E^b$, and thus one without its origin in E'. This clearly breaches the coherence of this extension. From this and the fact that $E \cup E^b \subseteq E'$, it holds that $E \subset E' \cap A$. Since $E' \cap A$ is admissible in FN, then E could not have been preferred in the first place. We reach a contradiction with the assumptions. Thus, if E is preferred in FN, then $E \cup E^b$ is preferred in FN^{sc}.

Let us now look at the other way around, i.e. showing that if $E \subseteq A'$ is preferred in FN^{sc}, then $E \cap A$ is preferred in FN. It is already admissible; if it is not maximal, then we have another $E' \subseteq A$ admissible extension of FN s.t. $(E \cap A) \subset E'$. From previous parts of this proof it should be clear that $(E' \cup E'^b)$ is admissible in FN^{sc} and that $E \subset (E' \cup E'^b)$. Thus, E could not have been preferred in FN^{sc} in the first place and we reach a contradiction.

The argument for the grounded extensions follows the same line of reasoning as for the preferred semantics. The grounded extension $E \subseteq A$ of FN is by Theorem 2.95 the least w.r.t. $\subseteq$ complete extension of FN. Moreover, $E \cup E^b$ is complete in FN^{sc}. If it is not grounded, then it means there exists a complete extension $E' \subseteq A'$ of FN^{sc} s.t. $E' \subset (E \cup E^b)$. From the defense proofs we can observe that E' has to contain any bypass arguments with its origin in E' and no bypass can appear without its origin. Consequently, it has to be the case that a "normal" argument is not present in E' and thus $(E' \cap A) \subset E$. As $E' \cap A$ is complete in FN, then E cannot be a grounded extension of FN. We thus reach a contradiction and it holds that if E is grounded in FN, then $E \cup E^b$ is grounded in FN^{sc}. Showing that if $E \subseteq A'$ is grounded in FN^{sc} then $E \cap A$ is grounded in FN follows the same line of reasoning as in the previous proofs.

Let us now focus on the stable semantics. From the earlier parts of this proof it follows that if $E \subseteq A$ is strongly coherent in FN, then $E \cup E^b$ is strongly coherent in FN^{sc} and if $E' \cup A'$ is strongly coherent in FN^{sc}, then so is $E' \cap A$ in FN. Moreover, if all the coherent sets for an argument $b \in A$ are attacked by E in FN, then all the coherent sets for b and b^b are also attacked by E (and $E \cup E^b$) in FN^{sc}. Similarly, if all coherent sets for an argument $b \in A'$ are attacked by E' in FN^{sc}, then all the coherent sets for b (or its origin, if it is a bypass) are attacked by $E' \cap A$ in FN. Consequently, if E attacks all the coherent sets of all arguments in $A \setminus E$ in FN, then also $E \cup E^b$ attacks all coherent sets of all arguments in $A' \setminus (E \cup E^b)$ in FN^{sc} and vice versa Thus, by Lemma 2.94, if E is stable in FN then so is $E \cup E^b$ in FN^{sc} and if E' is stable in FN^{sc}, then so is $E' \cap A$ in FN. □

Theorem 4.51. *Let $FN = (A, R, N)$ be an AFN and $FN^{sc} = (A', R, N')$ its bypass consistency form obtained through Translation 13. FN is weakly, relation and strongly valid iff FN^{sc} is. FN is in minimal form iff FN^{sc} is.*

Proof. Validity forms can be shown to hold using the proof of Theorem 4.50. If sets of arguments supporting a given argument were incomparable before the translation, then they remain incomparable afterwards. Moreover, the bypass arguments are supported by a single set only. Thus, minimality of FN^{sc} follows easily. Similar holds for the other way around due to the fact it is all occurrences in support sets of a given argument that are replaced, not just one. □

Theorem 4.54. *Let $ES = (A, R, E)$ be an EAS and $ES^{sc} = (A', R, E')$ its corresponding strongly consistent framework obtained through Translation 14. Let $S \subseteq A$, $S' \subseteq A'$ be sets of arguments and S^b the (possibly empty) set of bypass arguments generated by S in A'. If S is a σ–extension of ES, where $\sigma \in \{$conflict–free, self–supporting, admissible, preferred, complete, grounded, stable$\}$, then $S \cup S^b$ is a σ–extension of ES^{sc}. If S' is a σ–extension of ES^{sc} then $S' \setminus S^b$ is a σ–extension of ES.*

Proof. It is easy to see that conflict–freeness is preserved in both ways due to the fact that the attack relation R remains unchanged; the bypass arguments can appear randomly in any conflict–free extensions as they do not take part in any attacks.

We will now describe how e–support is preserved by the transformation. Let $a \in A$ be an argument and let us assume that there exists an evidential sequence $(\eta, ..., a_n)$ for a on some set $B \subseteq A$. We create a new sequence on $B \cup B^b$ by inserting bypass arguments, with the exception of the bypass of a_n. For example, given an argument a_i for which a bypass a_i^b was created, $(a_0, ..., a_i, a_{i+1}, .., a_n)$ becomes $(a_0, ..., a_i, a_i^b, a_{i+1}, .., a_n)$. We will show that the resulting sequence with bypasses is then still an evidential sequence for a. Since η required no support in E in the first place, it was a strongly consistent argument and thus is not affected by the translation. Consequently, it needs no support in E' and meets the first criterion. The sequence ends with $a_n = a$, thus meeting the endpoint requirement. Let a_i be a non–zero argument in the sequence. If a_i is a bypass argument, then naturally the evidential condition is satisfied - a_i is supported only by its creator, which by our method appears before it in the sequence. If it is not a bypass argument, then in the original sequence it was the case that $\exists C \subseteq A$ s.t. CEa_i and $C \subseteq \{a_0, ..., a_{i-1}\}$. If the bypasses for arguments in C were introduced and C replaced by C' in E' support, then since the bypasses appear just after their "owners" and before a_i, it holds that $\exists C' \subseteq A'$ s.t. $C'E'a_i$. If no bypasses were required, then $CE'a_i$ and in both of the cases the support requirement is satisfied. Thus, if a has an evidential sequence on B, then it also has one on $B \cup B^b$. Consequently, due to the relation between evidential sequences and e–support (Theorem 2.99), if B is self–supporting in ES, then $B \cup B^b$ is self–supporting in ES^{sc}. By a similar reasoning we can show that if $a \in A$ has an evidential sequence on $B' \subseteq A'$ in ES^{sc}, then it has one on $B' \cap A$ in ES. Therefore, if B' is self–supporting in ES^{sc}, then $B' \setminus A^b$ is self–supporting in ES. Moreover, it also holds that if a has an evidential sequence on B' in ES^{sc}, then so does its bypass argument on $B' \cup a^b$ if it exists – we can just easily add it to the end of the sequence.

We will now show that if an argument $a \in A$ is acceptable w.r.t. a set $S \subseteq A$ in ES, then it is also acceptable w.r.t. $S \cup S^b$ in ES^{sc} and the other way around. The fact that if a is e–supported by $S \cup \{a\}$ in ES then it is e–supported by $S \cup S^b \cup \{a\}$ in ES^{sc} and vice versa follows straightforwardly form the previous proof on self–supporting sets. Since the attack relation R remains unchanged, if a set T attacks a in ES, then T attacks a in ES^{sc} as well and vice versa. By adding the self–support analysis we can conclude that if a set T carries out an e–supported attack on a in ES, then so does $T \cup T^b$ in ES and the other way around. From this it follows easily that an argument $a \in A$ that acceptable w.r.t. S in ES will also be acceptable w.r.t. $S \cup S^b$ in ES^{sc} and if it is acceptable w.r.t. $S' \subseteq A'$ in ES^{sc}, it will also be acceptable w.r.t. $S' \cap A$ in ES. Finally, we can also observe that if argument a is acceptable w.r.t. S in ES, then so is a^b w.r.t. $S \cup S^b \cup \{a\}$ in ES^{sc}. The e–support requirement follows from the explanation above, and since a^b is not attacked in ES^{sc}, defense follows.

By the three properties above, we can see that any admissible or complete extension in ES is also admissible or complete in ES^{sc} upon extending by its bypass arguments and any admissible or complete extension of ES^{sc} will be admissible or complete in ES upon removing the bypass arguments. From the relation between the defense of an argument, the defense of its bypass and the fact that a bypass argument cannot appear in an extension without its origin being present, we can observe that any complete extension in ES^{sc} will be of the form $S \cup S^b$, where $S \subseteq A$ is a complete extension of ES.

Let us now assume that S is preferred in ES, but $S \cup S^b$ is not preferred in ES^{sc}. This means there is an admissible extension $S' \subseteq A'$ in ES^{sc} s.t. $S \cup S^b \subset S'$. We can observe this means that $S \subseteq (S' \cap A)$. If $S = S' \cap A$, then S' has to contain an auxiliary bypass argument without its origin and thus cannot be self–supporting to start with. Thus, $S \subset (S' \cap A)$ and as $S' \cap A$ is admissible in ES, S could not have been a preferred extension in the first place and we reach a contradiction. Let us now assume that $S \subseteq A'$ is preferred in ES^{sc}, but $S \cap A$ is not preferred in ES. This means there exists another admissible extension $S' \subseteq A$ of ES s.t. $(S \cap A) \subset S'$. The corresponding ES^{sc} set $S' \cup S'^b$ is admissible and $S \subset (S' \cup S'^b)$, which makes it impossible for S to be preferred in ES^{sc}. Thus, we can conclude that a preferred extension in ES produces a preferred extension in ES^{sc} and vice versa. The proof for the grounded semantics exploits the relations of the grounded extension to the complete extensions as stated in Theorem 2.112. It follows the same line of reasoning as the proof for preferred semantics and the grounded semantics proof of Theorem 4.50.

What remains to be shown is the correspondence between the stable extensions of both frameworks. Let $S \subseteq A$ be a stable extension in ES. We know that $S \cup S^b$ is at least self–supporting conflict–free in ES^{sc}. Based on the previous analysis, we can see that any argument $a \in A$ e–support attacked by S will also be e–support attacked by $S \cup S^b$; similar follows for its e–supporting sets. Moreover, any argument not e–supported by A will not be e–supported by $A' = A \cup A^b$ either. Should a be not e–supported, then neither is its bypass a^b. And since any sets e–supporting a^b are constructed from the ones of a, then if every self–supporting set containing a is attacked, then so is every self–supporting set containing a^b. Thus, we can conclude that $S \cup S^b$ is a stable extension of ES^{sc}. Showing that if $S' \subseteq A'$ is stable in ES^{sc} then so is $S' \cap A$ in ES can be proved in a similar manner – from the previous proofs we know that $S' \cap A$ is at least self–supporting and conflict–free. Moreover, if a self–supporting set $B \subseteq A'$ is attacked in ES^{sc} by S', then so is $B \cap A$ by $S' \cap A$ in ES – this is due to the correspondence between self–supporting sets and the fact that the bypass arguments do not participate in the attack relation. Consequently, if every self–supporting set for an argument $a \in A$ or the argument itself is attacked by S' in ES^{sc}, then the attacks are also carried out by $S' \cap A$ in ES. Thus, S' is a stable extension of ES. $\square$

Theorem 4.56. *Let $FN = (A, R, N)$ be an AFN and $FN^{sc} = (A', R', N')$ its corresponding strongly consistent framework obtained through Translation 15. Let E^b the (possibly empty) set of bypass arguments generated by a set $E \subseteq A$ in A'. If a set of arguments E is coherent in FN, then $E \cup E^b$ is pd–acyclic in FN^{sc}. If $E' \subseteq A'$ is pd–acyclic in FN^{sc}, then $E' \cap A$ is coherent in E. $E \subseteq A$ is a σ–extension of FN, where $\sigma \in \{$conflict–free, strongly coherent, admissible, preferred, complete, grounded$\}$, iff it is a σ–extension of FN^{sc}. Every stable extension E of FN^{sc} is stable in FN but not vice versa.*

Proof. We can observe that every extension of FN^{sc} that is at least conflict–free will always be a subset of A, not only A' – every argument outside A is after all a self–attacker.

Since no attacks are introduced between the arguments in A, every conflict–free extension in FN will be trivially conflict–free in FN^{sc}. As no auxiliary arguments will show up in the conflict–free extensions and the attacks among the remaining ones are the same between FN^{sc} and FN, it is also easy to see that every conflict–free extension of FN^{sc} will be conflict–free in FN.

The proof of coherence with bypass sets is exactly the same as in the Theorem 4.50, since in these semantics only the support relation is considered and the modifications to N between the two methods for strongly consistent normal form are the same. Consequently, we refer the reader to the previous proof. What needs to be shown is that the strongly coherent sets between the two frameworks are the same. Let $E \subseteq A$ be a strongly coherent set in FN. We can observe that due to conflict–freeness, $E \cap O^E = \emptyset$, i.e. E cannot contain any inconsistency origin argument of any member of E. This means that for every argument in E there is a powerful sequence consisting of arguments that will not be replaced by their bypasses in the support they carry out towards any other arguments in E. Consequently, the same powerful sequences can still be constructed in E in FN^{sc} and thus E remains coherent. Thus, if E is strongly coherent in FN, then it is strongly coherent in FN^{sc} [31] We will now show that if $E \subseteq A'$ is a strongly coherent extension of FN^{sc}, then it is also strongly coherent in FN. Again, we can observe that E is in fact a subset of A, since all arguments in $A' \setminus A$ are self–attackers. As E is easily conflict–free in FN, what remains to be shown is coherence. Due to the fact that bypass arguments cannot show up in the extensions, a powerful sequence built for any argument $a \in E$ has to satisfy the requirements through the members of the supporting sets that were not substituted by their bypasses in these sets. Thus, it is the "original" support relations that are being used to build the sequence and thus they can be easily used in FN as well. Therefore, we can conclude that if E is strongly coherent in FN^{sc}, it is also strongly coherent in FN.

Let us now show that if an argument $a \in A$ is defended by a strongly coherent set E in FN, then it is also defended by E in FN^{sc}. Since E is strongly coherent and defends a in FN, then $E \cup \{a\}$ is strongly coherent in FN as well. Consequently, E and $E \cup \{a\}$ are strongly coherent in FN^{sc}, which accounts for the support part of defense. From the coherence proof of Theorem 4.50 we can observe that we cannot create a powerful sequence in FN^{sc} that would not be present in FN after the removal of bypass arguments – every bypass needs to be supported by its original argument. As $R \subseteq R'$ and any coherent set of an argument $a \in A$ in FN^{sc} contains some coherent set of this argument in FN, we can observe that if E attacks all coherent sets of a given argument in FN, then it also attacks all coherent sets of this argument in FN^{sc}. Thus, if an argument $a \in A$ is defended by a strongly coherent set E in FN, then it is also defended by E in FN^{sc}.

Let us now focus on the other way around; i.e. showing that if an argument $a \in A$

[31] Please note that this direction holds in the "pure" bypass strong consistency form as well. The other direction works only for those sets that are subsets of A and the proof would have to be modified to show that relevant bypasses cannot occur due to the fact that they need their origins, which acceptance breaks conflict–freeness.

is defended by a strongly coherent set $E \subseteq A$ in FN^{sc}, then it is also defended by E in FN. Please observe that it cannot be the case that E defends a bypass argument; the set is after all conflict–free and a bypass is a self–attacker. Based on previous proofs we can conclude that $E \cup \{a\}$ is strongly coherent in FN^{sc} and thus in FN, which accounts for the support part of defense. Since bypass arguments do not attack anyone but themselves and are not attacked by any other argument, from the point of view of defense of a they are not relevant. Thus, if a coherent set of an argument $b \in A$ is attacked by E in FN^{sc}, it is only attacked on the non–bypass arguments. As the attacks not related to bypasses in FN^{sc} originate in FN, this means the coherent set for b, after removing the bypass arguments, is attacked by E in FN. Hence, an argument $b \in A$ that has all of its coherent sets attacked by a strongly coherent set E in FN^{sc} also has all of its coherent sets attacked by E in FN. We can conclude that if E defends a in FN^{sc}, then it defends it in FN as well.

Based on the proofs of strong coherence, defense and the fact that bypass arguments cannot be defended by strongly coherent sets, it should be clear that admissible and complete extensions in FN and FN^{sc} are the same. From this and Theorem 2.95 it also follows that grounded and preferred extensions coincide. What remains to be explained is the relation between the stable extensions.

In the previous proofs we have observed that if a strongly coherent set of arguments $E \subseteq A$ attacked all coherent sets of a given argument $a \in A$ in FN, then it did the same in FN^{sc} and vice versa. Consequently, a strongly coherent set $E \subseteq A$ attacking all coherent sets of arguments in $A' \setminus E$ in FN^{sc} is strongly coherent in FN and attacks all coherent sets of arguments $A \setminus E$ in FN. Thus, by Lemma 2.94, a stable extension of FN^{sc} is also stable in FN. The other way around does not hold due to the fact that while a stable extension $E \subseteq A$ of FN would still attack all coherent sets of arguments in $A \setminus E$ in FN^{sc}, it would not necessarily do so for arguments in $A' \setminus A$. If E contains an argument for which a bypass was constructed, then it obviously cannot attack the bypass and thus breaches the stability requirements. $\square$

Theorem 4.58. *Let $ES = (A, R, E)$ be an EAS and $ES^{sc} = (A', R, E')$ its corresponding strongly consistent framework obtained through Translation 16. Let $S \subseteq A$, $S' \subseteq A'$ be sets of arguments and S^b the (possibly empty) set of bypass arguments generated by S in A'. S is a σ–extension of ES, where $\sigma \in \{$conflict–free, self–supporting conflict–free, admissible, preferred, complete, grounded, stable$\}$, iff it is a σ–extension of ES.*

Proof. We can observe that due to the fact that every auxiliary argument is a self–attacker, every conflict–free extension of ES^{sc} will be a subset of A, not only A'. Since no new attacks are introduced between the arguments in A and those in A^b cannot appear in extensions, clearly the conflict–free extensions between ES and ES^{sc} coincide.

The proof for self–supporting sets with bypasses is the same as in Theorem 4.54 due to the fact that the E relation is the same between in the translated frameworks in both approaches. What needs to be shown is that the self–supporting conflict–free sets between ES and ES^{sc} are the same. Let $S \subseteq A$ be a self–supporting conflict–free set in ES.

We can observe that due to conflict–freeness, $S \cap O^S = \emptyset$, i.e. S cannot contain any inconsistency origin argument of any member of S. This means that for every argument in S there is an evidential sequence consisting of arguments that will not be replaced by their bypasses in the support they carry out towards any other arguments in S. Consequently, the same evidential sequences can still be constructed in S in ES^{sc} and thus S remains self–supporting and conflict–free. Therefore, if S is self–supporting conflict–free in ES, then it is self–supporting conflict–free in ES^{sc}. We will now show that if $S \subseteq A'$ is a self–supporting conflict–free extension of ES^{sc}, then it is also self–supporting conflict–free in ES. Again, we can observe that S is in fact a subset of A, since all arguments in $A' \setminus A$ are self–attackers. As S is easily conflict–free in ES, what remains to be shown is self–support. Due to the fact that bypass arguments cannot show up in the extensions, an evidential sequence built for any argument $a \in S$ in ES^{sc} has to satisfy the requirements through the members of the supporting sets that were not substituted by their bypasses in these sets. Thus, it is the "original" support relations that are being used to build the sequence and thus they can be easily used in ES as well. Therefore, we can conclude that if S is self–supporting conflict–free in ES^{sc}, it is also self–supporting conflict–free in ES.

Let us now prove that if an argument $a \in A$ is acceptable w.r.t. a self–supporting conflict–free set S in ES, then it is also acceptable w.r.t. S in ES^{sc}. Since S is self–supporting conflict–free and a is acceptable w.r.t. S in ES, then it means that there exists a set $B \subseteq S$ s.t. BEa and that this B does not contain arguments causing inconsistency in a. Therefore, B will not be affected by the translation, and S is self–supporting conflict–free in ES^{sc} and contains $B \subseteq S$ s.t. $BE'a$. Hence, we can observe that S e–supports a. Now, from the proof on self–supporting sets (see above and proof of Theorem 4.54) we could observe that every self–supporting set for an argument $a \in A$ in ES^{sc} contains a self–supporting set for an argument $a \in A$ in ES. Therefore, since the change to the attack relation concerns only A^b arguments and they attack nothing but themselves, it holds that if S attacks all minimal e–supported attacks by a set $T \subseteq A$ against an argument $a \in A$ in ES, then it does that as well in ES^{sc}. Therefore, if an argument $a \in A$ is acceptable w.r.t. S in ES, then it is acceptable w.r.t. S in ES^{sc}.

Let us now focus on the other direction, i.e. showing that if an argument $a \in A'$ is acceptable w.r.t. a self–supporting conflict–free set S in ES^{sc}, then it is also acceptable w.r.t. S in ES. Since S is conflict–free, we can observe that a cannot be a bypass argument. In the proof on self–supporting and conflict–free sets we can observe that if S e–supports a, then due to its properties it has be the case that the e–support occurs through the supporting sets in E' that do not contain self–attackers. Consequently, they could not have been affected by the translation and represent original supporting sets in E. Thus, S e–supports a in ES. We can observe that every self–supporting set in ES^{sc} can be transformed into one in ES by removing the bypass arguments and that bypass arguments attack and are attacked only by themselves. Therefore, if S attacks any e–supported attack on a in ES^{sc}, then it does so by attacking non–bypass arguments, and it holds that S attacks any e–supported attack on a in ES. Therefore, if a is acceptable w.r.t. S in ES^{sc}, then it is also acceptable w.r.t. S in ES.

Based on the proofs for self–supporting conflict–free extensions, acceptability and the fact that the bypass arguments cannot be defended by self–supporting conflict–free sets, it should be clear that admissible and complete extensions in ES and ES^{sc} are the same. From this and Theorem 2.112 it also follows that grounded and preferred extensions coincide. What remains to be explained is the relation between the stable extensions.

In the previous proofs we have observed that if a self–supporting conflict–free set of arguments $S \subseteq A$ attacked all self–supporting sets of a given argument $a \in A$ in ES, then it did the same in ES^{sc} and vice versa. Consequently, a self–supporting conflict–free set $S \subseteq A$ attacking all self–supporting sets of arguments in $A' \setminus S$ in ES^{sc} is self–supporting conflict–free in ES and attacks all self–supporting sets of arguments $A \setminus S$ in ES. Thus, a stable extension of ES^{sc} is also stable in ES. The other way around does not hold due to the fact that while a stable extension $S \subseteq A$ of ES would still attack all self–supporting sets of arguments in $A \setminus S$ in ES^{sc}, it would not necessarily do so for arguments in $A' \setminus A$. If S contains an argument for which a bypass was constructed, then it obviously cannot attack the bypass and thus breaches the stability requirements. □

Theorem 4.18. *Let $FN = (A, R, N)$ be an AFN and $FN^{wv} = (A', R', N')$ be its weak validity form. A set $E \subseteq A$ is a σ–extension of FN, where $\sigma \in \{$coherent, strongly coherent,admissible, preferred, complete, grounded, stable$\}$, iff it is a σ–extension of FN^{wv}. If $E \subseteq A$ is a conflict–free extension of FN, then $E \cap A'$ is conflict–free in FN^{wv}. If $E \subseteq A'$ is conflict–free in FN^{wv}, then it is conflict–free in FN.*

Proof. We can observe that any at least coherent extension E will not be just a subset of A, but also of A'. Moreover, please note that if an argument $a \in A$ is coherent, then it will not be the case that any set C supporting it through N will have all of its elements removed during trimming. Should it be the case that all arguments of C are "lost", then it means those arguments were not powerful in A and thus a could not have been powerful in A either. Thus, there will be no situation s.t. a support set becomes empty.

Let $E \subseteq A$ be a conflict–free extension of FN. Clearly, any subset of E is also conflict–free in FN, including $E \cap A'$. Since $R' = R \cap (A' \times A')$, then if there were no arguments attacking each other in $E \cap A'$ in FN, then there are none in FN^{wv} as well. Thus, $E \cap A'$ is conflict–free in FN^{wv}. Let now $E \subseteq A'$ be a conflict–free extension of FN^{wv}. Again, from the fact that $R' = R \cap (A' \times A')$ we can observe that if there are no attacks in E in FN^{wv}, then there are none in E in FN. Thus, the set is conflict–free in FN.

Let $a \in A$ be an argument that possesses a powerful sequence $(a_0, ..., a_n)$ on A. We can observe that a still appears in A'. Moreover, it also holds that for every argument a_i in the sequence, $(a_0, ..., a_i)$ is a powerful sequence for a_i on A. Thus, all of the arguments $a_0, ..., a_n$ appear in A'. From the definition of the trimmed subframework it holds that $N' = \{(C', a) \mid a \in A', C' \neq \emptyset, \exists(C, a) \in N \wedge C' = C \cap A'\}$. Moreover, from the previous explanation it holds that for an argument $a \in A'$, there is no set $C \subseteq A$ s.t. $(C, a) \in N$ for which $C \cap A' = \emptyset$. We can now observe that 1. if there was no $C \subseteq A$ supporting a_0 through N in FN, then there is no $C' \subseteq A'$ supporting a_0 through N' in

FN^{wv}; and 2. if $C \cap \{a_0, ..., a_{i-1}\} \neq \emptyset$ then $(C \cap A') \cap \{a_0, ..., a_{i-1}\} \neq \emptyset$ for any supporting set C of an argument a_i in the sequence. Consequently, we can observe that $(a_0, ..., a_n)$ is a powerful sequence on A' for a in FN^{wv}.

Let now $a \in A'$ be an argument possessing a powerful sequence $(a_0, ..., a_n)$ on A'. From the way N' is constructed we can observe that for any $C' \subseteq A'$ supporting an argument a_i through N' in FN^{wv} there is a set $C \subseteq A$ s.t. $C' \subseteq C$ which supports a through N in FN and vice versa. Moreover, it also holds that if an argument a_i was not supported at all through N' in FN, then it is not supported through N in FN. Thus, $(a_0, ..., a_n)$ is a powerful sequence for $a \in A'$ in FN.

From the sequence analysis it easily follows that a set of arguments $E \subseteq A$ is coherent in FN iff it is coherent in FN^{wv}. Similar holds for strong coherence. We now need to show that E defends an argument $a \in A$ in FN iff it defends it in FN^{wv}. We can observe that it cannot be the case that a is not in A' due to the fact that if E defends a in FN, then $E \cup \{a\}$ is coherent in FN. From the relation between the coherent extensions of FN and FN^{wv} it follows that $E \cup \{a\}$ is coherent in FN iff it is coherent in FN^{wv}. Consequently, now we only need to focus on the attack part of defense.

Let us assume that a is defended by E in FN, but not in FN^{wv}. This means there exists a coherent set $C \subseteq A'$ attacking a in FN^{wv} that is not attacked by E in FN^{wv}. This means that C is also coherent in FN and as $R' \subseteq R$, C attacks a in FN. Since both C and E are coherent, the elements they contain are powerful and thus the attack between them in R are also in R'. Consequently, if E did not attack C in FN^{wv}, then it could not have attacked it in FN either. We reach a contradiction. Therefore, if $a \in A$ is defended by E in FN, then it is defended in FN^{wv}.

Let us now assume that a is defended by E in FN^{wv}, but not in FN. Since the coherent sets match and $R' \subseteq R$, it can only be the case that there is an attack in $R \setminus R'$ carried out by a member of coherent set $C \subseteq A$ on a. However, since C is coherent and a is powerful in A, then $C \subseteq A'$ and $a \in A'$. Therefore, due to the way R' is created, it cannot be the case that this attack from a member of C on a is not in FN^{wv}. Thus, we reach a contradiction, and can conclude that if a is defended by E in FN^{wv}, then it is defended by E in FN.

Using the previous parts of the proof we can show that admissible, complete and preferred extensions coincide between the two frameworks. Since the grounded extension is the least complete one by Theorem 2.95, we can conclude that grounded extension is the same in FN and FN^{wv}. What remains to be analyzed is stability.

Let $E \subseteq A$ be a stable extension in FN. We know it is complete in FN^{wv}; let us however assume it is not stable. This means there exists an argument $a \in A' \setminus E$ s.t. a is not attacked by E through R' and for all $B \subseteq A'$ s.t. $BN'a$, $E \cap B \neq \emptyset$ in FN^{wv}. Since E is coherent in FN^{wv}, then it is coherent in FN. Since $a \in A'$, then we can observe that if there is no member of E attacking a in FN^{wv}, then there is no member of E attacking a in FN. If E has an element in common with every supporting set of a in FN^{wv}, then by construction it means that for all $B' \subseteq A$ s.t. $B'Na$, $E \cap B' \neq \emptyset$. Consequently, a is not attacked by E in FN and is sufficiently supported by E in FN as well. We reach a contradiction with the stability of E. Therefore, we can conclude that E is stable in FN^{wv}.

Let now $E \subseteq A'$ be stable in FN^{wv}, but not in FN. Again, we know E is complete in FN. Thus, it has to be the case that the deactivated set E^+ is not equal to $A \setminus E$ in FN. If $a \in A$ does not have a powerful sequence in A, then it naturally cannot be the case that E sufficiently supports a and a has to be in E^+. Moreover, it will not appear in A'. Thus, it can only be the case that there exists a coherent argument $a' \in A \setminus E$ that is not in E^+ in FN, but is in E^+ in FN^{wv}. However, it is easy to see that if E does not sufficiently support a' in FN^{wv}, then neither it does in FN. Moreover, since $R' \subseteq R$, it cannot be the case that it is attacked by E in FN^{wv} but not in FN. We reach a contradiction. Hence, we can conclude that a set is stable in FN iff it is stable in FN^{wv}. □

Theorem 4.20. *Let $ES = (A, R, E)$ be an EAS and $ES^{wv} = (A', R', E')$ be its weak validity form. A set $S \subseteq A$ is a σ–extension of ES, where $\sigma \in \{$self-supporting, strongly self–supporting, admissible, preferred, complete, grounded, stable$\}$, iff it is a σ–extension of ES^{wv}. If $S \subseteq A$ is a conflict–free extension of ES, then $S \cap A'$ is conflict–free in ES^{wv}. If $S \subseteq A'$ is conflict–free in ES^{wv}, then it is conflict–free in ES.*

Proof. We can observe that every extension S that is self–supporting in ES will be a subset of A', not just A. Any argument contained in a self–supporting set possesses and evidential sequence and thus will not be removed from the framework. Moreover, please note that if an argument $a \in A$ is e–supported by A, then it will retain at least one C supporting it through E in E'. Should it be the case that all C's are "lost", then it meant those arguments were not e–supported by A and thus a could not have been e–supported by A either. Finally, since η has a trivial evidential sequence (η), it will clearly be included in A'.

Let $S \subseteq A$ be a conflict–free extension of ES. Clearly, if S is conflict–free, then so is any $S' \subseteq S$. Therefore, $S \cap A'$ is conflict–free in ES. Since $R' \subseteq R$ and there is no conflict in $S \cap A'$ in ES, then there will be none in ES^{wv} and $S \cap A'$ is conflict–free in ES^{wv}.

Let $S \subseteq A'$ be a conflict–free extension of ES^{wv}. From the fact that $R' = R \cap ((2^{A'} \setminus \emptyset) \times A')$ we can observe that no attack from a subset of A' against an argument of A' is deleted from R. Thus, if there is no conflict in S in ES^{wv}, then it means there is no conflict in S in ES. Thus, S is conflict–free in ES.

Let $S \subseteq A$ be a self–supporting set of ES. This means that every argument has at least one evidential sequence on A (see Theorem 2.99). Consequently, every argument in S will be present in A'. Since $E' = E \cap ((2^{A'} \setminus \emptyset) \times A')$, we can see that no support relations between S and any of its members will be removed. Consequently, S is also self–supporting in ES^{wv}. From the fact that $E' \subseteq E$ it follows easily that any self–supporting set in ES^{wv} will also be self–supporting in ES. From the relation between self–supporting and conflict–free sets in ES and ES^{wv} we can also conclude that S is self–supporting conflict–free in ES iff it is such in ES^{wv}.

Let S be a self–supporting set. Assume that $a \in A$ is acceptable w.r.t. S in ES, but not in ES^{wv}. Since S e–supports a, a has an evidential sequence on A and thus will appear in

A'. Moreover, it will also be e–supported by S in ES^{wv}. Thus, it can only be the case that there is some minimal e–supported attack by a set $T \subseteq A'$ against a and no member of T is attacked by S in ES^{wv} (since S is self–supporting, every attack carried out by it will be e–supported). However, since T is trivially self–supporting, then any attacks from S to members of T will appear in R'. Thus, it cannot be the case that a is not acceptable w.r.t. S in ES^{wv} if it is in ES.

Now assume that $a \in A$ is acceptable w.r.t. S in ES^{wv}, but not in ES. By the analysis above it follows that a is e–supported by S in ES. Thus, it has to be the case that there is some minimal e–supported attack $T \subseteq A$ against a which is not attacked by S. However, it is easy to see that T is also a minimal e–supported attack against a in ES^{wv}, and since $R' \subseteq R$. it has to be the case that S attacks T in ES as well. Thus, we can conclude that an argument $a \in A$ is acceptable w.r.t. a self–supporting set S in ES iff it is acceptable w.r.t. S in ES^{wv}.

The fact that admissible, preferred and complete extensions coincide follows simply from the above. As the grounded extension is the least complete one by Theorem 2.112, we can conclude that grounded extensions of both frameworks coincide as well. Let us now focus on stable extensions and let $S \subseteq A$ be a self–supporting conflict–free set in ES. Clearly, it is also self–supporting conflict–free in ES^{wv}. From stability it follows that S attacks at least one element in every self–supporting set containing an argument $a \notin S$ in ES. Thus, based on the previously done analysis it holds that S attacks at least one element in every self–supporting set containing an argument $a \notin S$ in ES^{wv}. Thus, S is stable in ES^{wv}. Let us focus on the other way around now and let $S \subseteq A'$ be stable in ES^{wv}. Every self–supporting set in ES^{wv} is also self–supporting in ES. Moreover, if it is attacked by S in ES^{wv} then it is attacked by S in ES. Thus, S is self–supporting conflict–free in ES and attacks every self–supporting set containing an argument $a \in A' \setminus S$ in ES. Since arguments in $A \setminus A'$ do not appear in self–supporting sets, this property extends to $a \in A \setminus S$ and thus S meets the stability requirements in ES. □

Theorem 4.22. *Let $D = (A, L, C)$ be an ADF and $D^{wv} = (A', L', C')$ be its weak validity form. A set $E \subseteq A$ is a σ–extension of D, where $\sigma \in$ {pd–acyclic conflict–free, aa–admissible, aa–preferred, aa–complete, acyclic grounded, stable}, iff it is a σ–extension of D^{wv}.*

Proof. Please note that an argument that does not possess an acyclic pd–evaluation will not appear in any acyclic extension. Consequently, every extension E of D is a subset of A'. The proofs that the weak validity form preserves the listed semantics will be similar to the ones in the case of cleansed form (see Theorem 4.15).

We will start with the analysis of decisiveness and acyclic pd–evaluations. Let $a \in A$ be an argument that possesses an acyclic pd–evaluation (F, B) on A in D. Clearly, $F \subseteq A'$. Let v_a be the minimal decisively in interpretation for a used in the construction of (F, B). We will show that v_a and its limitation to A' are decisively in interpretations for a in D^{wv}, though not necessarily minimal ones. We can observe that $v_a^{\mathbf{t}} \subseteq A'$ and $v_a^{\mathbf{t}} \subseteq par(a)$ in D. Consequently, $v_a^{\mathbf{t}} \subseteq par(a)$ in D^{wv} as well. From this, the Definition

4.3 of the reduct and the fact that $C_a(v_a^{\mathbf{t}}) = in$ in D, it holds that $C'_a(v_a^{\mathbf{t}}) = in$ in D^{wv}. Assume v_a is not decisively in for a in D^{wv}; this means there is a set of arguments E where $v_a^{\mathbf{t}} \subseteq E \subseteq A' \setminus v_a^{\mathbf{f}}$ s.t. $C'_a(E \cap par(a)) = out$. Since $E \subseteq A'$, then $E \cap par(a)$ in D is the same as $E \cap par(a)$ in D^{wv}. Consequently, from the definition of reduct it holds that $C'_a(E \cap par(a)) = C_a(E \cap par(a))$. From the properties of E and the definition of decisiveness it now follows that v_a could not have been decisively in for a in D and we reach a contradiction. Consequently, if v_a is decisively in for a in D and is used in the construction of an acyclic pd–evaluation for a in A in D, then v_a (or more specifically, its subinterpretation limited to A') is decisively in for a in D^{wv}. Depending on the source framework, it can happen that the condition of a is reduced to an equivalent of a tautology (even if some parents show up as redundant ones). Thus, neither v_a nor its limitation to A' have to be minimally decisively in for a in D^{wv}. Nevertheless, a minimal interpretation can be "extracted", and we can thus conclude that a will have an acyclic pd–evaluation (F, B') on A' in D^{wv}, where $B' \subseteq B \cap A'$ depends on the "extraction".

Let now $a \in A'$ be an argument that possesses an acyclic pd–evaluation (F, B) on A in D^{wv} and let v_a be the minimal decisively in interpretation for a used in the construction of (F, B). Assume v_a is defined for a set $E \subseteq A'$; let now z_a be the $\mathbf{f}$–completion of v_a to $E \cup (A \setminus A')$. We will show that z_a is decisively in for a in D. From the definition of the reduct and the fact that v_a and z_a have the same $\mathbf{t}$ mappings it follows that if $C'_a(v_a^{\mathbf{t}}) = in$ in D^{wv}, then $C_a(z_a^{\mathbf{t}}) = in$ in D^{wv}. Now assume that there is some set of arguments E where $z_a^{\mathbf{t}} \subseteq E \subseteq A \setminus z_a^{\mathbf{f}}$ s.t. $C_a(E \cap par(a)) = out$. From the construction of z_a it means that $v_a^{\mathbf{t}} \subseteq E \subseteq A \setminus v_a^{\mathbf{f}}$. Again, from the definition of reduct, if such a set outs the condition of a in D, then it does so in D^{wv} as well and thus we contradict the decisiveness of v_a. Therefore, z_a is decisively in for a in D, even if not minimally. Nevertheless, a minimal interpretation can be "extracted", and we can conclude that a will have an acyclic pd–evaluation (F, B') on A in D, where $B \subseteq B' \subseteq (B \cup A \setminus A')$ depends on the "extraction". Please note that there can be more than one D evaluation "containing" (F, B). In particular, we did not consider extending v_a with $\mathbf{t}$ mappings to account for previously required parents becoming redundant. Consider a framework $(\{a, b\}, \{C_a = a, C_b = b \vee \neg a\})$. In this case b has two minimal decisively in interpretations, $v_1 = \{b : \mathbf{t}\}$ and $v_2 = \{a : \mathbf{f}\}$. The weak validity form of the frameworks is $(\{b\}, \{C_b = b \vee \top\})$. In this case b has a condition equivalent to $\top$ and its only minimal decisively in interpretation is just empty. It is contained both in v_1 and v_2 but we have described a method allowing to come back to v_2 only.

Let $E \subseteq A$ be a pd–acyclic conflict–free extension of D. Clearly, $E \subseteq A'$ as well. Since every argument $a \in E$ has an unblocked acyclic pd–evaluation (F, B) on E in D, then by the proof above it also has an acyclic pd–evaluation (F, B') s.t. $B' \subseteq B$ on E in D^{wv}. Since $B \cap E = \emptyset$, then $B' \cap E = \emptyset$ as well and the evaluation is unblocked. Consequently, E is pd–acyclic conflict–free in D^{wv}.

Let now $E \subseteq A'$ be a pd–acyclic conflict–free extension of D^{wv}. Since every argument $a \in E$ has an unblocked acyclic pd–evaluation (F, B) on E in D^{wv}, then by the proof above it also has an acyclic pd–evaluation (F, B') where $B \subseteq B' \subseteq (B \cup A \setminus A')$ on E in

D. It is easy to see that $(B \cup A \setminus A') \cap E = \emptyset$. Therefore, (F, B') will be not be blocked by E and E is pd–acyclic conflict–free in D as well.

Let E be a pd–acyclic conflict–free extension of D^{wv} and D. Let E^{a+}_{wv} and E^{a+} and be its acyclic discarded sets in both frameworks. We will show that $E^{a+} = E^{a+}_{wv} \cup (A \setminus A')$. An argument $a \in (A \setminus A')$ by construction does not posses an acyclic pd–evaluation and thus is trivially in E^{a+}. Let now $a \in E^{a+}_{wv}$ be an argument in the discarded set in the weakly valid ADF; from the proofs above we could have observed that every acyclic pd–evaluation (F, B) in D^{wv} can be "extended" through the blocking set to an acyclic pd–evaluation for a in D and that every acyclic pd–evaluation (F, B') for a in D could have been "trimmed" through the blocking set to an evaluation for a in D^{wv}. Since a is in the acyclic discarded set, then by Lemma 2.128 all of its acyclic pd–evaluations are blocked through the blocking set by E in D^{wv}, and by the explanation above it has to be the case that all acyclic pd–evaluations for a in D are blocked through the blocking set by E. Consequently, $a \in E^{a+}$. Let now $a \in E^{a+} \cap A'$. Assume that a is not in E^{a+}_{wv}. By Lemma 2.128 it means it possesses an acyclic pd–evaluation (F, B) on A' s.t. $B \cap E = \emptyset$. By the proofs above, this evaluation can be extended to an acyclic pd–evaluation (F, B') on A in D s.t. $B \subseteq B' \subseteq (B \cup A \setminus A')$. As $E \subseteq A'$ and $E \cap B = \emptyset$, then $B' \cap E = \emptyset$ and (F, B') cannot be blocked by E in D. Consequently, a could not have been in E^{a+} in the first place and we reach a contradiction. This brings us to the conclusion that $E^{a+} = E^{a+}_{wv} \cup (A \setminus A')$.

Let us move on to admissibility. Let E be an aa–admissible extension of D. Assume it is not aa–admissible in D^{wv}; since it is already pd–acyclic conflict–free, it means that there is an argument $a \in E$ s.t. none of its acyclic pd–evaluations (F, B) on E in D^{wv} has a blocking set contained in the acyclic discarded set. However, based on the presented relations between evaluations and the acyclic discarded sets in D and D^{wv}, we can observe that if no evaluation for a meets admissibility requirements in D^{wv}, then no evaluation can meet the admissibility requirements in D either. Thus, E could not have been aa–admissible in D in the first place and we reach a contradiction. Therefore, if E is aa–admissible in D, then it is aa–admissible in D^{wv}.

Let now E be aa–admissible in D^{wv}. From the previous parts of this proof we could observe that any acyclic pd–evaluation for $a \in E$ in D^{wv} can be extended into one in D by adding some of the $A \setminus A'$ elements to the blocking set. Since these elements are disjoint from E and trivially in E^{a+}, we can observe that if an acyclic pd–evaluation for a satisfies admissibility requirements in D^{wv}, then so does its "extension". Consequently, E is aa–admissible in D^{wv}.

We have shown that the aa–admissible extensions between our two frameworks coincide. We now need to show that it also holds for aa–complete sematics. Let E be an aa–complete extension of D. If it is not aa–complete in D^{wv}, it means that there is some argument $a \in A' \setminus E$ that is decisively in w.r.t. the acyclic range interpretation v^a_E of E in D^{wv}. Consequently, a minimal version of this interpretation can be used in constructing an acyclic pd–evaluation for a. However, from the proof on evaluations we can observe that extending the interpretation with **f** mappings for the set $A \setminus A'$ will create an interpretation that is decisively in for a in D. Since this interpretation will be contained in the acyclic range of E in D, then E could not have been aa–complete in D in the first place and we reach a contradiction. Consequently, E has to be aa–complete in D^{wv}.

Let now E be aa–complete in D^{wv}. Assume it is not complete in D; this means there is an argument $a \in A \setminus E$ that is decisively in w.r.t. acyclic range interpretation v_E^a of E in D. An argument not possessing an acyclic pd–evaluation will be decisively out w.r.t. v_E^a (see Lemma 2.128 and Proposition 2.150). Therefore, only $a \in A' \setminus E$ are the possible candidates. Since a is decisively in w.r.t. the range interpretation, then as observed in the evaluation part of the proof, we can remove the $A \setminus A'$ assignments from this interpretation and obtain one that is decisively in for a and equal to the acyclic range of E in D^{wv}. Consequently, a would have been decisively in w.r.t. the acyclic range of E of D^{wv} and we reach a contradiction with the completeness of E. Therefore, if E is aa–complete in D^{wv}, then it is complete in D.

We now move to stable semantics. Let $E \subseteq A$ be a stable extension of D; by Lemma 2.139 it is pd–acyclic conflict–free in D and $E^{a+} = A \setminus E$. By the previous parts of the proof, E is pd–acyclic conflict–free in D^{wv} and $E^{a+}_{wv} = E^{a+} \cap A'$. Thus, as $E \subseteq A'$, $E^{a+}_{wv} = A' \setminus E$ and by Lemma 2.139 it holds that E is a stable extension of D^{wv}. The proof that every stable extension of D^{wv} is stable in D follows similarly.

The fact that $E \subseteq A'$ is a preferred extension of D iff it is one in D^{wv} follows straightforwardly from the relation between admissible extensions. The coincidence of the acyclic grounded extensions between the two frameworks is a result of the relation between the complete extensions of D and D^{wv} and Theorem 2.158. □

Theorem 4.23. *Let $D = (A, L, C)$ be an ADF. If D is weakly valid, then it is also cleansed, but not vice versa.*

Proof. If D is weakly valid, then every argument possesses an acyclic pd–evaluation. Therefore, every argument also possesses a standard one, and D is cleansed. Now, if D is cleansed, then every argument has a standard evaluation. However, not every standard evaluation can be made into an acyclic one. For example, the framework $(\{a\}, \{C_a = a\})$ is in cleansed, but not in weakly valid form. □

Theorem 4.25. *Let $FN^{wv} = (A', R', N')$ be a weakly valid AFN and $FN^{rv} = (A', R', N'')$ its relation valid form. A set $E \subseteq A'$ is a σ–extension of FN^{wv}, where $\sigma \in$ $\{$conflict–free, coherent, strongly coherent, admissible, preferred, complete, grounded, stable$\}$ iff it is a σ–extension of FN^{rv}.*

Proof. We can observe that the sets of arguments and attacks in FN^{wv} and FN^{rv} are the same. Consequently, the conflict–free extensions between the two frameworks coincide.

Let us move on to coherence. Since we are dealing with a weakly valid AFN, then every argument $a \in A'$ possesses a powerful sequence on A'. We can remove a from this sequence to obtain either another powerful sequence for an argument directly before a, or an empty sequence if a is in fact the starting argument. The first leads to a coherent set on $A' \setminus \{a\}$, while the other produces an empty set, which as such is also coherent. Therefore, for every $a \in A'$ we can find a coherent set $C \subseteq A' \setminus \{a\}$, s.t. $\forall B \subseteq A$, if BNa then $C \cap B \neq \emptyset$. This means that for every set $B \subseteq A'$ s.t. BNa, $B \cap \bigcup coh(A' \setminus \{a\}) \neq \emptyset$. Consequently, it cannot be the case that an argument $a \in A'$ is supported by a set of arguments C in N', but there is no (nonempty) $C' \subseteq C$ supporting it in N''.

Let $E \subseteq A'$ be a coherent set of FN^{wv}. We will show that it is also coherent in FN^{rv}. Since E is coherent in FN^{wv}, then every argument $s \in E$ has a powerful sequence on E. Assume $(a_0, ..., a_n)$ is such a sequence for s. By the powerful conditions, a_0 requires no support through N' and thus will require no support through N'' either. Consequently, (a_0) is a powerful sequence both in FN^{wv} and FN^{rv}. Let us now focus on a_1. If it requires no support through N', then the analysis for a_0 can be repeated. Otherwise, we know that $\forall C \subseteq A'$ s.t. $CN'a$, $a_0 \in C$ and that $a_0 \in coh(A' \setminus \{a_1\})$. Thus, by the construction of FN^{rv} it is easy to see that $\forall C' \subseteq A'$ s.t. $C'N''a$, $a_0 \in C'$. From this follows that (a_0, a_1) is a valid powerful sequence for a_1 on E both in FN^{wv} and FN^{rv}. Let us now focus on a_2. Again, if no support is required, we repeat the analysis for a_0. Otherwise, we can observe that every set supporting a_2 in N' has an element in common with $\{a_0, a_1\}$. Since both a_0 and a_1 have powerful sequence on $A' \setminus \{a_2\}$, then every set supporting a_2 in N'' will have an element in common with $\{a_0, a_1\}$. We can continue like this until we reach $a_n = s$ and the conclusion that s has a powerful sequence on E. Therefore, every argument that is powerful in E in FN^{wv} will also be powerful in E in FN^{rv} and by the use of the same sequences. Thus, E is coherent in FN^{rv}.

Let $E \subseteq A'$ be now a coherent set of FN^{rv}. From the construction of N'' it follows that for every set $C \subseteq A'$ s.t. $CN's$ in FN^{wv} for an argument $s \in A'$, there exists a nonempty $C' \subseteq C$ s.t. $CN''s$ in FN^{rv}. Moreover, for every $B \subseteq A'$ s.t. $BN''s$ in FN^{rv} there is a set $B' \subseteq A'$ s.t. $B \subseteq B'$ and $B'N's$ in FN^{wv}. Consequently, if there exists a powerful sequence for s on E satisfying the support requirement through N'', then naturally the same sequence will satisfy it through N'. Therefore, E is a coherent set of FN^{rv}.

Since the coherent sets between the frameworks coincide and the attack relation remains unchanged, it holds that strongly coherent sets coincide as well. Moreover, we can also observe that an argument is defended by a coherent set E in FN^{wv} iff it is defended in FN^{rv}. Based on this we can show that the admissible, preferred, complete and stable extensions coincide. Moreover, by Theorem 2.95, also the grounded extensions are the same in both frameworks. $\square$

Theorem 4.28. *Let $ES = (A, R, E)$ be an EAS and $ES^{rv} = (A, R, E')$ its relation valid form. A set $S \subseteq A$ is a σ–extension of ES, where $\sigma \in \{$conflict–free, self-supporting, strongly self-supporting, admissible, preferred, complete, grounded, stable$\}$ iff it is a σ–extension of ES^{rv}.*

Proof.

The fact that the conflict–free sets coincide between ES and ES^{rv} follows simply from the fact that the set of arguments and the attack relation remain unchanged.

Let $S \subseteq A$ be a self–supporting set of ES. We will show that it is also self–supporting in ES^{rv}. Let s be an argument in S and $(a_0, ..., a_n)$ the evidential sequence for s on S.

It is easy to see that as $a_0 = \eta$, then a_0 is trivially e–supported by any set in ES^{rv} and $\{\eta\}$ is self–supporting. Let us now focus on a_1. By the evidential condition we know that $\{a_0\}Ea_1$. Thus, by the construction of E' it follows that $\{a_0\}E'a_1$. We can now move to a_2. From the requirements of the sequence it follows that there is a nonempty set $B \subseteq \{a_0, a_1\}$ s.t. BEa_2. Again, it will hold that $BE'a_2$. We can now continue in this manner with further elements in the sequence and repeat it for arguments in S, showing that every evidential sequence in ES carries over to ES^{rv}. Thus, S has to be self–supporting in ES^{rv}. The fact that every self–supporting set in ES^{rv} will also be self–supporting in ES is easy to prove since $E' \subseteq E$.

Since all of the evidential sequences and attacks are preserved in both of the frameworks, it is easy to see that also the e–supported attacks will be the same in ES and ES^{rv}. Consequently, any argument $a \in A$ acceptable w.r.t. a self–supporting set S in ES will also be acceptable in ES^{rv} and vice versa. Thus, it is easy to see that strongly self–supporting, admissible, preferred, complete grounded and stable extensions will coincide between the two frameworks. □

Theorem 4.31. *Let $FN = (A, R, N)$ be an AFN. If FN is strongly valid, then it is weakly and relation valid.*

Proof. Let FN be strongly valid. We can create a powerful sequence $(a_0, ..., a_n)$ containing all arguments in A. Clearly, $(a_0, .., a_{i-1})$ is a powerful sequence for a_{i-1}. Thus, every argument $a \in A$ will have such a sequence and FN is weakly valid. From the definition of $suf(a)$ and the fact that every set in it can be used in the construction of a powerful sequence for a it follows that FN is relation valid. □

Theorem 4.32. *Let $FN = (A, R, N)$ be a strongly valid AFN. A set of arguments $E \subseteq A$ is coherent iff for every argument $a \in E$ and set $C \subseteq A$ s.t. CNa, $C \cap E \neq \emptyset$.*

Proof. If a set $E \subseteq A$ is coherent, then every argument in it has a powerful sequence, and from the properties of the sequence we can easily observe that for every argument $a \in E$ and set $C \subseteq A$ s.t. CNa, $C \cap E \neq \emptyset$. Therefore, it suffices to focus on the other direction.

Let $E \subseteq A$ be a set of arguments s.t. every argument in this set is sufficiently supported by it. However, assume it is not coherent; this means there is an argument $a \in E$ that does not have a powerful sequence on E. Clearly, E cannot be ordered into a powerful sequence. For every argument $b \in E$, there is a subset $E_b \subseteq E$ s.t. $E_b \in suf(b)$, where $suf(b)$ is defined as in Definition 4.30. The function we obtain by assigning every $b \in E$ the set E_b cannot be ordered into a powerful sequence. Therefore, the function obtained by adding the assignments for arguments in $A \setminus E$ cannot be ordered into a powerful sequence either. Thus, FN cannot be strongly valid, and we reach a contradiction. □

Theorem 4.33. *Let $FN = (A, R, N)$ be an AFN. Let $sup(a) = \bigcup_{C \subseteq A, CNa} C$ denote all arguments supporting a and $suf(a) = \{S \mid S \subseteq sup(a)$ and $\forall C \subseteq$ s.t. CNa, $C \cap S \neq \emptyset\}$ stand for all subsets of $sup(a)$ that have an element in common with every support set of* *a. FN is strongly valid iff there exists a sequence $(a_0, ..., a_n)$ of all arguments in A s.t. given any function $f : A \to \{S \mid a \in A, S \in suf(a)\}$, $(a_0, ..., a_n)$ is a powerful sequence s.t. $f(a_0) = \emptyset$ and $f(a_i) \subseteq \{a_0, ..., a_{i-1}\}$ for $i > 0$.*

Proof. Let us start with showing that if FN is strongly valid, then a suitable sequence exists.

First of all, we can observe that if FN is strongly valid, then it is weakly valid too. Therefore, there exists an argument $a \in A$ that does not require support through N at all. We can collect such arguments into a set $A_0 \subseteq A$. Every support function f will assign $\emptyset$ to the elements in A_0. Any ordering of them will be a powerful sequence in FN on its own. If $A_0 = A$, then our proof is done. Let us thus assume it is not the case.

We can now find an argument $b \in A \setminus A_0$ s.t. for every set $S \in suf(b)$, $S \subseteq A_0$. If it were not the case, then for every argument $c \in A \setminus A_0$ we could find a set $S_c \in suf(c)$ s.t. S_c is not a subset of A_0. Again, we can use these sets to construct a support function that cannot produce a powerful sequence on A. We can collect all such arguments into a set $A_1 \subseteq A$. We add the arguments in A_1 in arbitrary order to the sequence created for A_0.

We can continue this line of reasoning until we go through all the arguments and obtain our sequence. Based on the construction, we can observe that this sequence will be powerful w.r.t. any support function f on A.

If there is a sequence s.t. independently of the support function, it is powerful, then clearly the strong validity restrictions are satisfied. $\square$

Theorem 4.34. *Let $FN = (A, R, N)$ be an AFN and $SG^{FN} = (A, N')$, where $N' = \{(a, b) \mid \exists E \subseteq A, a \in E \text{ s.t. } ENb\}$, the support graph induced by FN. FN is strongly valid iff SG^{FN} is a directed acyclic graph.*

Proof. If FN is strongly valid, then by Theorem 4.33 there exists a sequence seq of arguments s.t. independently of the created support function, it is a powerful sequence. From this and the definition of the powerful sequence, this means every two arguments a and b s.t. $\exists S \subseteq A$, $a \in S$ and SNb, a precedes b in seq. Therefore, it is easy to see that seq is a topological ordering of the support graph SG^{FN}. Hence, SG^{FN} is a directed acyclic graph.

If SG^{FN} is a directed acyclic graph, then there exists a topological ordering seq of its nodes. Based on the construction of SG^{FN}, this means that for every two arguments a and b s.t. $\exists S \subseteq A$, $a \in S$ and SNb, a precedes b in seq. Consequently, independently of the chosen support function f as in Definition 4.30, this sequence seq will be a powerful sequence of FN covering all of its arguments. Hence, by Theorem 4.33, FN is strongly valid. $\square$

Theorem 4.36. *Let $ES = (A, R, E)$ be an EAS. If ES is strongly valid, then it is weakly and relation valid.*

Proof. Follows similarly as in the proof of Theorem 4.31. $\square$

Theorem 4.37. *Let $ES = (A, R, E)$ be a strongly valid EAS. A set of arguments $S \subseteq A$ is self–supporting iff for every argument $a \in S$ there is a set $S' \subseteq S$ s.t. $S'Ea$.*

Proof. If a set $S \subseteq A$ is self–supporting, then every argument in S has an evidential sequence on S. Thus, from the properties of the sequence we can easily observe that for every argument $a \in S$, there exists $C \subseteq A$ s.t. CEa, $C \subseteq \emptyset$. Therefore, it suffices to focus on the other direction.

Let $S \subseteq A$ be a set of arguments s.t. every argument in this set is sufficiently supported by it. However, assume it is not self–supporting; this means there is an argument $a \in E$ that does not have an evidential sequence on E. This means that E cannot be ordered into an evidential sequence either. Nevertheless, for every $b \in E$, there is a subset $E_b \subseteq E$ s.t. E_bEb. The function we obtain by assigning every $b \in E$ the set E_b cannot be ordered into an evidential sequence. Therefore, a function obtained by adding the assignments for arguments in $A \setminus E$ cannot be ordered into an evidential sequence either. Thus, ES does not meet the requirements of Definition 4.35and cannot be strongly valid. We reach a contradiction with our assumptions. Hence, it every argument in a set is sufficiently supported by the set, then the set is self–supporting. □

Theorem 4.38. *Let $ES = (A, R, N)$ be an EAS. ES is strongly valid iff there exists a sequence $(a_0, ..., a_n)$ of all arguments in A s.t. given any function $f : A \to \{S \mid a \in A, S \subseteq A, SEa\}$, $(a_0, ..., a_n)$ is an evidential sequence s.t. $f(a_0) = \emptyset$ and for $i > 0$ and $f(a_i) \subseteq \{a_0, ..., a_{i-1}\}$.*

Proof. Let us show that if ES is strongly valid, then a suitable sequence exists.

We start our sequence with η. If $A = \{\eta\}$, then we are done. Let us therefore assume it is not the case. We can observe that if ES is strongly valid, then it is weakly valid as well, and for every non–η argument there exists a set supporting it through E. We can now find an argument $b \in A \setminus \{\eta\}$ s.t. b is supported only by $\{\eta\}$. If it were not the case, then for every argument $c \in A \setminus \{\eta\}$ we could find a set S_c s.t. S_cEc and S_c is not a subset of $\{\eta\}$. We can observe that the support function constructed with this assignment cannot possibly produce an evidential sequence on all arguments in A. We can collect all such arguments into a set $A_1 \subseteq A$. We add the arguments in A_1 in arbitrary order to the (η) sequence.

If $A = \{\eta\} \cup A_1$, then we are done. Let us therefore assume it is not the case. We can now find an argument $c \in A \setminus (\{\eta\} \cup A_1)$ s.t. for every set $S_c \subseteq A$ supporting c through E, $S_c \subseteq \{\eta\} \cup A_1$ Again, we can observe that if it were not the case, we could collect the sets not meeting these requirements and extract a support function from them that could not produce an evidential sequence. We can collect all such arguments into a set $A_2 \subseteq A$ and add them in arbitrary order to sequence consisting of η and arguments in A_1.

We can continue this line of reasoning until we go through all the arguments and obtain our sequence. Based on the construction, we can observe that this sequence will be evidential w.r.t. any support function f on A.

If there is a sequence s.t. independently of the support function, it is evidential, then clearly the strong validity restrictions are satisfied. □

Theorem 4.39. *Let $ES = (A, R, E)$ be an EAS s.t. $A \neq \emptyset$ and $SG^{ES} = (A, E')$, where $E' = \{(a, b) \mid \exists X \subseteq A, a \in X$ s.t. $XEb\}$, the support graph induced by ES. ES is strongly valid iff SG^{ES} is a rooted directed acyclic graph s.t. η is the root.*

Proof. If ES is strongly valid, then by Theorem 4.38 there exists a sequence seq of arguments s.t. independently of the created support function, it is an evidential sequence. From this and the definition of an evidential sequence, this means every two arguments a and b s.t. $\exists S \subseteq A$, $a \in S$ and SEb, a precedes b in seq. Therefore, it is easy to see that seq is a topological ordering of the support graph SG^{ES}. Hence, SG^{ES} is a directed acyclic graph. However, also by the definition of an evidential sequence, we can observe that for every non–η argument, at least one supporting set S has to exist. Moreover, since seq is always an evidential sequence, then there must exist a directed (support) path from η to any other argument in ES and therefore in SF^{ES} as well. Consequently, SG^{ES} is in fact a rooted directed acyclic graph with η as the root.

If SG^{FN} is a directed acyclic graph rooted at η, then there exists a topological ordering seq of its nodes and there is a path from the node representing evidence to every other node. This also means that η is the only node in SG^{ES} without any incoming edges and it will always be at the start of seq. Based on the construction of SG^{ES}, this means that for every two arguments a and b s.t. $\exists S \subseteq A$, $a \in S$ and SEb, a precedes b in seq and for non–η argument, at least one supporting set exists. Consequently, we can show that independently of the chosen support function f as in Definition 4.35, sequence seq will be an evidential sequence of FN covering all of its arguments. Hence, by Theorem 4.38, FN is strongly valid. □

Theorem 4.41. *Let $D = (A, L, C)$ be an ADF. If D is strongly valid, then it is weakly and relation valid.*

Proof. Let us start with D. We can observe that every pd–function in D is sound. For any argument $a \in A$ and its arbitrary minimal decisively in interpretation, we can find a pd–function assigning this interpretation to a. Due to soundness, this function will produce a standard evaluation for a. Based on the strong validity, this evaluation can be made acyclic w.r.t. the used pd–function. Clearly, we can "trim" it down to an acyclic pd–evaluation for a. Thus, every argument $a \in A$ will have an acyclic pd–evaluation and D is in the weakly valid form. Since such an evaluation can be obtained with an arbitrary decisively in interpretation for a and D is redundancy–free, then D is also relation valid. □

Theorem 4.42. *Let $D = (A, L, C)$ be a strongly valid ADF. A set of arguments $E \subseteq A$ is pd–acyclic iff for every argument $a \in E$ there exists a minimal decisively in interpretation v_a s.t. $v_a^{\mathbf{t}} \subseteq E$.*

Proof. By the definition of pd–acyclic sets, if E is pd–acyclic then every argument $a \in E$ has an acyclic pd–evaluation on E. Therefore, every $a \in E$ will have a decisively in interpretation s.t. its positive part is contained in E.

If every argument $a \in E$ has a decisively in interpretation s.t. its positive part is contained in E, then clearly we can create a suitable pd–function on E and create a standard evaluation on E containing all arguments in E in its pd–sequence. We can assign arbitrary interpretations to arguments in $A \setminus E$ to create a pd–function for A and an associated acyclic pd–evaluation. Since arguments in E depend only on other arguments in E, we can obtain an appropriate sub–evaluation of our evaluation and it will also be an acyclic one. Hence, E is pd–acyclic. $\square$

Theorem 4.43. *Let $D = (A, L, C)$ be an ADF. If D is strongly valid, then it is an AADF$^+$. If D is a redundancy–free cleansed AADF$^+$, then it is strongly valid.*

Proof. Let us assume that D is strongly valid, but not an AADF$^+$. This means there exists a sound pd–function pd^D_E on a set $E \subseteq A$ s.t. the standard evaluation (E, B) produced by pd^D_E cannot be made acyclic. We can thus represent it as a partially acyclic evaluation (F, G, B) s.t. $F \cup G = E$. We can observe that including additional mappings for $A \setminus E$ into the pd–function will not change the fact that F cannot be ordered into a pd–sequence. Consequently, A cannot be made into an acyclic pd–evaluation w.r.t. a pd–function subsuming pd^D_E and thus D could not have been strongly valid in the first place.

Let now D be a redundancy–free cleansed AADF$^+$. Since it is cleansed, then every argument will possess a standard evaluation and thus every acceptance condition will have a set of arguments it maps to in. Therefore, every pd–function in D will be sound. Now, as we can create a standard evaluation s.t. its pd–set is A and it can be made acyclic w.r.t. any pd–function we made it with, then clearly the strong validity requirements hold. $\square$

Theorem 4.44. *Let $D = (A, L, C)$ be a redundancy–free ADF. D is strongly valid iff there exists a sequence $(a_0, ..., a_n)$ of all arguments in A s.t. given any pd–function pd on A, $((a_0, ..., a_n), \bigcup_{i=0}^{n} pd(a_i))$ is an acyclic pd–evaluation.*

Proof. Let us start with showing that if D is strongly valid, then a suitable sequence exists. We can find an argument $a \in A$ s.t. for every minimal decisively in interpretation v_a of this argument, $v_a^{\mathbf{t}} = \emptyset$. If it were not the case, then to every argument we could assign an interpretation with a non–empty $\mathbf{t}$ part and we would not be able to construct an acyclic evaluation using such a pd–function. This clearly contradicts strong validity of D. We can put the arguments possessing minimal decisively in interpretations with empty $\mathbf{t}$ parts at the start of our sequence in an arbitrary order. We will denote the collection of these arguments with $A_0 \subseteq A$. If $A_0 = A$, then our proof is done. Let us thus assume it is not the case.

We can now find an argument $b \in A \setminus A_0$ s.t. for every minimal decisively in interpretation v_b of this argument, $v_b^{\mathbf{t}} \subseteq A_0$. If it were not the case, then for every argument in $A \setminus A_0$ we could find an interpretation s.t. its $\mathbf{t}$ part is not a subset of A_0. This also means that this $\mathbf{t}$ part would be non–empty. Again, we can use these interpretations to make a pd–function that cannot produce an acyclic pd–evaluation on A. We can collect all such arguments into a set $A_1 \subseteq A$. We add the arguments in A_1 in arbitrary order to the sequence created for A_0.

We can continue this line of reasoning until we go through all the arguments and obtain our sequence. Based on the construction, we can observe that this sequence will be a pd–sequence of an acyclic evaluation w.r.t. any pd–function on A.

If there is a sequence s.t. independently of the pd–function, it induces an acyclic evaluation, then clearly the strong validity restrictions are satisfied. $\square$

Theorem 4.45. *Let $D = (A, L, C)$ be a redundancy–free and cleansed ADF and $PDG^D = (A, L')$, where $L' = \{(a, b) \mid \exists v \in min\_dec(in, b)$ s.t. $a \in v^{\mathbf{t}}\}$, its associated positive dependency graph. D is strongly valid iff PDG^D is a directed acyclic graph.*

Proof. If D is strongly valid, then by Theorem 4.44 there exists a sequence seq of arguments s.t. independently of the created pd–function, it is a sequence of an acyclic pd–evaluation. This means that for every two arguments a and b s.t. $\exists v \in min\_dec(in, b)$ and $a \in v^{\mathbf{t}}$, a precedes b in seq. Therefore, it is easy to see that seq is a topological ordering of the positive dependency graph PDG^D. Hence, PDG^D is a directed acyclic graph.

If PDG^D is a directed acyclic graph, then there exists a topological ordering seq of its nodes. Based on the construction of PDG^D, this means that for every two arguments a and b s.t. $\exists v \in min\_dec(in, b)$ and $a \in v^{\mathbf{t}}$, a precedes b in seq. Consequently, independently of the chosen pd–function, this sequence seq will be always meet the requirements of a pd–sequence of D covering all of its arguments. Hence, by Theorem 4.44, D is strongly valid. $\square$

Lemma 4.67. *The following holds between the subclasses and normal forms of EAF(C)s:*

- $NDef^{EAF(C)} \subsetneq BH^{EAF(C)}$
- $NDef^{EAF(C)} \subsetneq SCons^{EAF(C)}$
- $BH^{EAF(C)} \subsetneq SCons^{EAF(C)}$
- $NDef^{EAFC} \subsetneq Bin^{EAFC} \subsetneq Min^{EAFC}$

Proof.

- Let $EF = (A, R, D)$ be an EAF in $NDef^{EAF}$. We can see see the framework as a single partition $((A, R), D)$, and since $D = \emptyset$, all of the bounded hierarchical requirements are easily satisfied. Similar holds for EAFCs. At the same time, we can easily imagine a bounded hierarchical EAF(C) that does have some defense attacks, thus the subset relation is strict.
- Trivial.

- Let $EF = (A, R, D)$ be an EAF in BH^{EAF} and $(((A_1, R_1), D_1), ..., ((A_n, R_n), D_n)$ its partition that satisfies the requirements from Definition 2.55. By this definition, every argument in A_i is attacked only by arguments in A_i and these conflicts are defense attacked only by arguments from A_{i+1} (or not defense attacked at all if $i = n$). Since $A_i \cap A_{i+1} = \emptyset$, strong consistency follows easily. Same holds for EAFCs. We can observe that the framework from Example 5 is strongly consistent. However, it is not bounded hierarchical. Thus, the subset relation is strict.

- If there are no defense attacks to start with, then the binary conditions for EAFCs are trivially satisfied. If every defense attacking set contains a single argument only, then the defense attacks targeted at a give conflict are clearly incomparable. Thus, the framework is in minimal form. We can easily produce framework that are minimal, but not binary, and those that are binary, but do have some defense attacks. Hence, the relation between the subclasses is strict.

□

Lemma 4.71. *The following holds between the subclasses and normal forms of AFNs:*

- $NSup^{AFN} \subsetneq SCons^{AFN}$
- $NSup^{AFN} \subsetneq SBin^{AFN}$
- $NSup^{AFN} \subsetneq SSig^{AFN}$
- $NSup^{AFN} = Sup_0^{AFN} \cap SV^{AFN}$
- $SBin^{AFN} \subsetneq Min^{AFN}$
- $SSig^{AFN} \subsetneq Min^{AFN}$
- $SV^{AFN} \subsetneq (WV^{AFN} \cap RV^{AFN})$

Proof.

- Trivial.
- Trivial.
- Trivial.
- Clearly, $NSup^{AFN} \subseteq Sup_i^{AFN}$ for any $0 \leq i$. If the framework does not contain any simple paths, then there is no supporting edge between two different arguments. However, there may be a support edge with the same target and source. Consequently, if it is also acyclic, then there is no supporting edge between any arguments, and thus $NSup^{AFN} \subseteq (Sup_0^{AFN} \cap SV^{AFN})$. Showing that $(Sup_0^{AFN} \cap SV^{AFN}) \subseteq NSup^{AFN}$ follows similarly.

- If every supporting set contains a single argument only, then the sets supporting a given argument are clearly incomparable. Thus, the framework is in minimal form. At the same time we can easily imagine a framework consisting of three arguments a, b and c, where $\{a, b\}$ supports c. Such a framework is minimal, but not support binary. Thus, the relation between the subclasses is strict.

- Every argument can be supported by at most one set. Thus, it cannot be supported by its (strict) subset and minimality follows easily. Since every binary framework is minimal and clearly not every singular framework has to be binary, the relation between the subclasses is strict.

- See Theorem 4.31. We can adapt Example 59 in order to show that not every weakly and relation valid AFN has to be strongly valid as well.

$\square$

Lemma 4.73. *The following holds between the subclasses and normal forms of EAFs:*

- $(ABin^{EAS} \cup ASig^{EAS}) \cap (SBin^{EAS} \cup SSig^{EAS})) \subsetneq Min^{EAS}$.
- $EvSup^{EAS} \subsetneq (SBin^{EAS} \cap SSig^{EAS})$
- $EvSup^{EAS} \subsetneq SCons^{EAS}$.
- $(EvSup^{EAS} \cap AllSup^{EAS}) \subsetneq (SCons^{EAS} \cap SV^{EAS})$
- $SV^{EAS} \subsetneq (WV^{EAS} \cap RV^{EAS})$
- $WV^{EAS} \subsetneq AllSup^{EAS}$

Proof.

- Please recall that minimal form of EAS deals with both attack and support relations. It is easy to see that attack binary and attack singular frameworks easily satisfy the minimality restrictions for conflict. Similarly, support binary and singular structures meet the requirements concerning support minimality. Therefore, the combination of these properties produces a minimal EAS. At the same time, the framework $(\{\eta, a, b, c, d\}, \{(\{a, b\}, c), (\{a, d\}, c)\}, \{(\{a, b\}, c), (\{a, d\}, c)\})$ is neither attack (support) singular nor binary. However, it is minimal. Thus, the subclass relation is strict.

- Follows straightforwardly from the definition of the pure evidence supported subclass. The framework $(\{\eta, a\}, \emptyset, \{(\{a\}, a)\})$ is both support binary and singular, but is not pure evidence supported. Therefore, the subclass relation is strict.

- By definition, η cannot participate in attacks on any argument. Since every set of arguments supporting a given argument through the evidence relation is precisely $\{\eta\}$, clearly the strong consistency restrictions are met.

- If a framework is in $EvSup^{EAS} \cap AllSup^{EAS}$, then every non–$\eta$ argument is supported by $\{\eta\}$ only. Thus, there is only one function f assigning a given argument its support as defined in Definition 4.35. We can observe that any sequence of arguments putting η in the beginning will be an evidential sequence. Thus, the framework is in SV^{EAS}. Moreover, based on the previous point of this proof, it is also in $SCons^{EAS}$. Please note the relation is strict; for example, the framework $(\{\eta, a\}, \emptyset, \{(\{\eta\}, a)\})$ is strongly consistent and strongly valid, but is not pure evidence supported.

- See Theorem 4.36. Moreover, we can use Example 59 to show that the relation is strict.

- Assume it is not the case and there exists a weakly valid EAS that is not all–supported. This means there exists a non–η argument that receives no support at all. However, this means it cannot possess an evidential sequence, and thus our EAS cannot be weakly valid. We reach a contradiction. However, not every all–supported EAS has to be weakly valid; again, the framework $(\{\eta, a\}, \emptyset, \{(\{a\}, a)\})$ is a simple counterexample.

□

Lemma 4.77. *The following holds between the subclasses and normal forms of ADFs:*

- $ADF^{AF} \subsetneq (ADF^{SETAF} \cap RFree^{ADF})$
- $ADF^{SETAF} \subsetneq (BADF \cap AADF^{+})$
- $ADF^{SETAF} \subsetneq WV^{ADF}$
- $ADF^{SETAF} \not\subseteq RFree^{ADF}$
- $BADF \not\subseteq AADF^{+}$ *and* $AADF^{+} \not\subseteq BADF$
- $WV^{ADF} \subsetneq Cln^{ADF}$
- $SV^{ADF} \subsetneq AADF^{+}$
- $(AADF^{+} \cap RFree^{ADF} \cap Cln^{ADF}) \subseteq SV^{ADF}$
- $SV^{ADF} \subsetneq (WV^{ADF} \cap RV^{ADF})$
- $(RV^{ADF} \cap Cln^{ADF}) \subsetneq WV^{ADF}$

Proof.

- The fact that $ADF^{AF} \subsetneq ADF^{SETAF}$ follows easily from the definitions of the subclasses. Let $D = (A, L, C) \in ADF^{AF}$. Based on the definition of the Dung–style conditions, we can observe that for every parent b of an argument $a \in A$, it holds that $C_a(\emptyset) = in$ and $C_a(\{b\}) = out$. Thus, the (b, a) link is not supporting, and therefore cannot be redundant. Hence D is redundancy–free. At the same time, the ADF $(\{a, b, c\}, \{C_a = \top, C_b = \top, C_c = \neq a \vee \neg b\})$ is SETAF–style and redundancy–free, but is not AF–style. Thus, the relation is strict.

- Let $D = (A, L, C) \in ADF^{SETAF}$. Let $(a, b) \in L$ and $E \subseteq par(b)$ a subset of parents of b in D. From the construction of the condition we can observe that if $C_b(E) = out$, then $C_b(E \cup \{a\}) = out$ as well. Therefore, the (a, b) link is attacking, and D is a BADF. Let $a \in A$ be an argument. a may have more than one minimal decisively in interpretation, however, in all of them the $\mathbf{t}$ part is empty and $\mathbf{f}$ corresponds to some subset of parents of a. Consequently, every argument has a standard evaluation and every standard evaluation can be made acyclic as any argument in the framework satisfies the a_0 requirements of acyclic pd–evaluations. Hence, D is an AADF$^+$. Finally, we can consider a framework $(\{a, b\}, \{C_a = \top, C_b = a\})$. It is both an AADF$^+$ and a BADF, but is not SETAF–style. Therefore, the relation is strict.

- See previous point.

- Let us consider the ADF consisting of arguments $\{a, b, c\}$. The functional conditions for a and b simply assign in to $\emptyset$. The condition for c assigns in to $\emptyset$ and $\{a\}$, while $\{b\}$ and $\{a, b\}$ are mapped to out. We can observe that the (a, c) link is in fact redundant; to whatever subset of parents we add a, the value of the condition stays the same. Therefore, the SETAF–style ADFs are not necessarily redundancy–free.

- See Example 32.

- See Theorem 4.23.

- See Theorem 4.43.

- See Theorem 4.43.

- See Theorem 4.41. We can adapt Example 59 to show that the relation is strict.

- Let $D = (A, L, C) \in RV^{ADF} \cap Cln^{ADF}$ be a relation valid and cleansed ADF. Since D is cleansed, it holds that for every argument $a \in A$ there exists a set $E \subseteq par(a)$ s.t. $C_a(E) = in$. Consequently, every argument possesses a decisively in interpretation (and thus a minimal one too). Due to the fact that D is relation valid as well, this minimal interpretation will be used in constructing an acyclic pd–evaluation for a. Thus, a will possess an acyclic pd–evaluation, and it holds that D is weakly valid.

□

15.3 Translating AFs: Proof Appendix

Theorem 5.6. *Let $F = (A, R)$ be a Dung's framework and $FR^F = (A, R)$ its corresponding AFRA obtained by Translation 18. If $E \subseteq A$ is an admissible (conflict–free) extension of F, then $E^{\rightarrow AFRA}$ is an admissible (conflict–free) extension of FR^F. If $E' \subseteq A \cup R$ is an admissible (conflict–free) extension of FR^F, then $E' \cap A$ might not be admissible (conflict–free) in F.*

Proof. Let $E \subseteq A$ be a conflict–free extension of F and let $E' = E^{\rightarrow AFRA}$. Since E is conflict–free, this means there is no attack between any two members of the set in F. Hence, by Translation 18, there is no such attack in FR^F as well and there will be no attack in E' with a target in E'. Thus, E' is conflict–free in FR^F. Let $E \subseteq A$ be a set of arguments in F that is not conflict–free. The set E contains only arguments and thus there will be no elements $x, y \in E$ s.t. x defeats y. Thus, E is conflict–free in FR^F.

Let $E \subseteq A$ be an admissible extension of F and let $E' = E^{\rightarrow AFRA}$. Based on the discussion above, E' is conflict–free in FR^F. Since for every $a \in A$ that attacks an argument in E there is some $b \in E$ attacking it, then by construction of E' and the fact that only arguments get attacked in FR^F, it means that for every attack $x \in R$ with a target in E' there is some attack in E' directed at the source of x and thus defeating x. Thus, every argument in E' is acceptable w.r.t. E'. Since there are no attacks directed at attacks in FR^F, it holds that if the source of an attack is acceptable then so is the attack itself. Thus, E' is admissible in FR^F.

To see that not every AFRA admissible extension corresponds to an AF admissible one, please consult Example 69. □

Theorem 5.7. *Let $F = (A, R)$ be a Dung's framework, $FR^F = (A, R)$ its corresponding AFRA obtained by Translation 18 and $\sigma \in \{$admissible, complete, preferred, grounded, stable$\}$ a semantics. If $E \subseteq A$ is a conflict–free extension of F, then $E^{\rightarrow AFRA}$ is a conflict–free extension of FR^F. If $E' \subseteq A \cup R$ is a conflict–free extension of FR^F, then $E = (E' \cap A) \cup \{src(x) \mid x \in E' \cap R\}$ might not be conflict–free in F. If E is a σ–extension of F, then $E^{\rightarrow AFRA}$ is a σ–extension of FR^F. If $E' \subseteq A \cup R$ is a σ–extension of FR^F, then $E = (E' \cap A) \cup \{src(x) \mid x \in E' \cap R\}$ is a σ–extension of F.*

Proof. The first direction follows already from Theorems 5.5 and 5.6. The same conflict–freeness analysis holds as well. Let us now focus on the other direction for other semantics.

Let $E' \subseteq A \cup R$ is an admissible extension of FR^F. Let us first assume that $E = (E' \cap A) \cup \{src(x) \mid x \in E' \cap R\}$ is not conflict–free in F, i.e. there are two arguments $a, b \in E$ s.t. aRb. This means that b or any attack with b as its source is defeated by the (a, b) attack in E'. As E' is admissible, it has to contain an attack defeating (a, b), and since in FR^F we only deal with argument attack targets, E' has to contain an attack (x, a), where $x \in A$. Since $a \in E$, then either $a \in E'$ or $(a, y) \in E'$ for some $y \in A$; as both of them are defeated by (x, a), E' cannot be conflict–free in FR^F. We reach a contradiction and thus it has to be the case that E is conflict–free in F. Let us now assume it is not admissible, i.e. there is some argument $a \in E$ attacked by an argument $b \in A$ which is in turn not attacked by any argument in E. By construction of E this means that there cannot be any attack directed at (b, a) in E' and thus neither a nor any attack carried out by a can be acceptable w.r.t. E'. Thus, E' is not admissible in FR^F and we reach a contradiction.

In the previous section we have already explained that due to the nature of FR^F, if argument $a \in A$ is defended by a given set, so is any attack that has a as the source. Moreover, it also holds that if an attack is defended then so is its source by Lemma 2.32. This brought us to the conclusion that every complete extension E' of FR^F is of the form $E' = (E' \cap A)^{\rightarrow AFRA}$. What this means for us is that the source of any attack is already in the set and thus $(E' \cap A) \cup \{src(x) \mid x \in E' \cap R\} = E' \cap A$. Consequently, we simply come back to Theorem 5.5. □

Theorem 5.9. *Let $EF = (A, R, D) \in NDef^{EAF}$ be an EAF without defense attacks. The following holds:*

- *an argument a defeats an argument b w.r.t. any set of arguments E iff $(a, b) \in R$*
- *a set of arguments $E \subseteq A$ is conflict–free extension of EF iff there are no $a, b \in E$ s.t. aRb*
- *given a set of arguments $E \subseteq A$, a set containing a pair $\{(x, y)\}$ s.t. x defeats$_E$ y is a reinstatement set on E for the defeat$_E$ by x on y iff $x \in E$.*
- *an argument $a \in A$ is acceptable w.r.t. a set of arguments $E \subseteq A$ iff for every argument b s.t. bRa, there is $c \in E$ s.t. cRb*
- *a set of arguments E is a stable extension of EF iff for every argument $b \notin E$, $\exists a \in E$ s.t. aRb*

Proof. Since there will never be any argument defense attacking (a, b), every attack will always result in a defeat.

If there are no attacks in the set to start with, then it is trivially conflict–free in EF. By definition, a set is conflict–free if for every attack aRb s.t. $a, b \in E$, there is a defense attack and $(b, a) \notin R$. Since $D = \emptyset$, aRb will never be "overriden" and thus if any conflicting arguments are present in the set, the condition will not be satisfied and the set will not be conflict–free. Thus, a conflict–free set will not contain any a, b s.t. aRb.

Concerning the reinstatement, the first condition that the pair is in the set is already satisfied. The second is simplified to just x being present in E. Due to lack of defense attacks altogether, the third condition of the original reinstatement definition is automatically satisfied and can be dropped.

Since in the simplified framework defeats can be replaced with attacks and $\{(c, b)\}$ is a trivial reinstatement set on E for the (c, b) attack (note that $c \in E$), the acceptability definition comes back to the original Dungean version easily.

The simplification of the stable semantics also follows from the defeat–attack equivalence. □

Theorem 5.10. *Let $F = (A, R)$ be a Dung's framework and $EF^F = (A, R, \emptyset)$ its corresponding EAF obtained through Translation 19. A set of arguments $E \subseteq A$ is a σ–extension of F, where $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ iff it is a σ–extension of EF^F.*

Proof. The fact that conflict–free, admissible, complete, preferred and stable extensions coincide between the two frameworks follows easily from Theorem 5.9. F and EF^F have the same arguments and attacks and with the simplifications, the definitions of the semantics become identical.

The EF^F framework is not an arbitrary EAF; it is in fact a bounded hierarchical one (Theorem 5.8). By Definition 2.57, its grounded extension is the least fixed point of the EAF characteristic operator. By Theorem 2.10, the grounded extension of F is also the least fixed point of the AF characteristic operator. The only difference is that the domain of the first is restricted only to conflict–free sets; however, since the grounded extension is obviously also conflict–free, we can conclude that the grounded extensions of F and EF^F coincide. □

Theorem 5.18. *Let $F = (A, R)$ be a Dung's framework, $D^F = (A, R, C)$ its corresponding ADF obtained through Translation 23 and $E \subseteq A$ a set of arguments. The following holds:*

- *E^- in F equals the union of parents of all arguments in E in D^F.*
- *E is conflict–free in F iff it is conflict–free in D^F.*
- *if E is conflict–free, then E^+ in F coincides with the discarded set of E in D^F.*

Proof.

- Obvious by Translation 23.
- If E is conflict–free in F, then for all arguments $a \in E$, $E \cap \{a\}^- = \emptyset$. Consequently, $E \cap par(a) = \emptyset$. Since by Translation 23 for all $a \in A$, $C_a(\emptyset) = in$, every argument in E has a satisfied acceptance condition in D^F and the set is (ADF) conflict–free. The other way around follows accordingly: since the arguments have a satisfied acceptance condition, none of its parents (and thus attackers) is in the set and (Dung) conflict–freeness is preserved.
- Let v be an interpretation mapping only and all elements of E to $\mathbf{t}$. If an argument a is in E^+, then $E \cap \{a\}^- \neq \emptyset$. Consequently, $E \cap par(a) \neq \emptyset$, and thus the condition of a is out. It is easy to see that the argument is decisively out w.r.t. v, and since v_E is its completion, then w.r.t. v_E as well. Thus, $v_E(a) = \mathbf{f}$, and a is in the (ADF) discarded set.

  Since a is in the (ADF) discarded set of E, then it is decisively out w.r.t. v_E by Theorem 2.150. Consequently, $v_E(C_a) = out$, i.e. $C_a(E \cap par(a)) = out$. Hence, by Translation 23, it has to be the case that $E \cap par(a) \neq \emptyset$, and thus $E \cap \{a\}^- \neq \emptyset$. Therefore, a qualifies for (Dung's) E^+ set.

□

Theorem 5.19. *Let $F = (A, R)$ be a Dung's framework, $D^F = (A, R, C)$ its corresponding ADF obtained through Translation 23. Let $E \subseteq A$ be a conflict–free set of F and D^F and $a \in A$ an argument. E defends a in F iff a is decisively in w.r.t. v_E in D^F.*

Proof. If E defends a, it means that $\{a\}^- \subseteq E^+$ in F. Consequently, $par(a) \subseteq E^+$ in D^F. Since every parent of a is falsified by v_E, a is decisively in w.r.t. v_E. The other way is straightforward. As a is decisively in w.r.t. v_E, then all of its parents are in the discarded set of D_F, and thus in E^+ of F. Since parents of a are precisely $\{a\}^-$, a is defended by E in F. □

Theorem 5.20. *Let $F = (A, R)$ be a Dung's framework and $D^F = (A, R, C)$ its corresponding ADF obtained through Translation 23. A set of arguments $E \subseteq A$ is a conflict–free extension of F iff it is (pd–acyclic) conflict–free in D^F. $E \subseteq A$ is a stable extensions of F iff it is (stable) model of D^F. $E \subseteq A$ is a grounded extensions of F iff it is (acyclic) grounded in D^F. $E \subseteq A$ is a σ extensions of F, where where $\sigma \in \{$admissible, preferred, complete$\}$ iff it is an xy–σ–extension of D^F for $x, y \in \{a, c\}$.*

Proof. Due to the fact that the semantics classification collapses for D^F (see Theorems 2.172 and 5.17), it suffices to focus on only conflict–free, grounded, model, and the cc–types of the ADF semantics.

The behavior of the conflict–free semantics was already shown in Theorem 5.18. By Theorem 5.19 all arguments in E are defended iff they are decisively in w.r.t. v_E, thus admissibility and completeness follow straightforwardly. Since both in F and D^F preferred extensions are subset maximal admissible, they obviously coincide. The grounded extension of F is by Theorem 2.10 the least w.r.t. set inclusion complete one. Same follows for D^F by Theorem 2.158, and since complete extensions coincide, so do grounded.

Let E be stable in D^F. By definition it is also a model and from Lemma 2.159 we know that $E^{a+} = A \setminus E$. By Theorems 2.172 and 5.17 it holds hat $E^{a+} = E^+$. From Theorem 5.18 we now have that E is conflict–free in F and $E^+ = A \setminus E$ in F. Thus, E is stable in F.

Now, since E is stable in F, then $E^+ = A \setminus E$ in F. Consequently, $E^+ = A \setminus E$ in D^F by Theorem 5.18. Thus, by Theorem 2.150, all elements in $A \setminus E$ are out. It is clear that model conditions are satisfied and thus by Theorems 2.172 and 5.17, E is stable in D^F. □

15.4 Translating SETAFs: Proof Appendix

Theorem 6.1. *Let $SF = (A, R)$ be a SETAF and F^{SF} its corresponding AF obtained by Translation 25. If $E \subseteq A$ is a σ–extension of SF, where $\sigma \in \{$conflict–free, admissible,* *preferred, complete, grounded, stable*$\}$*, then* $arg(E) \cup att(E)$ *is a* σ*–extension of* F^{SF}*. If a set of arguments* $E' \subseteq A'$ *is a* σ'*–extension of* F^{SF}*, where* $\sigma' \in \{$*admissible, preferred, complete, grounded, stable*$\}$*, then* $\bigcup E'$ *is a* σ*–extension of* SF*.*

Proof. First of all, let us note that if an argument $a \in A$ is attacked by a set $E \subseteq A$, then by the translation, the arguments $\{a\} \in A'$ and $B \in att(A)$ s.t. $a \in B$ are attacked by an AF argument corresponding to E contained in $arg(E) \cup att(E)$. If a is not attacked by E in SF, then $\{a\}$ is not attacked by $arg(E) \cup att(E)$ either.

Let E be a conflict–free extension of SF. By the above this means that no argument in $arg(E)$ is attacked by $E' = arg(E) \cup att(E)$ in F^{SF}. Moreover, since there is no argument in $arg(E)$ attacked by E', then by the construction of R' there is no argument in $att(E)$ attacked by E' either. Consequently, E' is conflict–free in F^{SF}. Please note it does not hold that a conflict–free extension of F^{SF} is conflict–free in SF, as already explained in Example 77.

Let us now show that if an argument $a \in A$ is defended by a set $E \subseteq A$ in SF, then the argument $\{a\} \in A'$ is defended by $E' = arg(E) \cup att(E)$ in F^{SF}. Let set $B \subseteq A$ be an arbitrary set attacking a in SF. We know that at least one argument $b \in B$ is attacked by E due to defense. This means that upon translating, the AF argument corresponding to B attacks $\{a\}$ and that any argument containing B is attacked by E'. Therefore, $\{a\}$ is defended by E' in F^{SF}. We can also observe that if a set of arguments $C \subseteq arg(A)$ is defended by a given set, then so are the arguments in $att(\bigcup C)$ – this comes from the fact that the conflicts directed at att arguments are propagated from their arg arguments. Moreover, if a given argument $C \in att(A)$ is defended by a given set, then so are all other arguments $C' \in A'$ s.t. $C' \subseteq C$. The fact that if E is admissible in SF, then so is $E' = arg(E) \cup att(E)$ in SF^F follows straightforwardly from the analysis above.

Let us assume that $E \subseteq A'$ is an admissible extension of F^{SF}, but $E' = \bigcup E$ is not admissible in SF. This means that either E' is not conflict–free in SF or there exists an argument $e \in E'$ that is not defended by E' in SF. If E' is not conflict–free, this means that there exists $X \subseteq E', b \in E'$ s.t. XRb in SF. Consequently, in F^{SF} there exist arguments $B \in E$ and $X \in A'$ s.t. $b \in B$ and $XR'B$. Since E defends B in F^{SF}, it has to be the case that $\exists C \in E$ s.t. $CR'X$. If $X \in arg(A)$, then by construction of R' every argument $X' \in att(A)$ s.t. $X \subseteq X'$ is also attacked by C. If $X \in att(A)$, then by construction it means that there exists some argument $X' \in arg(A)$ s.t. $X' \subseteq X$ which is again attacked by C. This means that whatever were the arguments in E that made the set X appear in E', there is a conflict in E and thus the set could not have been conflict–free in F^{SF} in the first place. We reach a contradiction. Let us now assume that there is an argument $e \in E'$ that is not defended by E'. This means that there exists a set of arguments $X \subseteq A$ s.t. XRe and no $x \in X, C \subseteq E'$ s.t. CRx. From this follows that in F^{SF} the argument $X \in A'$ attacks all arguments $F \in A'$ s.t. $e \in F$. Since E contains, and thus defends, at least one such F, it means that there exists an argument $X' \in E$ s.t. $X'R'X$. Hence, $X' \subseteq E'$ and $\exists x \in X$ s.t. $X'Rx$ in SF and we reach a contradiction. Therefore, E' has to be admissible in SF.

Let us now assume that $E \subseteq A$ is complete in SF, but $E' = arg(E) \cup att(E)$ is not complete in F^{SF}. Since we know that this set is at least admissible, it has to be the case that there is an argument $a \in A'$ which is defended by E' but is not in E'. If $a \in att(A)$, then it means that arguments in $a' \in arg(A)$ s.t. $a' \subseteq a$ are also defended. If they are all in the set, then by definition of E' so is a'. Thus, what we need to focus on is the case in which there is an argument $a \in arg(A)$ that is defended by E' in F^{SF} but is not included in E'. Let $B \in A'$ be an arbitrary attacker of a through R' in F^{SF}. Since E' defends a, then B is attacked by E', which by the translation means that there are some $X' \subseteq E, y \in B$ s.t. $(X', y) \in R$ in SF. This means that the attacking set B is attacked by E in SF and that the argument represented by a had to be defended by E in SF as well. Thus, we reach a contradiction and it follows that if E is complete in SF, then so is E' in F^{SF}.

Let $E \subseteq A'$ be a complete extension of F^{SF}. We can observe that $\exists B \subseteq A$ in SF s.t. $E = arg(B) \cup att(B)$. This comes from the fact that if a set of arguments $C \subseteq arg(A)$ is defended by a given set in F^{SF}, then so is $att(\bigcup C)$ due to the fact that attacks on an att argument is created from the attacks directed at its elements which form arg arguments. Moreover, if a given argument $C \in att(A)$ is defended by a given set in F^{SF}, then so are all other arguments $C' \in A'$ s.t. $C' \subseteq C$. Therefore, a complete extension contains all att arguments induced by ones in arg and if an att argument is contained, then so are its respective arg ones.

Let us now assume that even though $E \subseteq A'$ is complete in F^{SF}, $E' = \bigcup E$ is not complete in SF. We know that E' is at least admissible in SF. Thus, it has to be the case that there exists an argument $a \notin E'$ defended by E' in SF, i.e. $\forall C \subseteq A$ s.t. CRa, $\exists X \subseteq E', c \in C$ s.t. XRc. This means that there exists an argument $\{a\} \notin E$ in F^{SF} attacked by an argument $C \in A'$ and that there is an argument $X \in A'$ s.t. $XR'C$. Since $X \subseteq E'$ and a complete extension of F^{SF} is of the form $\exists B \subseteq A$ s.t. $E = arg(B) \cup att(B)$, it has to be the case that $X \in E$. Consequently, $\{a\}$ is defended by E in F^{SF} and the set could not have been complete in the first place – we reach a contradiction. Hence, E' has to be complete in SF.

Let us now assume that $E \subseteq A$ is preferred in SF, but $E' = arg(E) \cup att(E)$ is not preferred in F^{SF}. This means there exists a complete extension X in F^{SF} s.t. $E' \subset X$. However, by the analysis above we can show that $\bigcup X$ is then a complete extension of SF and from the way F^{SF} complete extensions are structured, that $E \subset \bigcup X$ in SF. Therefore, E could not have been preferred in the first place and we reach a contradiction.

Let us now assume that $E \subseteq A'$ is preferred in F^{SF}, but $E' = \bigcup E$ is not preferred in SF. This means there exists a complete extension $X \subseteq A$ s.t. $E' \subset X$. By the analysis above, we can show that $X' = arg(X) \cup att(X)$ is admissible in F^{SF}. It is easy to see that $E \subset X'$, and thus E could not have been preferred in F^{SF} in the first place. Consequently, E' is preferred in SF.

By using the fact that the grounded extension is the least complete one both in AFs and SETAFs by Theorems 2.10 and 2.24, we can prove in a way similar to preferred semantics that if E is grounded in SF, then so is $arg(E) \cup att(E)$ in F^{SF} and that if E' is grounded in F^{SF}, then so is $\bigcup E'$ in SF.

Finally, we are left with the stable semantics. Let E be stable in SF. We know that $E' = arg(E) \cup att(E)$ is at least complete in F^{SF}. Let us now assume that it is not stable, i.e. $\exists a \notin E'$ for which there is no $x \in E'$ s.t. $xR'a$. If $a \in att(A)$ and is not attacked by E', then by construction of R' it follows that any argument in $a' \in arg(A)$ s.t. $a' \subseteq a$ is not attacked by E' either. Consequently, we can focus on the case when the unattacked argument is just in $arg(A)$. However, by construction of R' and E' this means that the argument represented by a could not have been attacked by E in the first place. We reach a contradiction.

Let us assume that E is stable in F^{SF}, but $E' = \bigcup E$ is not stable in SF. We know it is at least complete. Thus, it has to be the case that there exists an argument $a \notin E'$ for which there is no $X \subseteq E'$ s.t. XRa. However, by completeness of E, this also means there exists an argument $\{a\} \notin E$ for which there is no $X \in E$ s.t. $XR'\{a\}$ in F^{SF}. Thus, E could not have been stable in the first place and we reach a contradiction. □

Theorem 6.2. *Let $SF = (A, R)$ be a SETAF, $F^{SF}_{def} = (A', R')$ its corresponding defender AF obtained through Translation 26 and $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$ a semantics. If $E \subseteq A$ is conflict–free in SF, then $E \cup \{(X, y) \mid X \subseteq E\} \cup \{x' \mid x' \in X', x \in (A \setminus E)\}$ is conflict–free in F^{SF}_{def}. If $E \subseteq A$ is a σ–extension of SF, then $E \cup \{(X, y) \mid X \subseteq E\} \cup \{x' \mid x' \in X', x \in E^+_{SF}\}$ is a σ–extension of F^{SF}_{def}. If $E' \subseteq A'$ is a σ–extension of F^{SF}_{def}, then $E' \cap A$ is a σ–extension of SF.*

Proof. Let E be conflict–free in SF and let $E' = E \cup \{(X, y) \mid X \subseteq E\} \cup \{x' \mid x' \in X', x \in (A \setminus E)\}$ be a set of arguments in F^{SF}_{def}. Translation 26 removes any attacks between arguments in A and at least partially "transfers" them to auxiliary arguments. Consequently, if E is conflict–free in SF, then E is trivially conflict–free in F^{SF}_{def}. From conflict–freeness of E in SF it follows that for any conflict (X, y) s.t. $X \subseteq E$, $y \notin E$. Therefore, $E \cup \{(X, y) \mid X \subseteq E\}$ is also conflict–free in F^{SF}_{def}. Let now $y' \in \{x' \mid x' \in X', x \in (A \setminus E)\}$. We can observe that y' is added only if y is not present in the set. Consequently, there is no conflict between E and y'. Since y is not in E, no conflict carried out by a set containing y is in E'. Therefore, y' does not attack any conflict argument in E', and we can finally conclude that E' is conflict–free in F^{SF}_{def}.

Not every conflict–free set of F^{SF}_{def} is conflict–free in SF – this comes simply from the conflict transfer. This behavior could have been already observed in Example 79.

Let us now focus on $E \subseteq A$ being an admissible extension of SF and let $E' = E \cup \{(X, y) \mid X \subseteq E\} \cup \{x' \mid x' \in X', x \in E^+_{SF}\}$ be a set of arguments in F^{SF}_{def}. We can observe that due to conflict–freeness of E in SF, $E \cap E^+_{SF} = \emptyset$, and thus $\{x' \mid x' \in X', x \in E^+_{SF}\} \subseteq \{x' \mid x' \in X', x \in (A \setminus E)\}$. Therefore, we can reuse the explanation above to show that E' is conflict–free in F^{SF}_{def}. Let us assume it is not admissible; it means there is an argument $e \in E'$ not defended by E'. This means that there exists an argument $b \in A'$ s.t. $(b, e) \in R'$ and no $c \in E'$ s.t. $(c, b) \in R'$. Let $e \in \{(X, y) \mid X \subseteq E\}$. By the construction of F^{SF}_{def}, it is only attacked by x' arguments corresponding to $x \in X$. However, by the construction of E', $x \in E'$. Since $(x, x') \in R'$, E' has the power to defend e. We reach a contradiction. Let us now focus on $y' \in \{x' \mid x' \in X', x \in E^+_{SF}\}$. If $y \in E^+_{SF}$, then there is a set $Y \subseteq E$ s.t. $(Y, y) \in R$. Consequently, $(Y, y) \in E'$ and $((Y, y), y) \in R'$. Thus, E' defends y' against y. Again, we reach a contradiction. Finally, we come to $e \in E$. If it is not defended by E' in F^{SF}_{def}, then it means that there is an argument $(X, e) \in A'$ s.t. for no $x \in X$, $x' \in E'$. By construction of E' it means that $x \notin E^+_{SF}$. Therefore, $(X, e) \in R$ and no element of X is attacked by E in SF, and this means that E could not have been admissible in SF in the first place. We reach a contradiction and can finally conclude that E' is admissible in F^{SF}_{def}.

Let now $E' \subseteq A'$ be an admissible extension of F^{SF}_{def}. We will show that $E = E' \cap A$ is admissible. First, we need to show that E is conflict–free. Assume it is not the case and that there exists $X \subseteq E, y \in E$ s.t. $(X, y) \in R$. This means that $y \in E'$, $(X, y) \in A'$ and $((X, y), y) \in R'$. Since E' defends y, it has to be the case that there is $x \in X$ s.t. $x' \in E'$. From conflict–freeness of E' in F^{SF}_{def} it thus follows that $x \notin E'$ and thus $x \notin E$. Therefore, X could not have been a subset of E and we reach a contradiction. We can thus conclude that E is conflict–free in SF. Let us now assume that it is not admissible, i.e. there is an argument $a \in E$ and a set $X \subseteq A$ s.t. $(X, a) \in R$, but no $x \in X$ is attacked by any subset of E. This means that $a \in E'$, $(X, a) \in A'$ and $((X, a), a) \in R'$. If there is no $(Y, x) \in R$ s.t. $Y \subseteq E$ for any $x \in X$, then it cannot be the case that a given (Y, x) is in E' – based on the translation, we can observe that without the presence of all arguments in Y in E', the (Y, x) attack argument cannot be defended by E'. We know they are not present, because if all arguments of Y were in E', they would have been in E as well. If no such (Y, x) is in E', then again due to admissibility of E', the argument x' cannot be in E'. Therefore, for no $x \in X$, $x' \in E'$. This means that the (X, a) attack argument is not attacked by any member of E' and thus E' could not have been admissible in F^{SF}_{def} in the first place. We reach a contradiction and can thus conclude that E has to be admissible in SF.

Let $E \subseteq A$ be a complete extension of SF and $E' = E \cup \{(X, y) \mid X \subseteq E\} \cup \{x' \mid x' \in X', x \in (A \setminus E)\}$ the admissible extension of F^{SF}_{def} associated with it. Assume E' is not complete; this means there exists an argument $a \in A' \setminus E'$ defended by E' in F^{SF}_{def}. Let us first assume that $a \in A$, i.e. a represents an argument from SF. If E' defends a, then it means that for every attack argument $(X, a) \in A'$ with an arbitrary X, there is an argument $x' \in E'$ s.t. $x \in X$. Since x' has to be defended by E' due to admissibility, it has to be the case that there is an attack argument $(Y, x) \in E'$. Which, again by admissibility, means that every $y \in Y$ is in E'. Therefore, every such y is in E, and thus E has the power to attack a member of X and defend a from the (X, a) conflict in SF. Consequently, a is defended by E but is not in E, and the set could not have been complete in the first place. We reach a contradiction. Let us now assume that $a = (X, y) \in R$, i.e. a represents an attack from SF. If E' defends a, then it means that every argument $x' \in A'$ s.t. $x \in X$ is attacked by E'. Since a given x' is only attacked by its x, then naturally $x \in E'$ and thus $X \subseteq E'$ and $X \subseteq E$. However, since X is contained in E, then by the construction of E' it has to be the case that $(X, y) \in E'$ and we reach a contradiction with the assumptions. Finally, let us focus on the case where $a = x' \in X'$, i.e. a is a primed version of an argument $x \in A$. Since the only attacker of x' is x itself, then E' defending x' means that there is an attack argument $(Y, x) \in E'$. From admissibility of E' it thus follows that $Y \subseteq E'$ and $Y \subseteq E$. Consequently, $x \in E^+_{SF}$, and by construction of E', $x' \in E'$. We again reach a contradiction and can finally conclude that if E is complete in SF, then so is E' in F^{SF}_{def}.

Let now $E' \subseteq A'$ be a complete extension of F^{SF}_{AF}. We will first show that E' is precisely of the form $E' = E \cup \{(X, y) \mid X \subseteq E\} \cup \{x' \mid x' \in X', x \in E^+_{SF}\}$ for a set $E \subseteq A$. Let $(X, y) \in E'$ be an attack argument in E'. By the admissibility of E' it thus follows that $X \subseteq E'$. Let us first consider E. By construction, it defends any attack arguments (Y, z) s.t. $Y \subseteq E$. Thus, by completeness of E', $(Y, z) \in E'$. We can therefore conclude that the source of any attack is in E' and any attack whose source is in E', is in E'. What remains to be shown is that $E' \cap X'$ is precisely $\{x' \mid x' \in X', x \in E^+_{SF}\}$. If $x \in E^+_{SF}$, then there is $Y \subseteq E$ s.t. $(Y, x) \in R$. Consequently, $Y \subseteq E'$ and by the explanation above, $(Y, x) \in E'$. This means that E' defends x', and due to completeness $x' \in E'$. Now assume that $x' \in E'$, but $x \notin E^+_{SF}$. Since $x' \in E'$, then by admissibility E' has to contain an attack argument (Y, x). Again, by admissibility, $Y \subseteq E'$ and thus $Y \subseteq E$. Consequently, $x \in E^+_{SF}$ and we reach a contradiction. Therefore, we can finally conclude that a complete extension E' of F^{SF}_{AF} has to be precisely of the form $E' = E \cup \{(X, y) \mid X \subseteq E\} \cup \{x' \mid x' \in X', x \in E^+_{SF}\}$ for a set $E \subseteq A$.

We will now show that if $E' \subseteq A'$ is a complete extension of F^{SF}_{def}, then so is $E = E' \cap A$ in SF. We know that E is admissible in SF. If it is not complete, then it means there is an argument $a \in A \setminus E$ defended by E in SF. Thus, for any set of arguments $X \subseteq A$ s.t. $(X, a) \in R$, there is $x \in X, Y \subseteq E$ s.t $(Y, x) \in R$. Since $Y \subseteq E$, then $Y \subseteq E'$, and thus by admissibility of E' $(Y, x) \in E'$. Consequently, E' defends x' and by completeness of E, $x' \in E'$. Now, $(x', (X, a)) \in R'$, and therefore E' can defend a from (X, a). Since the analysis was done for an arbitrary attack, it holds that E' defends a and as E' cannot contain a, we reach a contradiction. Hence, E has to be complete in SF.

We can now observe that there is a one to one relation between the complete extensions of both frameworks and that there is a subset relation between two complete extension of SF iff there is one between their corresponding extensions in F^{SF}_{def}. Therefore, we can show that the preferred extensions of SF and F^{SF}_{def} are related just like the complete ones. Since the grounded extension both in AFs and SETAFs is also the least complete (Theorems 2.10 and 2.24), this analysis extends to the grounded semantics as well. What is left to be analyzed is the stable semantics. Let $E \subseteq A$ be a stable extension of SF and $E' = E \cup \{(X, y) \mid X \subseteq E\} \cup \{x' \mid x' \in X', x \in E^+_{SF}\}$ its corresponding set in F^{SF}_{def}. By the analysis above and Theorem 2.23, we know that E' is preferred. Let us assume it is not stable, i.e. there exists an argument $a \in A' \setminus E'$ that is not attacked by E'. If $a \in A$, then it means there is no $(Y, a) \in E'$ attacking a and by admissibility of E', that Y is not a subset of E'. Therefore, Y is not a subset of E either and E could not have attacked a. We reach a contradiction with the stability of E in SF. If $a = x' \in X'$, then it means that $x \notin E'$ and thus $x \notin E$. By stability of E this means that $x \in E^+_{SF}$, which by construction means that $x' \in E'$. We reach a contradiction with the assumptions. Let now $a = (X, y) \in R$ be an attack argument. If it is not attacked by E', then it means that no x' s.t. $x \in X$ is in E'. However, from previous explanations it follows that if x' is not in E', then x is not in E^+_{SF}, which by stability of E means that $x \in E$. Therefore, $X \subseteq E$ and $X \subseteq E'$, and by completeness of E', $(X, y) \in E'$. We again reach a contradiction and can conclude that E' is stable in F^{SF}_{def}.

Let $E' \subseteq A'$ be a stable (and thus preferred by Theorem 2.9) extension of F^{SF}_{def} and $E = E' \cap A$ its corresponding set in SF. E is preferred in SF. Assume it is not stable; this means there is an argument $a \in A \setminus E$ s.t. no subset X of E attacks a. However, by previous analysis this means that $(X, a) \notin E'$ and that a is not attacked by E'. We reach a contradiction with stability of E' and thus E has to be stable in SF. □

Theorem 6.6. *Let $SF = (A, R)$ be a SETAF and $FN^{SF} = (A', R', N)$ its corresponding AFN obtained by Translation 29. If E is conflict–free in SF, then E is conflict–free in FN^{SF}. If E is a σ–extension of SF, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, then $E' = E \cup att(E)$ is a σ–extension of FN^{SF}. If $E' \subseteq A'$ is conflict–free in FN^{SF}, then $E = E' \cap A$ might not be conflict–free in SF. If E' is a σ'–extension of FN^{SF}, where $\sigma' \in \{$admissible, preferred, complete, grounded, stable$\}$, then $E = E' \cap A$ is a σ'–extension of SF.*

Proof. First of all, let us note that for every argument in FN^{SF} there exists a set $E \subseteq A'$ in which it is coherent. Every argument $a \in A$ has a trivial powerful sequence (a) due to the fact that it requires no support. An argument $b = \{a_1, ..a_n\} \in att(A)$ receives support only from arguments $a_1, .., a_n \in A$, and from the previous remark we can observe that $(a_1, ..., a_n, b)$ is a simple powerful sequence for b. Please note that the presented sequences are also minimal and the elements contained in them will need to be present in any minimal coherent sets for a and b respectively.

The fact that if E is conflict–free in SF, then both E and $E \cup att(E)$ are conflict–free in FN^{SF} follows straightforwardly from Translation 29. The reason why not every conflict–free set in FN^{SF} gives us a conflict–free set of SF is similar as in Theorem 6.1.

We will now show that if an argument $a \in A$ is defended by $E \subseteq A$ in SF, then it is defended by $E' = E \cup att(E)$. Since E defends a in SF, then for any set of arguments $X \subseteq A$ s.t. XRa, there exist $B \subseteq E, x \in X$ s.t. BRx. Consequently, in FN^{SF}, given any argument $X \in A'$ s.t. $XR'a$, there exists $B \in A'$ s.t. either $X \in A$ and $BR'X$, or $X \in att(A)$ and $BR'x$ for an $x \in X$. From the analysis about powerful sets we can thus observe that B has the power to attack any coherent set of X. Since E' is of the form $E \cup att(E)$, then $B \in E'$ and thus E' has the power to defend a against attacks. Finally, it is easy to see that both E' and $\{a\}$ are coherent. Consequently, so is their union, and E' defends a completely. We can also note that if a given set of arguments $C \subseteq A$ is defended by E' in FN^{SF}, then so are the arguments in $att(C)$ by $E' \cup C$ – due to the fact that att arguments are not attacked, only coherence analysis is required.

Let $E \subseteq A$ be an admissible extension of SF. From the previous analysis it follows that $E' = E \cup att(E)$ is conflict–free and defends is members in FN^{SF}. Moreover, as it is also coherent, then we can conclude that E' is an admissible extension of FN^{SF}.

Let $E' \subseteq A'$ be an admissible extension in FN^{SF}. This means that E' is strongly coherent (i.e. conflict–free and coherent) and defends its members. We will first show that $E = E' \cap A$ is conflict–free in SF. Assume it is not the case; this means there exists $B \subseteq E$ and $x \in E$ s.t. BRx. If $|B| = 1$, then it has to be the case that $B \in E'$ – if only an attack containing B was present in E', the set could not have been coherent and thus not admissible. For the same reason, $x \in E'$. Since $BR'x$, then E' could not have been conflict–free in FN^{SF} and we reach a contradiction. Let us now focus on the $|B| > 1$ case. Again, $x \in E'$, but it does not have to be the case that $B \in E'$. Nevertheless, it does hold that $B \subseteq E'$ – the arguments had to appear in E after all and even if an attack argument $B' \subset B$ was in E', then $B' \subseteq E'$ due to coherence. Since E' defends x, then E' attacks every coherent set of B, which from the previous explanations means that there is $a, b \in E'$ s.t. $b \in B$ and $aR'b$. Consequently, we reach a contradiction with the conflict–freeness of E' again and can conclude that E has to be admissible in SF.

We can now focus on defense. If E is not admissible in SF, then it means that there exists a set of arguments $X \subseteq A$ and an argument $s \in E$, s.t. XRs and no $B \subseteq E$ s.t. BRx for some $x \in X$. Consequently, there is an argument $X \in A'$ and $s \in E'$ s.t. $XR's$ and no argument $B \in E'$ s.t. $BR'X$ (if $X \in A$) or $BR'x$ for an $x \in X$ (if $X \in att(A)$). Thus, from the previous analysis we can see that the coherent set $\{X\}$ (or $\{x_1, ...x_n, X\}$ for $X = \{x_1, ..x_n\}$) for X is not attacked by E'. Therefore, E' could not have been admissible in FN^{SF} and we reach a contradiction. Thus, E is admissible in SF.

We can now move on to the complete semantics. Let $E \subseteq A$ be a complete extension of SF. We will show that $E' = E \cup att(E)$ is complete in FN^{SF}. We know it is admissible; if it is not complete, then it means there exists an argument $a \in A' \setminus E'$ that is defended by E'. If $a \in A$, then we can repeat the previously done analysis to show that $a \notin E$ and that E defends a in SF. This breaches the completeness of E. If $a \in att(A)$, then due to coherence part of defense in AFNs it has to be the case that $a \subseteq E$. However, from the construction of E' it means that $a \in E'$ and we reach a contradiction. Thus, $E \cup att(E)$ is complete in FN^{SF}.

Let $E' \subseteq A'$ be a complete extension of FN^{SF}. We can observe that in FN^{SF}, no attack argument is present without its normal arguments, and defending normal arguments leads to defense of the attack arguments derived from them. Consequently, it can be shown that every complete extension in FN^{SF} is of the form $S \cup att(S)$, where $S \subseteq A$. Let us now assume that $E = E' \cap A$ is not complete in SF. We know it is admissible, thus, it has to be the case that it defends an argument $a \notin E$. From the previous analysis it follows that if an argument is defended by E in SF, then it is defended by $E \cup att(E)$. Since E' is precisely $E \cup att(E)$, it has to be the case that E' defends a in FN^{SF}. As we have assumed that $a \notin E$, it holds that $a \notin E'$ and we contradict the completeness of E' in FN^{SF}. Consequently, $E' \cap A$ is complete in SF.

Let us now look at preferred semantics. Assume E is preferred in SF, but $E' = E \cup att(A)$ is not preferred in FN^{SF}. We know that E' is at least complete. By Theorem 2.95, it suffices to show that there is no AFN complete extension S s.t. $E' \subset S$. We know that $S \cap A$ is complete in SF by the previous parts of this proof and that $E \subseteq (S \cap A)$. As

E is preferred, it can only be the case that $E' \subset S$, but $E = S \cap A$. This means that there is an attack argument $b \in att(A)$ that is present in S, but not in E'. However, it should be clear from the previous parts of this proof that it cannot be the case and that it would contradict completeness, admissibility or construction of E'. Therefore E' is preferred in FN^{SF}.

Assume that $E' \subseteq A'$ is preferred in FN^{SF}, but $E = E' \cap A$ is not preferred in SF. By Theorems 2.24 and 2.95, this means there exists an complete extension S in SF s.t. $E \subseteq S$. However, since $S \cup att(S)$ is a complete extension of FN^{SF} and clearly $E' \subset (S \cup att(S))$, then E' could not have been preferred in the first place. We reach a contradiction. Consequently, E is preferred in SF.

Assume E is grounded in SF, but $E' = E \cup att(E)$ is not grounded in FN^{SF}. By Theorem 2.95, there must exist a complete extension S s.t. $S \subset E$. Since $S \cap A$ is also a complete extension of SF, then it has to be that $S \subset E'$, but $(S \cap A) = (E' \cap A)$. We can repeat the preferred analysis to show it cannot be the case and reach a contradiction. Consequently E' is the grounded extension of FN^{SF}.

The other direction of for grounded semantics is similar as in the preferred case. By Theorems 2.24 and 2.95, we know that the grounded extension is the least AFN/SETAF complete one. If E' is grounded in FN^{SF}, then $E \cap A$ is complete in SF. If it is not grounded in SF, then there exists a smaller complete extension, and following the proof for the preferred semantics we can show that in such a case we are able to construct a corresponding complete extension in FN^{SF} that would contain E'. This would contradict E' being grounded and thus we can conclude that $E' \cap A$ is the grounded extension of SF.

Finally, we come to the stable semantics. Assume E is stable in SF, but $E' = E \cup att(E)$ is not stable in FN^{SF}. By Theorem 2.24 we know that E is complete in SF, and therefore so is E' in FN^{SF}. Thus, it has to be the case that there exists an argument $X \in A' \setminus E'$ that is not in the deactivated set of E'. This means that X is not attacked and either is not supported by any set in N or receives sufficient support from E'. If $X \in A$, then it means it is not attacked by E' in FN^{SF}. However, then X could not have been attacked by E in SF as well and we reach a contradiction. If $X \in att(A)$, then it can only be the case it receives sufficient support from E'. However, this breaches the construction and completeness of E'. Thus, E' is stable in FN^{SF}.

By definition, every AFN stable extension is AFN complete. The same follows for SETAF by Theorem 2.24. Let us assume that E' is stable in FN^{SF}, but $E = E' \cap A$ is not stable in SF. This means there exists an argument $a \in A \setminus E$ s.t. there is no $S \subseteq E$, SRa. However, this implies that there is no $S \in E'$ s.t. $SR'a$, and since a requires no support in FN^{SF}, it could not have been in the deactivated set of E'. Consequently, E' could not have been stable in FN^{SF} in the first place and we reach a contradiction. □

Theorem 6.11. *Let $SF = (A, R)$ be a SETAF and $D^{SF} = (A, L, C)$ its corresponding ADF obtained through Translation 31. Then D^{SF} is an AADF$^+$ and a BADF. It is also cleansed and weakly valid. If SF is minimal, then D^{SF} is redundancy–free, relation and strongly valid.*

Proof. Almost every property follows from Lemma 4.77. We only need to show that if SF is minimal, then D^{SF} is redundancy–free, relation and strongly valid. Based on the proof of this Lemma, all links in D^{SF} are attacking. However, some might be supporting as well, which leads to redundancy. Let $X \subseteq A$ carry out a (minimal) attack on an argument $a \in A$ and let $x \in X$. Since X is minimal, then $X \subseteq \{x\}$ does not carry out an attack on a. Consequently, upon translating, $C_a(X \subseteq \{x\}) = in$ and $C_a(X) = out$. Hence, the (x, a) link is not supporting and thus is not redundant. We can repeat this analysis for any x and any attack in the framework and conclude that D^{SF} is redundancy–free when SF is minimal. Therefore, again by Lemma 4.77, it is relation and strongly valid. □

Theorem 6.12. *Let $SF = (A, R)$ be a SETAF and $D^{SF} = (A, L, C)$ its corresponding ADF obtained through Translation 31. A set of arguments E is a conflict–free extension of SF iff it is a conflict–free extension of D^{SF}.*

Proof. Assume E is conflict–free extension of SF, but not of D^{SF}. This means that there exists an argument $e \in E$ s.t. $\nexists E' \subseteq E$, $E'Re$, but $C_e(E \cap par(e)) = out$. However, by Translation 31, if $E \cap par(e)$ is mapped to out, then $\exists E' \subseteq E$ s.t. $E'Re$. We reach a contradiction.

Now assume E is conflict–free in D^{SF}, but not in SF. Hence, there is an argument $e \in E$ s.t. $C_e(E \cap par(e)) = in$, but $\exists E' \subseteq E$ s.t. $E'Re$. Again, by Translation 31 it is easy to see that it cannot be the case. □

Lemma 6.13. *Let $SF = (A, R)$ be a SETAF and $D^{SF} = (A, L, C)$ its corresponding ADF obtained through Translation 31. Let E be a conflict–free extension of SF (and thus of D^{SF}). The discarded set of E in SF coincides with the discarded set of E in D^{SF}.*

Proof. We will refer to the discarded set of E in SF with E^{att} in order to avoid confusion.

Let $a \in A$ be an argument in D^{SF}. We can observe that any minimal decisively in interpretation for a will have an empty $\mathbf{t}$ part and the $\mathbf{f}$ one will correspond to those (minimal) subsets $T \subseteq A$ s.t. $\forall S \subseteq A$, if SRa then $T \cap S \neq \emptyset$. We can thus construct trivial evaluations for a that will always be acyclic.

Let $a \in E^{att}$ be in the discarded set of SF. Therefore, $\exists E' \subseteq E$ s.t. $E'Ra$. Based on the previous explanations, we can observe that for any minimal decisively in interpretation v for a, $v^{\mathbf{f}} \cap E' \neq \emptyset$. Hence, any evaluation constructed for a will be blocked by E in D^{SF} and $E^{att} \subseteq E^+$.

By Proposition 2.150, the acceptance condition of any argument in E^+ in D^{SF} evaluates to out under E. And by construction, the acceptance condition of an argument is out w.r.t. E if $\exists E' \subseteq E$ attacking this argument in SF. Hence, whatever is in $E^+ \subseteq E^{att}$. We can therefore conclude that the discarded sets coincide. □

Lemma 6.14. *Let $SF = (A, R)$ be a SETAF and $D^{SF} = (A, L, C)$ its corresponding ADF obtained through Translation 31. A conflict–free set of arguments E defends an argument $a \in A$ in SF iff a is decisively in w.r.t. v_E in D^{SF}.*

Proof. We will refer to the discarded set of E in SF with E^{att} in order to avoid confusion.

Let $E \subseteq A$ be a conflict–free extension of SF. By Theorem 6.12, E is a conflict–free extension D^{SF} as well. Moreover, by Lemma 6.13, $E^{att} = E^{+}$. Assume that a is defended by E in SF, but is not decisively in w.r.t. the standard range v_E of E in D^{SF}. If a is not decisively in w.r.t. v_E, it means there exists a completion v' of v_E to $E \cup par(a)$ s.t. $C_a(v'^{\mathbf{t}} \cap par(a)) = out$. This means that $v'^{\mathbf{t}} \cap par(a)$ contains a set of arguments E' s.t. $E'Ra$. Since the set can be mapped to $\mathbf{t}$ in the completion, none of its members is mapped to **f** in v_E and thus none of them appears in E^{+}. Consequently, none of them is in E^{att} either. Therefore, E could not have defended a in SF. We reach a contradiction.

Let $E \subseteq A$ be conflict–free in D^{SF} and thus in SF. Assume that $a \in A$ is decisively in w.r.t. v_E, but is not defended by E. This means there exists a set of arguments B s.t. BRa and $B \cap E^{att} = \emptyset$. Consequently, there exists a set of arguments B s.t. $C_a(B) = out$ and $B \cap E^{+} = \emptyset$. If this is the case, then obviously a cannot be decisively in w.r.t. v_E and we reach a contradiction. □

Theorem 6.15. *Let $SF = (A, R)$ be a SETAF and $D^{SF} = (A, L, C)$ its corresponding ADF obtained through Translation 31. A set of arguments $E \subseteq A$ is a conflict–free extensions of SF iff it is (pd–acyclic) conflict–free in D^{F}. $E \subseteq A$ is a stable extensions of SF iff it is (stable) model of D^{F}. $E \subseteq A$ is a grounded extensions of SF iff it is (acyclic) grounded in D^{F}. $E \subseteq A$ is a σ extensions of SF, where where $\sigma \in \{$admissible, preferred, complete$\}$ iff it is an xy–σ–extension of D^{F} for $x, y \in \{a, c\}$.*

Proof. Due to the fact that the semantics classification collapses for D^{SF} (see Theorems 2.172 and 6.11), it suffices to focus only on the conflict–free, grounded, model, and the cc–types of the ADF semantics.

Conflict–freeness was already proved in Theorem 6.12. The fact that admissible extensions coincide follows straightforwardly from Theorem 6.12 and Lemma 6.14. Taking the ones maximal w.r.t. set inclusion obviously preserves it, thus preferred extensions correspond as well. Due to the correspondence between decisiveness and defense as seen in Lemma 6.14, complete extensions naturally coincide. By Theorem 2.24, the grounded extension of SF is the least w.r.t. set inclusion complete one. By Theorem 2.158, the grounded extension of D^{SF} is the least w.r.t. set inclusion cc–complete one. Therefore, the grounded extension is the same for both frameworks.

Let us finish with the analysis of stability. Assume E is stable in SF, but not in D^{SF}. This means that E is conflict–free in SF and $E^{att} = A \setminus E$. By Theorems 6.12, 2.172, 6.11 and Lemma 6.13, E is pd–acyclic conflict–free in D^{SF} and $E^{att} = E^{+}$. Hence, $E^{+} = A \setminus E$. All arguments in E^{+} are decisively out w.r.t. v_E, and thus there may be no argument $e \in E^{+}$ s.t. $C_e(E \cap par(e)) = in$. Therefore, the stability criterion in D^{SF} is satisfied.

Every ADF stable extension is a model, which is conflict–free in D^{SF} and thus also in SF. By Lemma 2.159 and Theorems 2.172, we have that $E^{+} = A \setminus E$ in D^{SF}. Thus, by Theorem 6.13, every argument in $A \setminus E$ is attacked by E.Thus, SETAF stability conditions are satisfied. □

15.5 Translating AFRAs: Proof Appendix

Theorem 7.2. *Let $FR = (A, R)$ be an AFRA and $F_m^{FR} = (A', R')$ its corresponding AF obtained through Translation 33. If $E \subseteq A \cup R$ is a σ–extension of FR, where $\sigma \in$ {conflict–free, complete, preferred, grounded, stable}, then $E' = E \cup \{x' \mid x \in (A \cap E^+)\}$ is a σ–extension of F_m^{FR}, where $E^+ = \{x \mid \exists y \in E$ s.t. y defeats $x\}$ is the discarded set of E in FR. This does not necessarily hold for admissible semantics. If $E' \subseteq A'$ is a σ'–extension of F_m^{FR}, where $\sigma' \in$ {admissible, complete, preferred, grounded, stable}, then $E = E' \cap (A \cup R)$ is a σ'–extension of FR. This does not necessarily hold for conflict–free semantics.*

Proof. Let $E \subseteq A \cup R$ be a conflict–free extension of FR. However, let us assume that $E' = E \cup \{x' \mid x \in (A \cap E^+\}$ is not, i.e. there are arguments $a, b \in E'$ s.t. $aR'b$ in F_m^{FR}. Let us assume that $a \in A \cap E'$. This means that b is a primed version of a. Since primed arguments are introduced into E' only for those arguments that are defeated by E and E cannot defeat any argument it contains due to conflict–freeness, we reach a contradiction. Let us assume that $a \in X' \cap E'$. This means that b is a conflict argument s.t. a is the source of b in FR. However, by the construction of E' it means that E defeats a. Thus, it defeats b as well and again we reach a contradiction with the conflict–freeness of E in FR. Let now $a \in R \cap E'$. It can be easily observed from the translation that if a attacks any argument b in E', then a corresponding conflict is present in E, and the set could not have been conflict–free in FR. Thus, we can finally conclude that if E is conflict–free in FR, then so is E' in F_m^{FR}.

Let $E' \subseteq A'$ be a conflict–free extension of F_m^{FR}. The set $E = E' \cap (A \cup R)$ is not necessarily a conflict–free extension of FR. This is due to the fact that indirect defeats in FR become indirect attacks in F_m^{FR} (i.e. the path between arguments is of odd length and bigger than 1). However, AF conflict–freeness only takes direct ones into account. Consider a simple Dung–style AFRA $(\{a, b, c\}, \{(a, b), (b, c)\})$ and its corresponding meta–level AF $(\{a, b, c, (a, b), (b, c), a', b', c'\}, \{(a, a'), (b, b'), (c, c'), (a', (a, b)), (b', (b, c)), ((a, b), b), ((b, c), c)\})$. We can observe that (a, b) does not attack (b, c) and thus those two arguments form a conflict–free set in the presented AF, even though (a, b) (indirectly) defeats (b, c) in the original AFRA.

Let us continue with the admissible semantics. Let $E \subseteq A \cup R$ be an admissible extension of FR. The set $E' = E \cup \{x' \mid x \in (A \cap E^+\}$ is not necessarily admissible in F_m^{FR}. We will use the same framework as in the conflict–freeness analysis. We can observe that the attack (a, b) forms an admissible extension of its own, as it is not defeated by any other conflict. However, $\{(a, b)\}$ is not an admissible extension of the corresponding meta–level AF; it is attacked by a'. Only $\{a, (a, b)\}$ is admissible, but it is not the set we have assumed in the original AFRA.

Let $E' \subseteq A'$ be an admissible extension of F_m^{FR}. Let us first assume that $E = E' \cap (A \cup R)$ is not conflict–free in FR. From the construction of F_m^{FR} we can observe that if an attack in E was directly defeating an argument or another attack in E, then the corresponding arguments would also be in conflict. We are only left with the option that an attack X in E is indirectly defeating another attack Y in E, i.e. $trg(X) = src(Y)$. However, in F_m^{FR}, the Y attack argument cannot appear in an admissible extension without its source argument, since $src(Y)$ defends Y against $src(Y)'$. Consequently, the arguments corresponding to X, Y and $src(Y)$ in F_m^{FR} are in E' and there is a conflict in R' between X and $src(Y)$. Thus, E' could not have been conflict–free in F_m^{FR} in the first place and we reach a contradiction. We have now established that E is conflict–free in FR; let us now assume that E is not admissible. This means that there is an attack $Z \in R$ and an element $W \in E$ s.t. Z defeats W and no attack $Y \in E$ s.t. Y defeats Z. First of all, this means that the attack argument Z is in conflict in R' either with the argument representing the source of W (if applicable) or the argument for W in F_m^{FR}. From the previous explanations we can observe that both W and its possible source need to be in E'. Thus, there is some conflict carried out against E' in F_m^{FR}. If there is no attack $Y \in E$ defeating Z, then neither the argument for Z nor the argument for its source are attacked by any conflict argument in E'. Moreover, if the argument for the source is not attacked, then its primed argument cannot be in E' due to the admissibility of E'. Consequently, the meta–argument for W is not defended by E' in F_m^{FR} and as it is contained in E', we reach a contradiction with admissibility of E'. Thus, we can conclude that if E' is admissible in F_m^{FR}, then so is E in FR.

Let $E \subseteq A \cup R$ be a complete extension of FR. Let us assume that $E' = E \cup \{x' \mid x \in (A \cap E^+\}$ is not complete in F_m^{FR}. By previous analysis, E' is admissible in F_m^{FR}. Therefore, it has to be the case that there is an argument $x \in A' \setminus E'$ that is defended by E' in F_m^{FR}. Let us assume that x is a standard argument $a \in A$. This means that every (y, a) argument in A' is attacked by E', i.e. either $(z, (y, a)) \in E'$ or $y' \in E'$. If it is the first case, then by the construction of E', $(z, (y, a)) \in E$ and a is defended by E. If it is the latter, then again by the construction of E', y is defeated by E and thus so is the (y, a) attack. Therefore, E defends a and as $a \notin E$, we reach a contradiction with the completeness of E in FR. Let us now assume that $x = a'$ for an argument $a \in A$. This means that a is attacked by E' and thus there is an argument (y, a) in E'. However, from this follows that $(y, a) \in E$ and that E defeats a. Thus, we reach a contradiction with the construction of E'. Finally, let $x = (a, j)$ for $a \in A$ and $j \in A \cup R$. As it is defended by E', then it is defended from a' and therefore it has to be the case that $a \in E'$. Moreover, for any attack argument $(z, (a, j))$, either $z' \in E'$ or $(v, (z, (a, j))) \in E'$. This means that first of all, $a \in E$ and (a, j) is defended by E from indirect defeats. If $z' \in E'$, then by the construction of E', E defeats z and thus indirectly defeats $(z, (a, j))$. If $(v, (z, (a, j))) \in E'$, then $(v, (z, (a, j))) \in E$ and E directly defeats $(z, (a, j))$. Therefore, x is defended by E even though $x \notin E$. We thus reach a contradiction with the completeness of E and can conclude that if E is complete in FR, then so is E' in F_m^{FR}. We can also observe that E' is the only complete extension of F_m^{FR} associated with E. Only the primed arguments are not directly defined by E. However, since we include a' for every $(x, a) \in E$, then every $(x, a) \in E'$ has the power to defend a' in F_m^{FR}. From this and the described proof we can observe that $\{x' \mid x \in (A \cap E^+\}$ is precisely the set of auxiliary arguments that a complete extensions associated with E can have.

Let now $E' \subseteq A'$ be a complete extension of F_m^{FR}. Let us assume that $E = E' \cap (A \cup R)$ is not complete in FR. By previous parts of this proof, E is admissible in FR. Therefore, it has to be the case that there is an element $x \in (A \cup R) \setminus E$ that is acceptable w.r.t. E. Firstly, we will consider the case in which x is an argument in A. As it is defended by E, then every attack (x, a) is directly or indirectly defeated by E, i.e. there exist either $(z, (x, a)) \in E$ or $(y, x) \in E$. Therefore, either $(z, (x, a)) \in E'$ or $(y, x) \in E'$. If it is the first case, then clearly (x, a) is attacked by E' and thus a is defended. If it is the latter, then by completeness, E' contains x'. As x' attacks (x, a) in R', then a is again defended by E'. We can also observe that these are the only possible attacks on a in F_m^{FR} and thus we reach a contradiction with the completeness of E'. Let us now consider the case in which x is an attack in R. If it is acceptable w.r.t. E, then every attack at x or at the source of x is defeated by E. By Lemma 2.32 we can observe that if x is acceptable w.r.t. E, then so is its source. Thus, from the previous explanations it holds that $src(x) \in E'$. Moreover, $src(x)$ is defended by E' due to admissibility and thus the attack argument for x is defended from $src(x)'$. Therefore, we need to focus only on direct attacks on x. From acceptability w.r.t. E it follows that for every attack (y, x), there is either an attack $(z, (y, x)) \in E$ or $(z, y) \in E$. If it is the first case, then the AF argument for $(z, (y, x))$ attacks the AF version of (y, x) and thus x is defended from attacks. If it is the latter, then $(z, y) \in E'$ and thus $y' \in E'$. Consequently, (y, x) is attacked by E' and again the attack argument for x is defended. Therefore, we reach a contradiction with the completeness of E' and can conclude that if E' is a complete extension of F_m^{FR}, then so is E of FR.

From previous parts of this proof it should be clear that there is one to one relation between the complete extensions of FR and F_m^{FR}. Furthermore, it holds that if $E_1 \subset E_2$ in FR, then $E_1' \subset E_2'$ in F_m^{FR}. Consequently, we can use Theorems 2.10 and 2.38 to show that the described relation for the preferred and grounded semantics is true. What remains to be shown is the relation between the stable extensions.

Let $E \subseteq A$ be a stable extension of FR and assume that $E' = E \cup \{x' \mid x \in (A \cap E^+)\}$ is not stable in F_m^{FR}. We know that E' is conflict–free in F_m^{FR} This means there is an argument $x \in A' \setminus E'$ that is not attacked by E'. If x is an argument $a \in A$, then it cannot be the case that there is an attack argument $(b, a) \in E'$. Thus, there is no $(b, a) \in E$ and a is not attacked by E. We reach a contradiction with the stability of E in FR. Let $x = a'$ for $a \in A$. If it is not attacked by E', then $a \notin E'$ and thus $a \notin E$. If $a \notin E$, then by stability of E, there is an attack $(b, a) \in E$. Consequently, E defeats a and we reach a contradiction with the construction of E'. Finally, let $x = (e, a)$ for $e \in A$, $a \in A \cup R$ be an attack argument. If it is not attacked by E', then $e' \notin E'$ and there is no attack argument $(f, (e, a))$ in E'. By the construction of E', if $e' \notin E'$, then $e \in E$. If there is no attack argument $(f, (e, a)) \in E'$, then there is no $(f, (e, a)) \in E$. Therefore, (e, a) is not defeated by E even though it is outside of the extension. We reach a contradiction with the stability of E in FR. Thus, if E is stable in FR, then so is E' in F_m^{FR}.

Let now $E' \subseteq A'$ be a stable extension of F_m^{FR}. Let us assume that $E = E' \cap (A \cup R)$ is not stable in FR. We know it is conflict–free in FR. This means there is an element $x \in (A \cup R) \setminus E$ that is not defeated by E, i.e. there is neither an attack (a, x) in E

nor $(a, src(x))$ in E in case $x \in R$. This means that $(a, x) \notin E'$ and $src(x)' \notin E'$. Consequently, E' cannot attack any argument representing x. We reach a contradiction with the stability of E' and can finally conclude that if E' is stable in F_m^{FR}, then so is E in FR. □

Theorem 7.3. *Let $FR = (A, R)$ be an AFRA and $BF^{FR} = (A', R', S)$ its corresponding BAF obtained through Translation 34. Let R^{sec} be the collection of first–tier secondary attacks in BF^{FR}. $E \subseteq A \cup R$ is a conflict–free (stable, d–grounded) extension of FR iff it is +conflict–free (stable, d–grounded) in BF^{FR} w.r.t. R^{sec}. E is a σ–extension of FR, where $\sigma \in$ {admissible, complete, preferred}, iff it is a d–σ–extension of FR w.r.t. (R^{sec}, R^{sec}).*

Proof. Let $a, b \in A \cup R$ be two elements in FR. We can observe that a directly defeats b in FR iff $(a, b) \in R'$ in BF^{FR}. If a indirectly defeats b in FR, then it directly defeats $src(b)$. Consequently, $(a, src(b)) \in R'$ and $(src(b), b) \in S$, and a secondary attacks b in BF^{FR}. Let now $a, b \in A'$ be arguments s.t. a secondary attacks b in BF^{FR}. We can observe that in our framework, an argument supporting any other argument cannot be supported itself. Thus, it suffices to focus on direct supports in our analysis. If a secondary attacks b, then there is an argument $c \in A$ s.t. $(c, b) \in S$ and $(a, c) \in R'$. By the construction of BF^{FR} this means that a directly defeats c and c is the source of b. Therefore, a indirectly defeats b in FR.

From this conflict analysis it follows easily that an element $a \in A \cup R$ is acceptable w.r.t. $E \subseteq A \cup R$ in FR iff E defends a w.r.t. R^{sec} in BF^{FR}. Consequently, the characteristic operator of FR and the d–characteristic operator of BF^{FR} w.r.t. R^{sec} coincide. Therefore, E is the grounded extension of FR iff it is the d–grounded extension of BF^{FR} w.r.t. R^{sec}.

The fact that the conflict–free and +conflict–free w.r.t. R^{sec} extensions coincide between the frameworks is a result of the correspondence between defeats and direct and indirect attacks in FR and BF^{FR}. Based on the defense analysis, it can be shown that the (d–) admissible, (d–) preferred and (d–) complete extensions w.r.t. R^{sec} also coincide in FR and BF^{FR}. From the conflict–freeness and attack analysis we can conclude that the stable extensions in both frameworks are the same as well. □

Theorem 7.6. *Let $FR = (A, R)$ be an AFRA and $FN^{FR} = (A', R', N')$ its corresponding AFN obtained through Translation 35. If a set $E \subseteq A \cup R$ is a σ–extension of FR, where $\sigma \in$ {conflict–free, complete, preferred, grounded, stable}, then it is a σ–extension of FN^{FR}. If $E = E^{src}$ is admissible in FR, then it is admissible in FN^{FR} and if it is conflict–free in FR, it is strongly coherent in FN^{FR}. It might not be the case for $E \neq E^{src}$.*

Not every conflict–free extension of FN^{FR} is conflict–free in FR. If a set $E' \subseteq A'$ is strongly coherent in FN^{FR}, then it is conflict–free in FR. If a set $E' \subseteq A'$ is a σ'–extension of FN^{FR}, where $\sigma' \in$ {admissible, complete, preferred, grounded, stable}, then E' is a σ'–extension of FR.

Proof. Let $E \subseteq A \cup R$ be a conflict–free extension of FR. This means there are no $V, W \in E$ s.t. V defeats W. If there are no two elements directly defeating each other, then by Translation 35 there are no two AFN arguments V, W s.t. $VR'W$. Thus, E is conflict–free in FN^{FR}. We can observe that E does not have to be coherent (and thus strongly coherent) in FN^{FR}. A set consisting of a single attack where the source and target are different will form a conflict–free set of its own. However, since the source of the attack is not present in the set, the coherence restrictions are not satisfied.

Let us now assume that $E^{src} \subseteq A \cup R$ is conflict–free in FR. We know that it is at least conflict–free in FN^{FR}. By Theorem 7.4, FN^{FR} is stronly valid. We can observe that if $a \in E \cap A$ in FR, then a is trivially coherent in E^{src} in FN^{FR}. If $a \in E \cap R$, then we know that $src(a) \in E^{src}$ and thus a is sufficiently supported through N'. Consequently, a is coherent in E^{src} in FN^{FR} and we can finally conclude that E^{src} is strongly coherent in FN^{FR}.

Let $E' \subseteq A'$ be conflict–free in FN^{FR}. We will show it does not have to be conflict–free in FR. Let us assume that $FR = (\{a, b, c\}, \{(a, b), (b, c)\})$ is a simple, AF–style AFRA. The corresponding AFN is $FN^{FR} = (\{a, b, c, (a, b), (b, c)\}, \{((a, b), b), ((b, c), c)\}, \{(\{a\}, (a, b)), (\{b\}, (b, c))\})$. The set $\{(a, b), (b, c)\}$ is conflict–free in FN^{FR}; the indirect attacks are not taken into account by the AFN conflict–freeness. However, indirect defeats do count in AFRA conflict–freeness, and (a, b) indirectly defeats (b, c) and the set is not conflict–free in FR.

Let $E' \subseteq A'$ be strongly coherent in FN^{FR}. Since it is conflict–free, there are no two arguments $V, W \in A'$ s.t. $VR'W$. By Translation 35 this means that there is no direct defeat in E' in FR. Now, due to coherence, for every $V \in E' \cap R$ it holds that $src(V) \in E'$. Therefore, by AFN conflict–freeness it cannot be the case that there exists $Y \in E'$ s.t. $YR'src(V)$ in FN^{FR}. Consequently, there can be no indirect defeats in E' in AFRA either, and E' is conflict–free in FR.

Having analyzed conflict–freeness and coherence, we can now move on to admissibility. Let $E \subseteq A \cup R$ be an admissible extension of FR. It does not have to be coherent in FN^{FR}. We can again assume that $FR = (\{a, b, c\}, \{(a, b), (b, c)\})$ and that its corresponding AFN is $FN^{FR} = (\{a, b, c, (a, b), (b, c)\}, \{((a, b), b), ((b, c), c)\}, \{(\{a\}, (a, b)), (\{b\}, (b, c))\})$. The extension $\{(a, b), c\}$ is admissible in FR. However, due to lack of a, it is not even coherent in FN^{FR}, let alone admissible.

Let $E^{src} \subseteq A \cup R$ be an admissible extension of FR. We know it is at least strongly coherent in FN^{FR}. Consequently, in order to show the defense, we need to prove that any coherent set $C \subseteq A'$ s.t. $\exists c \in C, a \in E^{src}, cR'a$, is in turn attacked by E^{src}. Let us assume it is not the case. Any AFN argument has a single minimal coherent set, namely $\{x\}$ for $x \in A$ and $\{src(x), x\}$ for $x \in R$. It is easy to see that if there exists an unattacked coherent set, then the minimal one cannot be attacked either. Assume that argument $a \in E^{src}$ is not defended by FN^{FR}. This means that it is attacked by some argument $x \in A'$ and neither x nor $src(x)$ (if applicable) are attacked. From the construction of FN^{FR} it follows that x directly defeats a and there is no argument (directly or indirectly) defeating

x in E^{src} in FR. Thus, a could not have been acceptable w.r.t. E^{src} in FR. We reach a contradiction with the assumptions. Hence, it has to be the case that E^{src} is admissible in FN^{FR}.

Let $E' \subseteq A'$ be admissible in FN^{FR}. Since it is coherent, it is easy to see that $E' = E'^{src}$. We know that E' is at least conflict–free in FR. What remains to be shown is that all arguments in E' are acceptable w.r.t. E' in FR. Let us assume this is not the case, i.e. $\exists a \in E', V \in R$ s.t. V defeats a and there is no $X \in E'$ s.t. X defeats V. If V directly defeats a, then by the construction of FN^{FR}, $VR'a$. If V defeats a indirectly, then $VR'src(a)$, which by the fact that $E' = E'^{src}$ means that E' is still attacked by V in FN^{FR}. Since there is no $X \in E'$ s.t. X defeats V, this means that $(X, V) \notin R'$ and $(X, src(V)) \notin R'$. Consequently, E' could not have attacked any coherent sets containing V in and thus could not have been admissible in FN^{FR} in the first place. We reach a contradiction. Therefore, E' has to be admissible in FR.

Let us now assume that $E \subseteq A \cup R$ is complete in FR. By Lemma 2.32 it is easy to see that $E = E^{src}$. This means that E is at least admissible in FN^{FR}. Let us now assume that it is not AFN complete; this means there exists an argument $a \notin E$ which is defended by E. Since a is defended, then either $a \in A$ or jointly $a \in R$ and $src(a) \in E$. Moreover, for any argument $V \in A'$ attacking a, E attacks $\{V\}$ or $\{V, src(V)\}$ if $V \in R$. Therefore, E defeats any direct defeater of a in FR. If $a \in A$, then it has only direct defeaters, and therefore E defends a in FR; we reach a contradiction with the completeness of E in FR. Let us thus assume that $a \in R$. Since $src(a) \in E$ and $src(a)$ is defended by E, then from the previous parts of the proof we can observe that this means that any element indirectly defeating a is also defeated. Consequently, a must have been acceptable w.r.t. E in FR and thus E could not have been complete in the first place. We reach a contradiction again. Therefore, we can conclude that E is complete in FN^{FR}.

Let us now assume that $E' \subseteq A \cup R$ is complete in FN^{FR}, but not in FR. We know it is at least admissible in FR. Moreover, since it is strongly coherent in FN^{FR}, it is easy to see that $E' = E'^{src}$. If E' is not AFRA complete, this means there exists an element $a \in (A \cup R) \setminus E'$ which is acceptable w.r.t. E'. Consequently, all elements defeating a are in turn defeated by E'. If $a \in A$, then for every $V \in (A \cup R)$ s.t. $a = trg(V)$, there exists $W \in E'$ s.t. $V = trg(W)$ or $src(V) = trg(W)$. Consequently, by the construction of FN^{FR}, it follows that $VR'a$ and either $WR'V$ or $WR'src(V)$ in FN^{FR}. Therefore, E' has the power to attack any coherent set for V, and as $\{a\} \cup E'$ is trivially coherent, then a must have been defended by E'. We reach a contradiction. Thus, let us focus on the case where $a \in R$. If a is acceptable w.r.t. E' in FR, then by Lemma 2.32, so is $src(a)$. If $src(a) \notin E'$, then we can repeat the previous analysis and reach a contradiction with our assumptions. If $src(a) \in E'$, then $\{a\} \cup E'$ is coherent. Moreover, since E' directly or indirectly defeats any direct attacks on a, then by all the analysis done above we can conclude that any coherent set attacking a is also in turn attacked. Consequently, a had to be defended by E' in FN^{FR} and E' could not have been complete in the first place. We reach a contradiction. Therefore, E' is complete in FR.

We have shown that complete extensions between FR and FN^{FR} coincide. By using

Theorems 2.38 and 2.95 we can conclude that also preferred and grounded extensions coincide. What remains to be shown is stability.

Let us assume that $E \subseteq A \cup R$ is stable in FR. We know it is at least complete in FN^{FR} by Theorem 2.38 and the previous parts of the proof. Moreover, E directly or indirectly defeats any element $V \notin E$ in FR. By the construction of FN^{FR}, this means that for every $V \notin E$, there exists $W \in E$ s.t. either $WR'V$ or $WR'src(V)$. If it is the first case, then naturally $V \in E^+$. If it is the latter, then due to conflict–freeness it has to be the case that $src(V) \notin E$. Consequently, $\exists Y \in A'$ s.t. $Y \notin E$ and $\{Y\}N'V$, which again means that $V \in E^+$. Therefore, E is stable in FN^{FR}.

Let us now assume that E is stable in FN. We know it is at least conflict–free in FR. Consider an argument $a \in E^+$. If there exists an argument $Y \in E$ s.t. $YR'a$, then Y directly defeats a in FR. If a is in the deactivated set due to lack of support, then it means that there exists $V \in A'$ s.t. $V \notin E$ and $\{V\}N'a$. Since $V = src(a)$, then $V \in A$. Moreover, as $V \notin E$ and there exists no set supporting V through N', then it has to be the case that E attacks V in R'. Thus, a is indirectly defeated by E in FR and we can conclude that E is AFRA stable. □

15.6 Translating EAFs and EAFCs: Proof Appendix

Theorem 8.2. *Let $EF = (A, R, D)$ be a bounded hierarchical EAF or an EAF s.t. there are no arguments $a, b \in A$ for which $(a, b), (b, a) \in R$ and EFC^{EF} its corresponding EAFC obtained trough Translation 36. A set $E \subseteq A$ is a σ–extension of EF, where $\sigma \in$ $\{$conflict–free, admissible, complete, preferred, grounded, stable$\}$, iff it is a σ–extension of EFC.*

Proof. This theorem is easily proved by Lemma 2.60. On the listed types of EAFs, the conflict–freeness definition can be replaced with a defeat based one. Consequently, the EAF and EAFC semantics are defined in the same manner and share their extensions. □

Theorem 8.4. *Let $EFC = (A, R, D)$ be a bounded hierarchical EAFC or an EAFC s.t. there are no arguments $a, b \in A$ for which $(a, b), (b, a) \in R$ and EF^{EFC} its corresponding EAFC obtained trough Translation 37. If $E \subseteq A$ is a σ–extension of EFC, where $\sigma \in$ $\{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ then $E' = E \cup \{(a, (b, c)) \mid (a, (b, c)) \in Gr_D, a \subseteq E\} \cup \{x' \mid E$ defeats$_E$ x and there is a reinstatement set for this defeat on $E\}$ is a σ–extension of EF^{EFC}. If $E' \subseteq A'$ is a σ'–extension of EF^{EFC}, where $\sigma' \in$ $\{$admissible, complete, preferred, grounded, stable$\}$ then $E = E' \cap A$ is a σ'–extension of EFC. This does not necessarily hold for conflict–free semantics.*

Proof. Let $E \subseteq A$ be a conflict–free extension of EFC. We will show that if E does not defeat$_E$ an argument $a \in A$ in EFC, then $E' = E \cup \{(a, (b, c)) \mid (a, (b, c)) \in Gr_D, a \subseteq E\} \cup \{x' \mid E$ defeats$_E$ x and there is a reinstatement set for this defeat on $E\}$ does not defeat$_{E'}$ an argument $a \in A$ in EF^{EFC}. Assume it is not the case, i.e. a is not defeated$_E$

by E, but is defeated$_{E'}$ by E'. By construction, $R' \cap (A \times A) = R$. Therefore, it can only be the case that there is an argument b in E (thus, one in E') attacking a, but while there is a defense attack for it carried out by $c \subseteq E$, there is none in E'. However, if $|c| = 1$, then $(c, (b, a)) \in D'$, and if $|c| > 1$, then $((c, (b, a)), (b, a)) \in D'$ and by the construction of E', $(c, (b, a)) \in E'$. In other words, a defense attack is present, and we reach a contradiction. Furthermore, we can observe that if a primed argument is included in E', then due to the conflict–freeness of E, the argument it represents cannot be in E' and neither any defense attack argument it takes part in carrying out. Thus, we can finally conclude that E' is conflict–free in EFC.

Let $E' \subseteq A'$ be a conflict–free extension of EF^{EFC}. To show that $E = E' \cap A$ is not necessarily conflict–free in EFC, let us look at the frameworks depicted in Figure 67. While $\{(\{a, b, c\}, (d, e)), d, e\}$ is a conflict–free extension of the corresponding EAF, $\{d, e\}$ is not conflict–free in the source EAFC.

Let $E \subseteq A$ be an admissible extension of EFC. The corresponding set $E' \subseteq A'$ is conflict–free in EF^{EFC}. We will show that if $a \in E$ defeats$_E$ an argument $b \in A$ and there is a reinstatement set for this defeat$_E$ in E, then a defeats$_{E'}$ b and there is a reinstatement set for this defeat$_{E'}$ in E'. First of all, if a defeats$_E$ b, then $(a, b) \in R$ and there is no $c \subseteq E$ s.t. $(c, (a, b)) \in D$. By the translation and the construction of E', this means that $(a, b) \in R'$ and there is either no $c \in E'$ s.t. $(c, (a, b)) \in D'$ or no $(c, (a, b)) \in E'$ s.t. $((c, (a, b)), (a, b)) \in D'$. Thus a defeats$_{E'}$ b. Let the set of pairs $R_E = \{(x_1, y_1), ..., (x_n, y_n)\}$ be the reinstatement set for the defeat$_E$ by a on b. We will transform it into a reinstatement set for this defeat in EF^{EFC}. We can observe that R_E is a reinstatement set for any of the defeats in the set. This means that for every y_i, its primed version (if it exists) is in E'. Let $(y', (d, (e, f))) \in R'$ be an arbitrary attack carried out by a primed argument. We can observe that by the construction of EF^{EFC}, it is never defense attacked. Thus, $\{(y', (d, (e, f)))\}$ is a trivial reinstatement set for this defeat. Let $P \subseteq R'$ be the collection of all attacks carried out by the primed arguments present in E'. We will now show that $R_E \cup P$ is a reinstatement set for the a–b defeat in E' in EF^{EFC}. First of all, based on the previous discussions, we can observe that every pair in the set is a defeat w.r.t. E'. Let now $(z_i, (x_i, y_i)) \in D'$ be a defense attack on any of the pairs in $R_E \cup P$. From the explanations on reinstatement sets for defeats by primed arguments we can see that it suffices to focus on R_E. If $(z_i, (x_i, y_i)) \in D \setminus Gr_D$, then due to the fact that R_E is a reinstatement set in EFC, R_E contains a suitable defeat on z_i. If $(z_i, (x_i, y_i)) \in Gr_D$, then by the construction of E' there is an argument $z \in z_i$ s.t. $z' \in E'$ and by the construction of EF^{EFC}, $(z', (z_i, (x_i, y_i))) \in R'$. Moreover, $(z', (z_i, (x_i, y_i))) \in P$. Therefore, a suitable defeat for the defense attack is present in $R_E \cup P$ again. Thus, $R_E \cup P$ is a reinstatement set for the defeat$_{E'}$ by a on b.

Based on this explanation it now holds that every argument $a \in E' \cap A$ is acceptable w.r.t. E'. Furthermore, based on the construction of E' and previous explanations, every argument $a \in E' \cap X'$ is also acceptable w.r.t. E'. For a given defense argument $(c, (b, a)) \in E' \cap Gr_D$ it is the case that $c \subseteq E'$. Since $(c, (b, a))$ is attacked only by arguments x' s.t. $x \in c$ and no $(x, x') \in R'$ attack can be defense attacked in EF^{EFC}, then

$\{(x, x')\}$ is a trivial reinstatement set for the x–x' defeat and $(c, (b, a))$ is acceptable w.r.t. E'. We can thus finally conclude that if E is admissible in EFC, then so is E' in EF^{EFC}.

Let $E' \subseteq A'$ be an admissible extension of EF^{EFC}. Let us assume that $E = E' \cap A$ is not conflict–free in EFC. Since $R \subseteq R'$, this can only be the case that there is an attack $(a, b) \in R$ for $a, b \in E$ that is defense attacked in D' by an element $c \in E'$, but not in D by any argument in E. If $c \in A$, then $c \in E$ and $(c, (a, b)) \in D$. Consequently, the (a, b) conflict is defense attacked in D by E. If $c = (d, (a, b)) \in Gr_D$, then by the admissibility of E' in EF^{EFC}, $d \subseteq E'$ and thus $d \subseteq E$. Therefore, we can conclude that if (a, b) is defense attacked in D' in E', then it is defense attacked in D in E. Thus, we reach a contradiction and E is conflict–free in EFC. Let $a \in E$ be an argument for which there exists an argument $b \in A$ s.t. b defeats$_E$ a. We can observe that b defeats$_{E'}$ a as well; first of all, $(b, a) \in R \subseteq R'$ and if E' contained a defense attacking argument, then the set carrying out the attack would have to (due to defense) be contained in E' and thus in E. By the admissibility of E', it means that there exists $c \in E'$ defeating$_{E'}$ b in a way that this defeat has a reinstatement set. Let $R_{E'}$ be a suitable reinstatement set. Based on the previous discussions, we can notice that any attack carried out by a primed argument $x' \in E'$ is a defeat and that this defeat forms a reinstatement set of its own. Thus, we can add it to $R_{E'}$, and $R_{E'}$ is still a reinstatement set. Moreover, x' is acceptable w.r.t. E', and thus x is in turn defeated by E' with reinstatement. Let us add this reinstatement set to $R_{E'}$ as well; again, $R_{E'}$ remains a reinstatement set. We will now continue analysis with the completed version of $R_{E'}$.

We can observe that if c defeats$_{E'}$ b and $b \in A$, then $c \in A$. Moreover, if c defeats$_{E'}$ b, then c defeats$_E$ b as well – what remains to be shown that this defeat is reinstated. We will now prove that $R_E = \{(x_1, y_1), ..., (x_n, y_n)\} = R_{E'} \cap R$ is a suitable reinstatement set for this defeat in EFC. Let us assume it is not the case; this means there exists a set $F \subseteq A$ s.t. $(F, (x_i, y_i)) \in D$ and no $(e, f) \in R_E$ for an argument $f \in F$. By the construction of EF^{EFC}, it means that either $|F| = 1$ and $(F, (x_i, y_i)) \in D'$, or $|F| > 1$ and $((F, (x_i, y_i)), (x_i, y_i)) \in D'$. If it is the first case, then there is a suitable $(e, F) \in R_{E'}$ and thus in R_E; we reach a contradiction. If it is the latter, then there is an argument $f \in F$ s.t. $f' \in E'$ and $(f', (F, (x_i, y_i))) \in R_{E'}$. Based on the modifications done do $R_{E'}$ and the way R' is constructed, we can observe that there is a defeat$_E$ $(g, f) \in R_{E'}$ and that $g, f \in A$. Therefore, $(g, f) \in R_E$ and as $f \in F$, the $(F, (x_i, y_i))$ defense attacked is dealt with by R_E and we reach a contradiction. Thus, R_E is a reinstatement for the defeat$_E$ by c on b, and hence a is acceptable w.r.t. E. We can now conclude that if E' is admissible in EF^{EFC}, then so is E in EFC.

Based on the relation between the admissible extensions of EFC and EF^{EFC} and the defeats and reinstatement sets, we can observe that if $E' \subseteq A'$ is complete in EF^{EFC}, then $E = E' \cap A$ is complete in EFC. Furthermore, if $E \subseteq A$ is complete in EFC, then $E' = E \cup \{(a, (b, c)) \mid (a, (b, c)) \in Gr_D, a \subseteq E\} \cup \{x' \mid E$ defeats$_E$ x and there is a reinstatement set for this defeat on $E\}$ is complete in EF^{EFC} – it is easy to show that no further Gr_D or X' are acceptable w.r.t. E' and none of them can be removed from E'. Finally, E' is the only extension that can be associated with E.

The proof for the preferred semantics follows easily from completeness. Since we are working with finite frameworks, the grounded extensions of both frameworks can be obtained by starting with the empty set and iteratively applying the characteristic operators. Let $E = E' = \emptyset$. We can observe that the same arguments are acceptable w.r.t. E in EFC as w.r.t. E' in EF^{EFC}. We can add them to the sets. Let us now repeat the operator for E', but consider only arguments in $A' \setminus A$. Clearly, we can add to E' any Gr_D argument s.t. the source of the defense attack is in E'. Furthermore, we need to add primed arguments acceptable w.r.t. E'. This means that the original argument it represents is defeated by E' with reinstatement, and thus it is also defeated by E with reinstatement. Therefore, right now we have two admissible extensions E and E' related in the way as stated in the theorem. Let us now apply the operator again. Just like previously the same arguments are acceptable w.r.t. E in EFC as w.r.t. E' in EF^{EFC}. We can add them to the sets and repeat the $A' \setminus A$ step for E'. We can repeat this line of reasoning till we obtain or grounded extensions and conclude that they are related the same way complete extensions are.

Let us now focus on stable semantics. Let $E \subseteq A$ be a stable extension of EFC and assume that $E' = E \cup \{(a,(b,c)) \mid (a,(b,c)) \in Gr_D, a \subseteq E\} \cup \{x' \mid E \text{ defeats}_E\ x$ and there is a reinstatement set for this defeat on $E\}$ is not stable in EF^{EFC}. Taking into account previous analysis, it means there is an argument $a \in (A' \setminus E')$ for which there is no argument $b \in E'$ s.t. b defeats$_E$ a. Since $R \subseteq R'$, it has to be the case that a is in fact the primed argument for b or that a is a Gr_D argument and for no argument carrying out the a defense attack, its primed version is in E'. Let us focus on the first case; since the (b, b') attack cannot be defense attacked, then it has to be the case that $b \notin E'$ and thus $b \notin E$. However, by stability of E, E would defeat$_E$ b and there would be a reinstatement set for this defeat. Therefore, by the construction of E', $b' \in E'$ and we reach a contradiction. Let us now focus on the Gr_D case. As the primed arguments are not in E', therefore their originals are not defeated by E, and thus have to be in E. Consequently, by the construction of E', a has to be in E' and we reach a contradiction. Thus, if E is stable in EFC, then E' is stable in EF^{EFC}.

Let now $E' \subseteq A$ be a stable extension of EF^{EFC}. The set $E = E' \cap A$ is complete in EFC and from the previous parts of this proof it holds that if E' defeats$_{E'}$ an argument $a \in A$, then so does E. Thus, E is stable in EFC. □

Theorem 8.11. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF and $SF^{EF} = (A', R')$ its corresponding SETAF obtained through Translation 40. If $E \subseteq A$ is a σ–extension of $bh - EF$, where $\sigma \in$ {conflict–free, admissible, complete, preferred, grounded, stable}, then $E' = E \cup \{x' \mid x' \in X', x \in E^+\}$ is a σ–extension of SF^{EF}. If $E' \subseteq A'$ is a σ'–extension of SF^{EF}, where $\sigma' \in$ {admissible, complete, preferred, grounded, stable}, then $E' \cap A$ is a σ'–extension of $bh - EF$. This does not necessarily hold for conflict–free semantics.*

Proof. Let $E \subseteq A$ be a conflict–free extension of $bh - EF$ and $E' = E \cup \{x' \mid x' \in X', x \in E^+\}$ a set of arguments in SF^{EF}. Assume E' is not conflict–free in SF^{EF}. This means there exists $F \subseteq E', a \in E'$ s.t. $(F, a) \in R'$. If $a = x' \in X'$ (i.e. a is a primed argument), then based on the construction of SF^{EF} it has to be the case that $F = \{x\} \subseteq E'$ and hence $x \in E$. However, this means that $E \cap E^+ \neq \emptyset$, which breaches the conflict–freeness of E in $bh - EF$. If $a \in A$, then it means that $F = datt'(x, a) \cup \{x\}$ for some argument $x \in A$ s.t. $(x, a) \in R$. Since $F \subseteq E'$ and $x \in A$, it holds that $x \in E$ and, based on the construction of E' and conflict–freeness of E, there is no $c \in A$ s.t. $(c, (x, a)) \in D$ in E. However, this means that x defeats$_E$ a and we reach a contradiction with our initial assumption that E is conflict–free in $bh - EF$.

Not every conflict–free set of SF^{EF} corresponds to a conflict–free set of $bh-EF$. We can look at Example 95. Although the set $\{a, b\}$ is not conflict–free in the source EAF, it is conflict–free in the target SETAF; only $\{a, b, c'\}$ is not conflict–free anymore.

Let $E \subseteq A$ be an admissible extension of $bh - EF$ and E' the corresponding conflict–free extension of SF^{EF}. Assume E' is not admissible. This means there is an argument $a \in E'$ that is not defended by E', i.e. there is a subset $F \subseteq A'$ s.t. $(F, x) \in R'$, but no $B \subseteq E'$ and $f \in F$ s.t. $(B, f) \in R'$. If $a = x' \in X'$, then F has to be of the form $F = \{x\}$ for an argument $x \in A$. Moreover, based on the construction of E', x is defeated$_E$ by E with reinstatement. Consequently, there has to be an argument $y \in E$ s.t. $(y, x) \in R$ and for any $c \in A$ s.t. $(c, (y, x)) \in D$, c is defeated$_E$ with reinstatement. This, by the construction of E', means that $y \in E'$ and for any $c, c' \in E'$. Hence, there is a set $B = datt'(y, x) \cup \{y\} \subseteq E'$ s.t. $(B, x) \in R'$. Therefore, E' can attack F and defend a. We reach a contradiction with our assumptions.

Let us now assume that $a \in A$ is a standard argument in E' that is not defended by E'. From the fact that $(F, a) \in R'$ and $a \in A$ it follows that F represents a set conflict induced by an attacker of a in R. Let thus $F = \{f_{att}, f'_1, ..., f'_k\}$ be the form of F, where $(f_{att}, a) \in R$. We can distinguish two cases; one in which f_{att} defeats$_E$ a and one in which it does not. If f_{att} defeats$_E$ a, then based on the admissibility of E in $bh - EF$, there has to be an argument $b \in E$ s.t. b defeats$_E$ f_{att} with reinstatement. Hence, E can defeat$_E$ with reinstatement also any argument defense attacking the (b, f_{att}) conflict. We can therefore use the analysis performed in the previous step to show that there is a set $B = datt'(b, f_{att}) \cup \{b\} \subseteq E'$ s.t. $(B, f_{att}) \in R'$. Hence, E' can defend a in this case. If f_{att} does not defeat$_E$ a in E, then there is an argument $c \in E$ s.t. $(c, (f_att, a)) \in D)$. Consequently, there is an argument $f'_i \in F$ s.t. $(\{c\}, f'_i) \in R'$. Again, E' can defend a against F. We reach a contradiction with our assumptions. We can therefore conclude that if E is admissible in $bh - EF$, then E' is admissible in SF^{EF}.

Let $E' \subseteq A'$ be an admissible extension of SF^{EF} and $E = E' \cap A$ a set of arguments in $bh - EF$. Assume that E is not conflict–free; this means there are arguments $a, b \in E$ s.t. a defeats$_E$ b. Consequently, there is also no $c \in E$ s.t. $(c, (a, b)) \in D$. Based on the construction of SF^{EF} there has to exist a set of arguments $F \subseteq A'$ s.t. $a \in F$ and $(F, b) \in R'$. If $|F| = 1$ (i.e. there was no defense attack for (a, b) in $bh - EF$ at all), then we breach the conflict–freeness of E' in SF^{EF}. If $|F| > 1$, then we can observe that $a \in E$ and due to the fact that there is no $c \in E$ s.t. $(c, (a, b)) \in D$, then no such c is in E' and the respective c' is not in E'^+. This means that no argument in F is attacked by E'

and b cannot be defended by E'. We breach the admissibility of E'. Therefore, it has to be the case that E is conflict–free in $bh - EF$.

Let us now assume that E it is not admissible in $bh - EF$, i.e. there are arguments $a \in E$, $b \in A$ s.t. b defeats$_E$ a, but no argument $c \in E$ defeats$_E$ b with reinstatement. We can observe that there will be an attacking set in SF^{EF} corresponding to the (b, a) attack and the $datt'(b, a)$ arguments. Thus, there is an attacker of a in R' that needs to be dealt with by E'.

Let us focus on the case where b defeats$_E$ a, but no $c \in E$ defeats$_E$ b. This means that either no c attacks b in R, or for every such attack there is a defense attacker present in E. If it is the first case, then there is no suitable attack in E' either. If the latter, then no primed argument for the defense attackers can be in E', and thus the set of arguments corresponding to c that attacks b in R' is not contained in E'. Therefore, b is not attacked by E'. Furthermore, as b defeats$_E$ a, then no defense attacker of the (b, a) conflict is in E. Thus, it is also not in E', and no primed argument in $datt'(b, a)$ can be attacked by E'. We can therefore conclude that E' does not defend a and we breach the admissibility of E'.

We can now focus on the case where there is an argument c defeating$_E$ b, but the defeat is not reinstated. According to Theorem 2.46, there is a sequence $((z_1, (x_1, y_1)), ..., (z_n, (x_n, y_n)))$ of distinct defense attacks from D s.t. $(x_n, y_n) = (c, b)$ s.t. for every $(z_i, (x_i, y_i))$ where $1 < i \leq n$, either no argument h in E defeats$_E$ z_i or for every such defeat, there exists an argument $l \in A$ s.t. $(l, (h, z_i)) \in \{(z_1, (x_1, y_1)), ..., (z_{i-1}, (x_{i-1}, y_{i-1}))\}$, and no argument in E defeats$_E$ z_1. If no argument $d \in E$ defeats$_E$ z_i, then either d does not attack z_i or for every such attack, a defense attacker e is present in E. This means that $d \notin E'$ or $e' \notin E'$, and thus we can show that no set of elements in SF^{EF} is fully contained in E'. Therefore, E' does not attack z_1, and thus the primed version of this argument (if it exists) is not in E'. Let us move on to z_2; if E does not defeat$_E$, then we can repeat the previous analysis and conclude that E' does not attack z_2 and cannot contain z'_2. If E defeats$_E$ z_2 with an argument $x \in E$, then by the construction of the sequence, this attack is defense attacked by z_1. Consequently, although $x \in E'$, $z'_1 \notin E'$, and E' cannot attack z_2 with the set corresponding to x. Thus, we can show that E' does not attack z_2 and thus $z'_2 \notin E'$. We can continue in this manner until we reach z_n and the conclusion that z_n is not attacked by E' and $z'_n \notin E'$. Therefore, we can see that $y_n = b$ is not attacked by E'. Moreover, from the previous analysis we can observe that no argument in $datt'(b, a)$ is attacked by E' either. Consequently, it cannot be the case that E' defends a and we reach a contradiction with the admissibility of E'. We can finally conclude that if E' is admissible in SF^{EF}, then E is admissible in $bh - EF$.

Let $E \subseteq A$ be a complete extension of $bh - EF$ and $E' = E \cup \{x' \mid x' \in X', x \in E^+\}$ the corresponding admissible extension of SF^{EF}. Let us assume E' is not complete; this means there is an argument $a \in A' \setminus E'$ that is defended by E'. Let us first consider the case where a is a standard argument, i.e. $a \in A$. Therefore, every set of arguments $B \subseteq A'$ attacking a has an element $b \in B$ attacked by some set $F \subseteq E'$. Let $b = x' \in X'$ be a primed argument. This means that $F = \{x\}$ and $x \in E$. Consequently, the attack on a for which B was created is defense attacked by a member of E and does not become a defeat and does not have to be defended against in $bh - EF$. Let $b \in A$ be a standard argument. In this case, F contains an argument attacking b in R and a primed argument for every defense attacker of this conflict. By the construction of E' this means that E contains an attacker of b and that every defense attacker for this conflict is defeated$_E$ by E with a reinstatement. Consequently, E defeats$_E$ b with reinstatement. Therefore, it can be shown that if E' defends a standard argument $a \notin E'$, then E defends an argument $a \notin E$. We breach the completeness of E in $bh - EF$.

Let $a = x' \in X'$ be now a primed argument defended by E' but not contained in E'. If it is defended by E', then it has to be the case that argument x is attacked by a subset $F \subseteq E'$. However, we can repeat the previous analysis to show that E defeats$_E$ x with reinstatement. This means that x' has to be in E' and we breach the construction of E'. We can finally conclude that if E is complete in $bh - EF$, then E' is complete in SF^{EF}.

We can also observe that this is the only complete extension of SF^{EF} associated with E. The only freedom we have in forming E' concerns the primed arguments. Let us assume that E'_1 and E'_2 are two different complete extensions of SF^{EF} s.t. $E'_1 \cap A = E'_2 \cap A$, i.e. they only differ by X' arguments. Let us assume $a' \in E'_1$, but not in E'_2. This means that E'_1 attacks a and E'_2 does not. Due to the fact that $E'_1 \cap A = E'_2 \cap A$, this can only mean that every attacking set of a contains a primed argument that is not present in E'_2, but at least one such full set is in E'_1. Let b' be such a primed argument. It can only be attacked b; again, it means that E'_1 can attack b, while E'_2 cannot. We can observe that due to the fact that the source EAF is bounded hierarchical, $a \neq b$. We can again find a primed argument present in E'_1, but not in E'_2, that is contained in a set attacking b, and note that its origin has to be different from a and b. We can continue in this manner until we find a standard argument that is only attacked by standard arguments. We are bound to reach this spot due to the fact that the source EAF is bounded hierarchical. However, we can then see that as both E'_1 and E'_2 share the same set of standard arguments, they are both capable of attacking this argument. We thus a reach a contradiction and it can be shown that $E'_1 = E'_2$.

Let $E' \subseteq A$ be a complete extension of SF^{EF}. From the previous analysis we can conclude that it is precisely of the form $E \cup \{x' \mid x' \in X', x \in E^+\}$, where E is an admissible extension of $bh - EF$. From the completeness of E' in SF^{EF} it follows that for every argument $a \in A' \setminus E'$ there is a set $F \subseteq A'$ attacking it that does not contain any argument attacked by E' (i.e. no argument outside of E' is defended by E'). We will now show that it also means that there is no argument outside of E that is defended by E in $bh - EF$. Let $a \in A$ be an argument and let F be of the form $\{f_{att}, f'_1, ..., f'_n\}$ (it is possible that $n = 0$). If no primed argument f'_i is attacked by E', then it has to be the case that $f_i \notin E'$. This means that no argument in E defense attacks the (f_{att}, a) conflict and f_{att} defeats$_E$ a. Consequently, there is no unattacked argument in A that is not in E. Based on the completeness of E', there is no set $B \subseteq E'$ attacking f_{att}. If no such set exists in A' at all, then clearly f_{att} is not attacked in R at all either and thus cannot be defeated$_E$ by E. If such a set exists, then it has to be of the form $\{b_{att}, b'_1, ..., b'_j\}$ (it is possible that $j = 0$) and at least one of its elements is not in E'. If $b_{att} \notin E'$, then clearly E cannot defeat$_E$ f_{att} (through b_{att} at least). If $b_{att} \in E'$ but $b'_i \notin E$, then based on the construction of E', E does not defeat b_i with reinstatement. Therefore, E cannot be defeating$_E$ f_{att} with reinstatement and a cannot be acceptable w.r.t. E. We can conclude that for every argument outside of E there is an argument defeating$_E$ it and that this argument is not defeated$_E$ with reinstatement by E. Hence, no argument outside of E is defended by E and the set is complete in $bh - EF$.

We can observe that there is one–to–one relation between the complete extensions of SF^{EF} and $bh - EF$. Moreover, it is easy to show that if $E_1 \subseteq E_2$ are two complete extension of $bh - EF$, then the corresponding E'_1 and E'_2 extensions SF^{EF} are also of the form $E'_1 \subseteq E'_2$. Therefore, the preferred extensions of $bh - EF$ and SF^{EF} are related in the same the complete ones. By Theorems 2.24 and 2.58, the same holds for grounded extensions as well.

Let $E \subseteq A$ be a stable extension of $bh - EF$ and $E' = E \cup \{x' \mid x' \in X', x \in E^+\}$ the corresponding conflict–free set in SF^{EF}. We can observe that E defeats$_E$ every argument $a \in A \setminus E$. It can be easily shown that due to the fact that every argument outside E is defeated, the collection of all defeats carried out by E forms a reinstatement set for them. Therefore, for every defeat$_E$ by an argument $b \in E$ on an argument $a \in A \setminus E$, if there is an argument $c \in A$ s.t. $(c, (b, a)) \in D$, then $c \in E^+$. Therefore, $c' \in E'$, and by collecting such primed arguments and b we obtain a subset of E' attacking a in R'. Consequently, every standard argument outside of E' is attacked by E' in SF^{EF}. Moreover, we can notice that if $x' \in A' \setminus E'$ is a primed argument outside of E', then by the construction of E' and stability of E, $x \in E'$. Thus, x' is attacked by E'. We can conclude that E' is a stable extension of SF^{EF}.

Let $E' \subseteq A'$ be a stable extension of SF^{EF}. By Theorem 2.24 it is a preferred extension of SF^{EF}, and thus $E = E' \cap A$ is a conflict–free set of $bh - EF$. It holds that for every argument $a \in A' \setminus E'$, there is a subset $F \subseteq E'$ s.t. $(F, a) \in R'$. Let us limit ourselves to a being a standard argument and let F be of the form $\{f_{att}, f'_1, ..., f'_n\}$. From the construction of SF^{EF} and E it holds that $(f_{att}, a) \in R$ and $f_{att} \in E$. Moreover, by conflict–freeness of E', for no f'_i, $f_i \notin E'$ and thus $f_i \notin E$. Therefore, no defense attacker of the (f_{att}, a) conflict is present in E, and f_{att} defeats$_E$ a. We can thus show easily that E is a stable extension of $bh - EF$. □

Theorem 8.15. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF and $FR^{EF} = (A, R \cup D)$ its corresponding AFRA obtained through Translation 42. If $E \subseteq A$ is a σ–extension of $bh - EF$, where $\sigma \in \{$conflict–free, admissible, complete, grounded, preferred, stable$\}$, then there is a σ–extension $E' \subseteq (A \cup R \cup D)$ of FR^{EF} s.t. $E = E' \cap A$. If $E' \subseteq A \cup R \cup D$ is a σ'–extension of FR^{EF}, where $\sigma' \in \{$complete, grounded, preferred, stable$\}$, then $E = E' \cap A$ is a σ'–extension of $bh - EF$. This does not necessarily hold for conflict–free and admissible semantics.*

Proof. Let $E \subseteq A$ be a conflict–free extension of $bh - EF$. Clearly, E is conflict–free in FR^{EF} – after all, $E \cap (R \cup D) = \emptyset$. Consequently, every conflict–free extension of $bh - EF$ will be conflict–free in FR^{EF}. However, for the same reasons, it will not hold the other way around – due to the way AFRAs work, every set of arguments is conflict–free in FR^{EF}, while it is clearly does not have to be the case in $bh-EF$.

By Translation 38, $F=(A',R')$ is an AF corresponding to $bh-EF$, where $A'=A\cup R\cup D\cup X'$ for $X'=\{x'\mid x\in A\}$ and $R'=\{(x,x')\mid x\in A\}\cup\{(x',a)\mid a\in R\cup D, src(a)=x\}\cup\{(a,b)\mid trg(a)=b$, and either $(a\in R$ and $b\in A)$, or $(a\in D$ and $b\in R)\}$. We can observe that FR^{EF} translated to an AF with the use of Translation 33 will be the same. We can thus use this to prove the correspondence between $bh-EF$ and FR^{EF} in our proof.

Let $\sigma\in\{$admissible, complete, grounded, stable, preferred$\}$ be a semantics. By Theorem 8.6, if $E\subseteq A$ is a σ–extension of $bh-EF$, then there is a set $E'\subseteq A'$ s.t. $E=E'\cap A$ is a σ–extension of F. By Theorem 7.2 if $E'\subseteq A'$ is a σ–extension of F, then $E'\cap(A\cup R\cup D)$ is a σ–extension of FR^{EF}. Thus, we can conclude that if $E\subseteq A$ is a σ–extension of $bh-EF$, then there is an σ–extension $E'\subseteq A\cup R\cup D$ of FR^{EF} s.t. $E'\cap A=E$.

Not every admissible extension of FR^{EF} represents an admissible extension of $bh-EF$. We can consider a simple, AF–style framework $(\{a,b,c\},\{(a,b),(b,c)\})$. If we take AFRA semantics, $\{(a,b),c\}$ is admissible. However, $\{c\}$ is not admissible in the EAF case.

Let now $\sigma\in\{$complete, grounded, stable, preferred$\}$ be a semantics. By Theorem 7.2 if $E\subseteq A\cup R\cup D$ is a σ–extension of FR^{EF}, then $E'=E\cup\{x'\mid x\in(A\cap E^+)\}$ is a σ–extension of F. By Theorem 8.6, if $E'\subseteq A'$ is a σ–extension of F, then $E''=E'\cap A$ is a σ–extension of $bh-EF$. Thus, we can conclude that if $E\subseteq(A\cup R\cup D)$ is a σ–extension of FR^{EF}, then $E\cap A$ is an σ–extension of $bh-EF$.

As a final note we can observe that for complete, grounded, stable and preferred semantics, the relation between $bh-EF$ and FR^{EF} extensions is one–to–one. □

Theorem 8.18. *Let $bh-EF=(A,R,D)$ be a bounded hierarchical EAF and $FN^{EF}=(A',R',N')$ its corresponding AFN obtained through Translation 43. FN^{EF} is (strongly) consistent, minimal, weakly, relation and strongly valid.*

Proof. We can observe that only arguments $a\in A$ can receive support in FN^{EF}. Moreover, by construction, every supporting set contains a unique primed argument related to an attacker of a. Consequently, the support sets of a are incomparable and FN^{EF} is in minimal form.

Since $bh-EF$ is bounded hierarchical, according to Definition 2.55 it has a partition $(((A_1,R_1),D_1),...,((A_k,R_k),D_k)$ s.t. $A=\bigcup_{i=1}^k A_i$, $R=\bigcup_{i=1}^k R_i$, $D=\bigcup_{i=1}^k D_i$, for every $i=1...k$ (A_i,R_i) is a Dung's framework, and $(c,(a,b))\in D_i$ implies $(a,b)\in R_i$, $c\in A_{i+1}$, and $D_k=\emptyset$.

Let us now show that FN^{EF} is strongly valid. According to Definition 4.30, FN^{EF} is strongly valid iff for every function $f:A'\to\{S\mid a\in A', S\in suf(a)\}$ where $suf(a)=\{S\mid S\subseteq sup(a)$ and $\forall C\subseteq$ s.t. $CN'a$, $C\cap S\neq\emptyset\}$ and $sup(a)=\bigcup_{C\subseteq A,CN'a}C$, we can create a powerful sequence $(a_0,...,a_n)$ consisting of all elements of A' s.t. $f(a_i)\subseteq\{a_0,...,a_{i-1}\}$ for $1\leq i\leq n$ and $f(a_0)=\emptyset$. Let f be such a function.

Before we start, let us notice that by construction no primed argument in X' requires support in N'. Thus, $f(x) = \emptyset$ for an argument in $x \in X'$. Thus, any ordering on X' is a powerful sequence. Let us pick any and denote it $seq(X')$.

Let us now focus on $((A_k, R_k), D_k)$. Since $D_k = \emptyset$, then no attack in R_k is defense attacked and argument in A_k requires support. Consequently, $f(a) = \emptyset$ for $a \in A_k$. Thus, any ordering on A_k is a powerful sequence. Let us pick any and denote it $seq(A_k)$. We can observe that the sequence created by appending $seq(A_k)$ to $seq(X')$ is also powerful.

Let us now focus on $((A_{k-1}, R_{k-1}), D_{k-1})$. By the construction of the partition, any attack in R_{k-1} can be defense attacked only by D_{k-1} arguments and $D_{k-1} \subseteq A_k$. By the construction of FN^{EF}, every supporting set of an argument $a \in A_{k-1}$ is a subset of $D_{k-1} \cup X'$ and thus of $A_k \cup X'$. In other words, it has to be the case that $f(a) \subseteq A_k \cup X'$. Let $seq(A_{k-1})$ be an arbitrary ordering of elements of A_{k-1}. We can observe that appending $seq(A_{k-1})$ to the joined $seq(X')$ and $seq(A_k)$ sequence is again a powerful sequence.

We can repeat this procedure until we reach $((A_1, R_1), D_1)$ and that sequence built from joining (in this order) $seq(X')$, $seq(A_k)$, $seq(A_{k-1})$,...,$seq(A_1)$ is again a powerful sequence. We have thus created a powerful sequence from elements of A' using an arbitrary f function. We can conclude that FN^{EF} has to be strongly valid, and thus by Theorem 4.31 weakly and relation valid as well.

Since $bh - EF$ is (strongly) consistent (see Lemma 4.73), there are no arguments $b, c \in A$ s.t. $(b, a) \in R$ and $(b, (c, a)) \in D$ for an argument $a \in A$. As primed arguments cannot attack anything, we can conclude that the sets of attackers and supporters of a in FN^{EF} are disjoint and thus the framework is strongly consistent. □

Theorem 8.19. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF and $FN^{EF} = (A', R', N')$ its corresponding AFN obtained through Translation 43. If $E \subseteq A$ is a conflict–free extension of $bh - EF$, then $E' = E \cup \{x' \mid x' \in X, x \in A \setminus E\}$ is strongly coherent in FN^{EF}. If $E \subseteq A$ is a σ–extension of $bh - EF$, where $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$, then $E' = E \cup \{x' \mid x' \in X, x \in E^+\}$ is a σ–extension of FN^{EF}. If $E' \subseteq A'$ is a σ'–extension of FN^{EF}, where $\sigma' \in \{$admissible, complete, preferred, grounded, stable$\}$, then $E' \cap A$ is a σ'–extension of $bh - EF$. This does not necessarily hold for conflict–free semantics.*

Proof. Let $E \subseteq A$ be a conflict–free extension of $bh - EF$. This means that for every direct attack in E, there is a defense attack. Consequently, $((E \times E) \cap R) \subset R_D$. Since $R_D \cap R' = \emptyset$, then it cannot be the case that there is a conflict in E in FN^{EF}. Thus, E is conflict–free in FN^{EF}.

Let $E \subseteq A$ be a conflict–free extension of $bh - EF$ and let us define a set $E' = E \cup \{x' \mid x' \in X, x \in A \setminus E\}$. Assume that E' is not strongly coherent in FN^{EF}. Since E is conflict–free, no argument in X' carries out an attack in R'. Moreover, primed arguments are included only if their origins (and the only attackers) are not. Therefore, E' is conflict–free in FN^{EF}. By Theorem 8.18, FN^{EAF} is strongly valid. Consequently, in order to prove coherence, it suffices to show that for every argument $a \in E'$ and set $B \subseteq A'$ s.t.$BN'a$, $B \cap E' \neq \emptyset$. If a is a primed argument, then it requires support, and thus the condition is satisfied. Let now $a \in E$. If it is not attacked in R in $bh - EF$ or none of its attacks is defense attacked, then a requires no support in N' in FN^{EF} and thus the condition is again satisfied. Let us assume it is not the case, i.e. there is an attack $(b, a) \in R_D$. Therefore, there exists a support set $B \subseteq A'$ for a consisting of b' and every argument $c \in A$ s.t. $(c, (b, a)) \in D$. If $b \notin E$, then $b' \in E'$ and thus $B \cap E' \neq \emptyset$. If $b \in E$, then due to conflict–freeness of E, there has to be an argument $c \in E$ s.t. $(c, (b, a)) \in D$. Thus, $c \in E'$, and again $B \cap E' \neq \emptyset$. Therefore, E' is a strongly coherent set of FN^{EF}.

Let $E' \subseteq A'$ be a conflict–free extension of FN^{EF}. The set $E = E' \cap A$ is not necessarily conflict–free in $bh - EF$. We can consider the frameworks described in Example 99 – although $\{a, b\}$ is not a conflict–free extension of the source EAF, it is conflict–free in the target AFN.

Let $E' \subseteq A'$ be a strongly coherent extension of FN^{EF}. Assume $E = E' \cap A$ is not conflict–free in $bh - EF$. This means there are arguments $a, b \in E$ s.t. $(a, b) \in R$ and $\nexists c \in E, (c, (a, b)) \in D$. Since $(a, b) \notin R'$, then by the construction of FN^{EF} it means that $(a, b) \in R_D$. Thus, $a' \in A'$ and there is a set $Y \subseteq A'$ s.t $a' \in Y$ and $YN'b$. Due to conflict–freeness, $a' \notin E'$, and as E' is strongly coherent, then there is an argument $c \in E' \cap A$ s.t. $c \in Y$. By construction, $(c, (a, b)) \in D$. Therefore, there is in fact an argument defense attacking (a, b) in E and E has to be conflict–free in $bh - EF$.

Let $E \subseteq A$ be an admissible extension of $bh - EF$. We define $E' = E \cup \{x' \mid x' \in X, x \in E^+\}$. Due to conflict–freeness of E, $E \cap E^+ = \emptyset$ and $\{x' \mid x' \in X, x \in E^+\} \subseteq \{x' \mid x' \in X, x \in A \setminus E\}$. Thus, from previous explanations it is clear that E' is conflict–free in FN^{EF}. We now need to show it is coherent. Again, by Theorem 8.18, it suffices to show that for every argument $a \in E'$ and set $B \subseteq A'$ s.t. $BN'a$, $B \cap E' \neq \emptyset$. Based on previous explanations, we only need to focus on arguments a s.t. there is an attack $(b, a) \in R_D$. Associated with every such attack there is a support set $B \subseteq A'$ for a consisting of b' and every argument $c \in A$ s.t. $(c, (b, a)) \in D$. If b does not defeat$_E$ a, then there has to be an argument $c \in E$ s.t. $(c, (b, a)) \in D$. Thus, $c \in E'$ and $B \cap E' \neq \emptyset$. Consequently, the condition is satisfied. If b defeats$_E$ a, then due to admissibility of E, there is an argument $d \in E$ that defeats$_E$ b and there is a reinstatement set on E for this defeat. Consequently, by the construction of E', $b' \in E'$ and again $B \cap E' \neq \emptyset$. Therefore, E' is strongly coherent in FN^{EF}.

Now, we need to show that every argument $a \in E'$ is defended by E' in FN^{EF}. Since E' is coherent, it suffices to focus on the attack part of defense. Let us assume that not every argument is defended and thus there exists an argument $b \in A'$ s.t. $(b, a) \in R'$ and not every coherent set B for b is attacked by E. This gives us two options as to the form of (b, a); either $a = b'$ or $(b, a) \in R \setminus R_D$. If $a = b'$, then by the construction of E', E defeats$_E$ b in $bh - EF$ and there is a reinstatement set for such a defeat. If the latter, then b defeats$_E$ a (note there is no defense attack on it) and thus again needs to be defeated by E in a way there exists an appropriate reinstatement set. Let d be an argument in E s.t. d defeats$_E$ b and let $\{(x_1, y_1), ..., (x_n, y_n)\}$ be a reinstatement set on E for this defeat. If $(d, b) \in R \setminus R_D$, then $(d, b) \in R'$ and as $d \in E$, then $d \in E'$. Direct attack on b means that every coherent set for b is attacked and thus E is capable of defense; we reach a contradiction. If $(d, b) \in R_D$, then we undergo a support transformation. Let $C \subseteq \{y_1, ..., y_n\}$ be the set of all and only arguments that defense attack (d, b). Consequently, $C \cup \{d'\}$ is a support set for b. Since $d \in E'$ and $(d, d') \in R'$, the coherent sets for b built with d' are attacked. We now need to iterate through C. Since d possesses a reinstatement set on E, then it means that for every argument in $y_i \in C$ there is an argument $x_i \in E$ attacking it. If this conflict is in $R \setminus R_D$, then it also appears in R' and another coherent set of b is dealt with. If the conflict is in R_D, then it undergoes a support transformation yet again. We can repeat this procedure till we reach the endings of the "support tree" for b, in which all leaves are now attacked by E' in R', as warranted by the reinstatement set and strong validity of FN^{EF} (see Theorem 8.18). Thus, all coherent sets for b are in fact attacked by E', and we reach a contradiction with assumption. We can finally conclude that E' is admissible in FN^{EF}.

Let $E' \subseteq A'$ be an admissible extension of FN^{EF}. We know that $E = E' \cap A$ is conflict–free in $bh - EF$. It now remains to show that every argument $a \in E$ is acceptable w.r.t. E, i.e. that for every argument b defeating$_E$ a w.r.t. E, there is an argument $c \in E$ s.t. c defeats$_E$ b and this defeat has a reinstatement set on E. Let us assume it is not the case. If b defeats$_E$ a, then either $(b, a) \in R \setminus R_D$ (and thus there is no defense attack on it), or $(b, a) \in R_D$ and there is an argument $d \in A$ s.t. $(d, (b, a)) \in D$. However, we can observe that since the (b, a) conflict becomes a defeat w.r.t. E, then none of such d's can be in E and thus not in E'. If $(b, a) \in R \setminus R_D$, then $(b, a) \in R'$. If $(b, a) \in R_D$, then $(b, b') \in R'$, and as E' is coherent and no supporter generated by a defense attacker is in E', then $b' \in E'$. Therefore, b is attacking an argument in E' and needs to be defended from in FN^{EF}. Let us assume that there is no $c \in E$ s.t. c defeats$_E$ b in $bh - EF$. This means that either there is no $(c, b) \notin R$, or for every $(c, b) \in R$ we can find an argument $d \in E$ s.t. $(d, (c, b)) \in D$. If it is the first case, then $(c, b) \notin R'$ and b has no supporting set in N' associated with c'. We can observe that every supporting set of b in N' contains a primed argument corresponding to an attacker of b in R and due to absence of these attackers in E', the primed arguments themselves are not attacked by E. Since they require no support in N', we can construct a coherent set for b containing b and the relevant primed arguments and no member of this set is attacked by E' in R'. Therefore, E' cannot defend a from b. If $(c, b) \in R$ and $\exists d \in E, (d, (c, b)) \in D$, then $(c, c') \in R'$ and there is a set $B \subseteq A'$ s.t. $\{c', d\} \subseteq B$ and $BN'b$. This means that for every supporting set for b, we can either find an standard argument in it that is contained in E, or like explained before, a primed argument that is not attacked by E. Every primed argument $f' \in A'$ has a trivial unattacked coherent set $\{f'\}$, and every argument d has to have an unattacked coherent set on E'. Since FN^{EF} is strongly valid, we can easily recombine these coherent sets for an unattacked coherent set for b. Therefore, yet again E' cannot defend a from b. Let us move on to the case in which an argument $m \in E$ can defeat$_E$ b, but we lack a reinstatement set for it. By Theorem 2.46, for any such defeat there exists a sequence $((z_1, (x_1, y_1)), ..., (z_n, (x_n, y_n)))$ of distinct defense attacks from D that contains a defense attack on (m, b) and meets the following requirements: 1. every (x_i, y_i) conflict is unique, 2. for every $(z_i, (x_i, y_i))$ where $1 < i \leq n$, either no argument h in E defeats$_E$ z_i or for every such defeat, there exists an $(h,(b,z_i)) \in \{(z_1,(x_1,y_1)),...,(z_{i-1},(x_{i-1},y_{i-1}))\}$, 3. and no argument in E defeats$_E$ z_1. Based on the previous part of the proof we can observe that z_1 possesses a coherent sequence on A' that is not attacked by E'. Moreover, by the construction of FN^{EF} and properties of the sequence, for every set of arguments supporting z_2, either z_1 or a primed argument not attacked by E' is included. Thus, we can again obtain a coherent sequence for z_2 that is not attacked by E'. We can continue with this line of reasoning until we reach a conclusion that there is an unattacked coherent sequence for z_n and thus for b. Therefore, E' could not have defended itself against b and we reach a contradiction with the admissibility of E' in FN^{EF}. We can conclude that if E' is admissible in FN^{EF}, then so is E in $bh-EF$.

Let $E \subseteq A$ be a complete extension of $bh-EF$. The set $E' = E \cup \{x' \mid x' \in X, x \in E^+\}$ is an admissible extension of FN^{EF}. We wil now show it is complete.

Let $a \in A \setminus E$ be an argument. Since $E \subseteq A$ is complete, then it means that there exists an argument $b \in A$ s.t. b defeats$_E$ a, and either there is no argument $c \in E$ s.t. c defeats$_E$ b or there is no reinstatement set for such a defeat. If b defeats$_E$ a, then either $(b,a) \in R'$, or $(b,b') \in R'$ and $\{datt(b,a) \cup \{b'\}\}N'a$ in a way that $datt(b,a) \cap E = \emptyset$ (this also means that $datt(b,a) \cap E' = \emptyset$) [32]. Let us focus on the first case. If no argument $c \in E$ defeats$_E$ b, then either there is no $(c,b) \in R$ or $\exists d \in E$ s.t. $(d,(c,b)) \in D$. If it is the first case, then as explained in the previous parts of the proof, we can create a coherent set for b consisting of primed versions of its attackers (in R), which by construction are not attacked by E' in R'. Thus, E' cannot defend a against b. If it is the case that the attacks are defense attacked, then we can create a coherent set for b by joining the primed versions of its attackers in R' that are not in E' and the coherent sets for the relevant d's in E' (existence of such sets is ensured by strong coherence of E'). Since this set is not attacked, E' cannot defend a again. If c defeats$_E$ b, but the defeat does not have a reinstatement set, then by repeating the previous part of the proof we can show that there exists a coherent set for b on A' that is not attacked by E'. Thus, it cannot be the case that E' defends a. Let us now focus on the case in which $(b,b') \in R$. For the same reasons as in the previous case, E' cannot defend b' and thus $b' \notin E'$. Consequently, $\{datt(b,a) \cup \{b'\}\} \cap E' = \emptyset$ and therefore $E' \cup \{a\}$ is not a coherent set. Therefore, E' cannot defend a.

Let $a \in X'$ be a primed argument. By construction, x' is added to E' only if E defeats$_E$ x and there is a reinstatement set for this defeat on E. Since $x' \notin E'$, we either lack the defeat or the reinstatement for x. Therefore, we can repeat the similar part of the proof for a being a standard argument and show that E' cannot defend x'.

We can thus conclude that there is no argument $a \in A \setminus E'$ that is defended by E' in FN^{EF} and therefore, E' is complete. We can also show this is the only complete extension that can be associated with E. No further X' arguments can be defended by our extension. Moreover, none of the X' can be removed from it either, since the fact that the original arguments are defeated in EF leads to their defense in FN^{EF}.

Let $E' \subseteq A'$ be a complete extension of FN^{EF}. The set $E = E' \cap A$ is admissible in $bh-EF$. Let us assume it is not complete; it means there exists an argument $a \in A \setminus E$

[32]Recall that $datt(b,a) = \{c \mid (c,(b,a)) \in D\}$

that is defended by E, i.e. for every argument b that defeats$_E$ a, there is an argument $c \in E$ s.t. c defeats$_E$ b and there is a reinstatement set for this defeat. Due to conflict–freeness of E and E', we can observe that b cannot be present in the extensions. We can repeat the part of the EAF–AFN admissibility proof in order to show that if $c \in E$ defeats$_E$ b and there is a reinstatement set for this defeat, then E' attacks all coherent sets for b in FN^{EF}. Let us now check whether $E' \cup \{a\}$ is coherent. Let $att(a) = \{d \mid (d,a) \in R\}$ be the set of attackers of a in R. If for no attackers $d \in att(a)$ $(d,a) \in R_D$, then a requires no support and $E' \cup \{a\}$ is coherent. If $(d,a) \in R_D$, then we have two options: d defeats$_E$ a or not. If not, then a defense attacker for (d,a) is present in E. Thus, it is in E', and the supporting set generated for this conflict has a shared element with E'. Let us assume d defeats$_E$ a – we basically come back to our b. Since $(d,a) \in R_D$, then b possesses a primed argument and a corresponding set that is supporting a. From the fact that E' attacks all coherent sets of d and d' has a trivial coherent set $\{d'\}$, it follows that E' defends d' and therefore $d' \in E'$ by the completeness of E'. Hence, the support set related to d' has a shared element with E'. We have now covered all cases and every support there is for a has an element in common with E'. Therefore, $E' \cup \{a\}$ has to be coherent and E' defends a. This contradicts the completeness of E' in FN^{EF}. We can now conclude that if E' is complete in FN^{EF}, then E is complete in $bh - EF$.

By Theorem 2.95 and 2.54, the preferred extensions of $bh - EF$ and FN^{EF} can be defined as maximal preferred ones. We can observe that the source and target complete extensions are in a one–to–one relation and that for two complete extensions of $bh - EF$ E_1 and E_2 s.t. $E_1 \subset E_2$, it holds that their corresponding sets E_1' and E_2' in FN^{EF} are related in the same manner, i.e. $E_1' \subset E_2'$. Thus, we can conclude that if E is preferred in $bh - EF$, then $E' = E \cup \{x' \mid x' \in X, x \in E^+\}$ is preferred in FN^{EF}, and if $E' \subseteq A$ is preferred in FN^{EF}, then $E = E' \cap A$ is preferred in $bh - EF$. Since we are dealing with a bounded hierarchical EAF, then by Definition 2.57 the grounded extension of is the least complete extension of $bh - EF$. Similarly, the grounded extension is the least complete extension in FN^{EF} by Theorem 2.95. Therefore, the same relation holds for the grounded semantics as for the preferred.

What remains to be shown is that the stable semantics are also preserved. Let $E \subseteq A$ be a stable extension of $bh - EF$. By Theorem 2.54, it is complete. Thus, $E' = E \cup \{x' \mid x' \in X, E$ defeats$_E$ x and there is a reinstatement set for this defeat on $E\}$ is complete in FN^{EF}. Since E is stable, then for every argument $a \in A \setminus E$, there is an argument $b \in E$ s.t. b defeats$_E$ a. Furthermore, we can observe that for this defeat, there is a reinstatement set on E – this comes simply from the fact that every argument outside the set is defeated$_E$ and thus the collection of all such defeats coming from E forms a reinstatement set. From the previous analysis it holds that either $(b,a) \in R'$, or there is a support set of a containing b' that is disjoint from E'. Consequently, a is in the deactivated set of E'. Let us assume there is a primed argument $x' \notin E'$ that is not in the deactivated set. Since it receives no support through N', then it has to be the case that its original argument x is not in E'. However, this means that E defeats$_E$ x, and as it is with reinstatement, then we reach a contradiction with the construction of E'. Therefore, E' is a stable extension of FN^{EF}.

Let $E' \subseteq A'$ be a stable extension of FN^{EF}. By Definition 2.86, it is complete in FN^{EF}. Consequently, $E = E' \cap A$ is complete (and therefore, conflict–free) in $bh - EF$. By Lemma 2.94, E' attacks every coherent set of an argument $a \in A' \setminus E'$. Using the previous explanations we can thus show that, under the assumption that $a \in A$, E defeats$_E$ a with reinstatement. Therefore, E is complete and defeats$_E$ every argument in $A \setminus E$. We can conclude that E is stable in $bh - EF$. □

Theorem 8.21. *Let $bh - EF = (A, R, D)$ be a bounded hierarchical EAF and $FN^{EF} = (A', R', N')$ its corresponding AFN obtained through Translation 44. If a set $E \subseteq A$ is a σ–extension of $bh - EF$, where $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$, then there is a σ–extension $E' \subseteq A'$ of FN^{EF} s.t. $E' \cap A = E$. If a set $E' \subseteq A'$ is a σ'–extension of FN^{EF}, where $\sigma' \in \{$admissible, complete, preferred, grounded, stable$\}$, then $E = E' \cap A$ is a σ'–extension of $bh - EF$.*

Proof. Let $FR = (A, R \cup D)$ be the AFRA corresponding to $bh - EF$ and obtained through Translation 42. We can observe that the AFN associated with FR (Translation 35) is the same as FN^{EF}.

By Theorem 8.15, for every δ–extension $E \subseteq A$ of $bh - EF$, where $\delta \in \{$conflict–free, admissible, complete, grounded, preferred, stable$\}$ is a semantics, there is a δ–extension $E' \subseteq A$ s.t. $E' \cap A = E$. By Theorem 7.6, if $E' \subseteq A \cup R \cup D$ is a δ'–extension of FR, where $\delta \in \{$conflict–free, complete, grounded, preferred, stable$\}$, then it is a δ'–extension of FN^{EF}. However, by the same theorem, if E' is source–complete in FR, i.e. $E' = E' \cup \{src(V) \mid V \in E \cap (R \cup D)\}$ (see Definition 7.5), then it is admissible in FN^{EF}. It can be shown that for every admissible extension of $bh - EF$ there is a corresponding admissible one in FR that is source complete. Consequently, there will exist a suitable extension on FN^{EF}. With this we can conclude that if a set $E \subseteq A$ is a σ–extension of $bh - EF$, where $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$, then there is a σ–extension $E' \subseteq A'$ of FN^{EF} s.t. $E' \cap A = E$.

For the same reasons as in the AFRA–AFN case, not every conflict–free extension of FN^{EF} will produce a conflict–free extension of $bh - EF$. By Theorem 7.6, if $E' \subseteq A'$ is a δ–extension of FN^{EF}, where $\delta \in \{$admissible, complete, grounded, preferred, stable$\}$ is a semantics, then E' is a δ–extension of FR. By Theorem 8.15, if E' is a δ'–extension of FR, where $\delta' \in \{$complete, grounded, preferred, stable$\}$, then $E = E' \cap A$ is a δ'–extension of $bh - EF$. We can observe that due to the fact that all δ semantics are also coherent in FN^{EF}, then attacks have to be accompanied by its sources and thus E' will always be source–complete in FR. Again, it can be shown that if E' is admissible and source complete in FR, then $E = E' \cap A$ is admissible in $bh - EF$. Therefore, we can conclude that if a set $E' \subseteq A'$ is a σ'–extension of FN^{EF}, where $\sigma' \in \{$admissible, complete, preferred, grounded, stable$\}$, then $E = E' \cap A$ is a σ'–extension of $bh - EF$. □

Theorem 8.28. *Let $EF = (A, R, D)$ be an EAF s.t. it is bounded hierarchical, or (strongly) consistent and without symmetric attacks. Let $D^{EF} = (A, L, C)$ be its corresponding ADF obtained through Translation 47. D^{EF} is a BADF. It is also in redundancy–free, cleansed, and weakly valid form. It is not necessarily an AADF$^+$ and does not have* *to be in relation or strongly valid form. If EF is bounded hierarchical, then D^{EF} is an AADF$^+$ and is in relation and strongly valid forms.*

Proof. Let $a, b \in A$ be arguments s.t. $(a, b) \in R$. By consistency it means there is no other attack on b that would be defense attacked by a. If a given set E defeats$_E$ b, then so does $E \cup \{a\}$. Therefore, there is no subset F of parents of b in D^{EF} s.t. $C_b(F) = out$ and $C_b(F \cup \{a\}) = in$. The (a, b) link in D^{EF} is thus an attacking one based on Definition 2.114. Furthermore, it cannot be supporting – due to consistency, $C_b(\emptyset) = in$ and $C_b(\{a\}) = out$.

Let now $a, b \in A$ be arguments s.t. there is $c \in A$, $(a, (c, b)) \in D$. Due to consistency, it cannot be the case that $(a, b) \in R$. This means that if E does not defeat$_E$ b, then neither does $E \cup \{a\}$. Therefore, there is no subset F of parents of b in D^{EF} s.t. $C_b(F) = in$ and $C_b(F \cup \{a\}) = out$. The (a, b) link in D^{EF} is thus a supporting one based on Definition 2.114. Furthermore, it cannot be attacking – due to consistency, $C_b(\{c\}) = out$ and $C_b(\{a, c\}) = in$.

Since every link in D^{EF} is either supporting or attacking, our framework is a BADF. As none of the links is supporting and attacking at the same time, then the framework is also redundancy–free.

For every argument $a \in A$, $C_a(\emptyset)$ is trivially in due to the fact that no attacker of a in R is present in the set. Since every argument has a set for which the condition is in, then every argument has a trivial associated decisively in interpretation without $\mathbf{t}$ mappings and where every argument in A is mapped to $\mathbf{f}$. Thus, a minimal interpretation extracted from it can only have less $\mathbf{f}$ mappings (if it is different at all). The collection of such interpretations can be used as a basis for an acyclic pd–evaluation consisting of all the elements of A. Thus, every argument has an acyclic pd–evaluation (even if a self–blocking one) and D^{EF} is both in cleansed and weakly valid form.

We can observe that the framework presented in Example 103 is not an AADF$^+$. It is neither in relation nor in strongly valid form.

Let us now assume that EF is bounded hierarchical and let $(((A_1, R_1), D_1), ..., ((A_n, R_n), D_n))$ be its partition satisfying the requirements in Definition 2.55. Let us start with $((A_n, R_n), D_n)$. We can observe that as $D_n = \emptyset$, then all of the parents of $a \in A_n$ are in A_n. Furthermore, they are only connected by the R_n relation, which means that all arguments in A_n in D^{EF} have Dung–style acceptance conditions. Therefore, as observed in the proof of Theorem 5.17, every argument in A_n has precisely one minimal decisively in interpretation that does not contain any $\mathbf{t}$ mappings and every standard evaluation on A_n can be made acyclic. Let us now focus on $((A_{n-1}, R_{n-1}), D_{n-1})$. Notice that $D_{n-1} \subseteq A_n$. Every argument $a \in A_{n-1}$ depends only on arguments in $A_{n-1} \cup A_n$. Furthermore, if a minimal decisively in interpretation for a contains $\mathbf{t}$ mappings, then those mappings can be in A_n only. Therefore, any ordering on A_n extended with any ordering on A_{n-1} will give us a pd–sequence of an acyclic pd–evaluation, independently of the chosen minimal decisively in interpretations for the arguments. Therefore, every standard evaluation on $A_{n-1} \cup A_n$ can be made acyclic. We can continue this line of reasoning until we reach $((A_1, R_1), D_1)$ and the conclusion that every standard evaluation on $A = \bigcup_{i=1}^{n} A_n$ can be made acyclic. Thus, D^{EF} is an AADF$^+$. From the previous parts of the proof it follows that it is also redundancy–free and cleansed. Consequently, by Theorem 4.43, it is also strongly valid, and by Theorem 4.41, relation valid. □

Theorem 8.29. *Let EF be an EAF s.t. it is bounded hierarchical, or (strongly) consistent and without symmetric attacks. Let $D^{EF} = (A, L, C)$ be its corresponding ADF obtained through Translation 47. A set of arguments $E \subseteq A$ is a conflict–free extension of EF iff it is conflict–free in D^{EF}.*

Proof. The proof follows easily from Theorem 8.2 and Theorem 8.39 that will be discussed in the next section. □

Lemma 8.30. *Let EF be an EAF s.t. it is bounded hierarchical, or (strongly) consistent and without symmetric attacks. Let $D^{EF} = (A, C)$ be its corresponding ADF obtained through Translation 47. Let E be a conflict–free extension of EF (and thus of D^{EF}). The discarded set of E in EF coincides with the partially acyclic discarded set of E in D^{EF}.*

Proof. The proof follows easily from Theorem 8.2 and Lemma 8.40 that will be discussed in the next section. □

Lemma 8.31. *Let EF be an EAF s.t. it is bounded hierarchical, or (strongly) consistent and without symmetric attacks. Let $D^{EF} = (A, C)$ be its corresponding ADF obtained through Translation 47. A conflict–free set of arguments E defends an argument $a \in A$ in EF iff a is decisively in w.r.t. the partially acyclic range v_E^p of E in D^{EF}.*

Proof. The proof follows easily from Theorem 8.2 and Lemma 8.41 that will be discussed in the next section. □

Theorem 8.32. *Let EF be an EAF s.t. it is bounded hierarchical, or (strongly) consistent and without symmetric attacks. Let $D^{EF} = (A, C)$ be its corresponding ADF obtained through Translation 47. A set of arguments $E \subseteq A$ is a conflict–free extension of EF iff it is conflict–free in D^{EF}. E is a stable extension of EF iff it is a model of D^{EF}. E is a grounded extension of EF iff it is the acyclic grounded extension of D^{EF}. E is a σ–extension of EF, where $\sigma \in \{$admissible, complete, preferred$\}$, iff it is a ca_2–σ–extension of D^{EF}.*

Proof. The proof follows easily from Theorem 8.2 and Theorem 8.42 that will be discussed in the next section. □

Theorem 8.36. *Let $EF = (A, R, D)$ be a bounded hierarchical EAF or an EAF without symmetric attacks and $D^{EF} = (A', L, C)$ its corresponding ADF obtained through Translation 48. D^{EF} is a BADF. It is also in redundancy–free, cleansed, and weakly valid form. However, it is not necessarily an AADF$^+$ and does not have to be in relation or strongly valid form. If EF is a bounded hierarchical EAF, then D^{EF} is an AADF$^+$ and is in relation and strongly valid forms.*

Proof. Let $a, b \in A$ be arguments s.t. $(a, b) \in R$. Based on the transformation we can observe that a occurs only once in the condition of b as a negative literal – if there is another attack on b that is defense attacked by a, then a bypass argument for a is introduced. This means that even if a given set E defeats$_E$ b but $E \cup \{a\}$ does not, there is no subset F of parents of b in D^{EF} s.t. $C_b(F) = out$ and $C_b(F \cup \{a\}) = in$. The (a, b) link in D^{EF} is thus an attacking one based on Definition 2.114. Furthermore, it cannot be supporting – $C_b(\emptyset) = in$ and based on the translation, $C_b(\{a\}) = out$.

Let now $a, b \in A$ be arguments s.t. there is $c \in A$, $(a, (c, b)) \in D$. If it is the case that $(a, b) \in R$, then we come back to the previous analysis and it is the a^b–b link that we need to focus on. We can observe that $C_b(\{c\}) = out$ and $C_b(\{a^b, c\}) = in$ by the construction of the conditions. Therefore, the (a^b, b) link is not attacking. Furthermore, the condition of C_b is out only if an attacker from R is present and every possible (modified) defense attacker is absent. a^b never attacks b in R – it is simply not present in the framework. Therefore, there is no subset F of parents of b in D^{EF} s.t. $C_b(F) = in$ and $C_b(F \cup \{a^b\}) = out$ and the link is supporting. If $(a, b) \notin R$, then this means that if E does not defeat$_E$ b, then neither does $E \cup \{a\}$. Therefore, there is no subset F of parents of b in D^{EF} s.t. $C_b(F) = in$ and $C_b(F \cup \{a\}) = out$. The (a, b) link in D^{EF} is thus a supporting one based on Definition 2.114. Furthermore, it cannot be attacking – c attacks b and thus $C_b(\{c\}) = out$ and as a does not attack b and overrides the conflict from c, $C_b(\{a, c\}) = in$.

Finally, every bypass argument $a^b \in A^b$ has a straightforward condition $C_{a^b} = a$ and the (a, a^b) link is easily supporting and not attacking. We can conclude that every link in D^{EF} is either supporting or attacking. Therefore, our framework is a BADF. As none of the links is supporting and attacking at the same time, then the framework is also redundancy–free.

The proof that D^{EF} is in cleansed and weakly valid form is the same as in Theorem 8.28.

We can observe that the framework presented in Example 103 is not an AADF$^+$. It is neither in relation nor in strongly valid form. By Lemma 4.67, if EF is bounded hierarchical, then it is (strongly) consistent. Thus, by Theorem 8.28, if EF is bounded hierarchical, then D^{EF} is an AADF$^+$ and is in relation and strongly valid forms. □

Theorem 8.37. *Let $EF = (A, R, D)$ be a bounded hierarchical EAF or an EAF without symmetric attacks and $D^{EF} = (A', L, C)$ its corresponding ADF obtained through Translation 48. Let E^b denote the (possibly empty) set of bypass arguments generated by E in A'.*

If a set of arguments $E \subseteq A$ is a conflict–free extension of EF then $E' = E \cup E^b$ is conflict–free in D^{EF}. If $E' \subseteq A'$ is conflict–free in D^{EF}, then $E = E' \cap A$ is conflict–free in EF.

If a set of arguments $E \subseteq A$ is a stable extension of EF then $E' = E \cup E^b$ is a model of D^{EF}. If $E' \subseteq A'$ is a model of D^{EF}, then $E = E' \cap A$ is stable in EF.

If a set of arguments $E \subseteq A$ is the grounded extension of EF then $E' = E \cup E^b$ is the acyclic grounded extension of D^{EF}. If $E' \subseteq A'$ is the acyclic grounded extension of D^{EF}, *then $E = E' \cap A$ is grounded in EF.*

If $E \subseteq A$ is a σ–extension of EF, where $\sigma \in \{$admissible, complete, preferred$\}$, then $E' = E \cup E^b$ is a ca$_2$–σ–extension of D^{EF}. If $E' \subseteq A'$ ca$_2$–σ–extension of D^{EF}, then $E = E' \cap A$ is a σ–extension of EF.

Proof. The proof follows easily from Theorems 8.2 and 8.47. □

Theorem 8.38. *Let $EFC = (A, R, D)$ be a strongly consistent EAFC and $D^{EFC} = (A, C)$ its corresponding ADF obtained through Translation 49. D^{EF} is a BADF. It is also in cleansed and weakly valid form. It is not necessarily an AADF$^+$ and does not have to be in redundancy–free, relation or strongly valid form. if EFC is minimal, then D^{EF} is redundancy–free. If EFC is bounded hierarchical, then D^{EFC} is an AADF$^+$, and if it is additionally minimal, then D^{EFC} is in relation and strongly valid forms.*

Proof. Let $a, b \in A$ be arguments s.t. $(a, b) \in R$. By strong consistency it means there is no other attack on b that would be defense attacked by a set containing a. If a given set E has a subset defeating$_E$ b, then so does $E \cup \{a\}$. Therefore, there is no subset F of parents of b in D^{EFC} s.t. $C_b(F) = out$ and $C_b(F \cup \{a\}) = in$. The (a, b) link in D^{EFC} is thus an attacking one based on Definition 2.114. Furthermore, it cannot be supporting – due to consistency, $C_b(\emptyset) = in$ and $C_b(\{a\}) = out$.

Let now $a, b \in A$ be arguments s.t. there is $c \in A$, $G \subseteq A$, $a \in G$ and $(G, (c, b)) \in D$. Due to consistency, it cannot be the case that $(a, b) \in R$. This means that if E does not defeat$_E$ b, then neither does $E \cup \{a\}$. Therefore, there is no subset F of parents of b in D^{EFC} s.t. $C_b(F) = in$ and $C_b(F \cup \{a\}) = out$. The (a, b) link in D^{EFC} is thus a supporting one based on Definition 2.114. However, there might not exist a subset F' of parents of b s.t. $C_b(F') = out$ and $C_b(F' \cup \{a\}) = in$. Let us consider a situation in which the (c, b) attack is defense attacked by $\{d\}$ and $\{a, d\}$. The condition of b is in for $\emptyset$, $\{d\}$ and $\{c, d\}$ as well as for $\{a\}$, $\{a, d\}$ and $\{c, d, a\}$. Therefore, it is not necessarily the case that the (a, b) link is not attacking. Let us now assume that EFC is minimal; this means there is no $G' \subset G$ s.t. $(G', (c, b)) \in D$. Due to consistency restrictions, c cannot be defense attacking the conflict it carries out and thus $C_b(\{c\}) = out$. Moreover, $C_b((G \setminus \{a\}) \cup \{c\} = out$ as well – no defense attacking set for the (c, b) conflict is present. However, $C_b(G \cup \{c\} = in$ – now the defense attacking set is present, and due to consistency constraints none of the arguments in G can be attacking b. Thus, the (a, b) link cannot be attacking anymore.

Since every link in D^{EFC} is either supporting or attacking, our framework is a BADF. As some of the links can be supporting and attacking at the same time, it might not be redundancy–free. However, assuming that EFC is minimal addresses this issue.

The proof for the cleansed form, weakly valid form, and the fact that D^{EFC} is an AADF$^+$ if EFC is bounded hierarchical, can be easily adapted from Theorem 8.28. If EFC is minimal and bounded hierarchical, then the D^{EFC} is a redundancy–free AADF$^+$. As it is also in cleansed form, then by Theorem 4.43 and 4.41, D^{EFC} is also strongly and relation valid. □

Theorem 8.39. *Let EFC be a strongly consistent EAFC and $D^{EFC} = (A, L, C)$ its corresponding ADF obtained through Translation 49. A set of arguments $E \subseteq A$ is a conflict–free extension of EFC iff it is a conflict–free extension of D^{EFC}.*

Proof. Let $E \subseteq A$ be a conflict–free extension of EFC. This means that given an argument $a \in E$, it is either not attacked at all in E or every attack carried out by a member of E is defense attacked by a subset of E. Thus, from the functional version of the acceptance conditions in Translation 49 we can observe that $C_a(E \cap par(a)) = in$. Consequently, if E is conflict–free in EFC, then every argument in E has a satisfied acceptance condition w.r.t. E in D^{EFC}. This means that E is conflict–free in D^{EFC}.

Let now $E \subseteq A$ be a conflict–free extension of D^{EFC}. This means that for any argument $a \in E$, $C_a(E \cap par(a)) = in$. By the construction of the condition it means that either there is no argument $b \in E$ s.t. $(b, a) \in R$, or for any such attack there is a subset of E defense attacking it. Consequently, there are no defeats$_E$ in E in EFC and thus E is conflict–free in EFC as well. □

Lemma 8.40. *Let EFC be a strongly consistent EAFC and $D^{EFC} = (A, L, C)$ its corresponding ADF obtained through Translation 49. Let $E \subseteq A$ be a conflict–free extension of EFC (and thus of D^{EFC}). The discarded set of E in EFC coincides with the partially acyclic discarded set of E in D^{EFC}.*

Proof. Let us first note on how (minimal) decisively in interpretations for arguments in A look like. Due to the fact that we are dealing with a strongly consistent framework, then from the propositional acceptance conditions we can observe that for any attack subformula of the condition, the interpretation has to either map the attacker to $\mathbf{f}$ or at least one defense attacking sets to $\mathbf{t}$. Thus, even though technically speaking EAFCs are attack–based frameworks, the minimal interpretations can contain $\mathbf{t}$ assignments, which was not the case in e.g. AFs or SETAFs. If the framework was not consistent, then we could obtain new minimal decisively in interpretations that would be contained in the described ones. For example, the condition of b in the framework $(\{a, b\}, \{(a, b)\}, \{(a, (a, b))\})$ would be equivalent to $\top$ and thus an empty translation would have been also possible, despite the fact that the argument is attacked by a and $\{b\}$ is not an admissible extension of EFC.

Let $E \subseteq A$ be a conflict–free extension of EFC. By Lemma 8.39, E is conflict–free in D^{EFC}. We define the set E^+ as the collection of those arguments $b \in A$ s.t. an argument $a \in E$ defeats$_E$ b and there is a reinstatement set for this defeat on E. Clearly, by conflict–freeness of E, $E \cap E^+ = \emptyset$. We will show that this set is equal to the partially discarded set E^{p+} in D^{EFC}.

Let $b \in E^+$ in EFC. Assume it does not qualify for E^{p+} in D^{EFC}; this means that b has a partially acyclic evaluation (F, G, B) on A s.t. $B \cap E = \emptyset$ and $F \subseteq E$. Let $G = (a_0, ..., a_n)$ be the pd–sequence of the evaluation. Due to the construction of the sequence, the $\mathbf{t}$ part of the decisively in interpretation v_{a_0} used for a_0 in the construction of (F, G, B) is contained in F and thus in E. Since $B \cap E = \emptyset$, $v^{\mathbf{f}}_{a_0} \cap E = \emptyset$. Therefore, by the construction of the decisively in interpretations in consistent frameworks and the nature of the acceptance conditions in D^{EFC}, this means that if there is an attacker of a_0 present in the set, then the conflict is defense attacked by a subset of E and thus there is no defeat. Consequently, a_0 does not qualify for E^+.

Let us continue onto a_1. Assume that v_{a_1} is its associated minimal decisively in interpretation. We know that $v_{a_1}^{\mathbf{t}} \subseteq E \cup \{a_0\}$ and that $v_{a_1}^{\mathbf{f}} \cap E = \emptyset$. From the construction of interpretations and conditions, this means that if there is an attack carried out at a_1 by some element of E, then it is defense attacked by a subset of $E \cup \{a_0\}$. Since a_0 is not defeated by any argument in E, then either no argument in E defeats a_1 (i.e. no attacker of a_1 is present or $a_0 \in E$) or for no defeat by E on a_1 there is a reinstatement set on E. Consequently, a_1 does not qualify for E^+ in EFC. We can continue reasoning in this manner till we reach $a_n = b$ and the conclusion that if b has a partially acyclic evaluation (F, G, B) s.t. $F' \subseteq E$ and $B \cap E = \emptyset$ in D^{EFC}, then it cannot be in E^+ in EFC.

We have just shown that $E^+ \subseteq E^{p+}$. We now need to prove that there is no argument $b \in E^{p+}$ in D^{EFC} that is not in E^+ in EFC. Assume it is not the case; therefore, either no argument in E defeats$_E$ b or no such defeat has a reinstatement set on E in EFC, even though $b \in E^{p+}$ in D^{EFC}. Let us focus on the first case. If there is no defeat, then there is either no attack on b from E in the first place, or for every attack there is a subset of E carrying out an appropriate defense attack. Consequently, we can observe that the acceptance condition of b w.r.t. $E \cap par(b)$ should be mapped to in. Thus, by Proposition 2.150, b could not have been in E^{p+} and we reach a contradiction with the assumptions. Let us now focus on the case where there is a defeat on b by an argument $d \in E$, but it lacks a reinstatement set on E. By Theorem 2.64, there exists a sequence of distinct defense attacks $((Z_1, (x_1, y_1)), ..., (Z_n, (x_n, y_n)))$ s.t. $(x_n, y_n) = (d, b)$, each (x_i, y_i) attack is unique, no argument in E defeats$_E$ any element $z \in Z_1$, and for every other $(Z_i, (x_i, y_i))$ in the sequence, either no argument $h \in E$ defeats$_E$ any element $z' \in Z_i$ or for every such defeat there is a set of arguments $L \subseteq A$ s.t. $(L, (h, z') \in \{(Z_1, (x_1, y_1)), ..., (Z_{i-1}, (x_{i-1}, y_{i-1}))\}$. Let us start with the set Z_1. We can observe that if E does not defeat$_E$ any argument in Z_1, then the conditions of the arguments in Z_1 are in fact satisfied by E. Thus, no element of Z_1 is in the partially acyclic discarded set by Proposition 2.150. Let us now consider Z_2 and let $z \in Z_2$ be an argument. If it is not defeated$_E$ by E, then we come back to the previous case and can show that z cannot be in the partially acyclic discarded set. If it is defeated$_E$, then the condition of z is out w.r.t. E. However, we can observe that by the construction, the condition of z w.r.t. $E \cup Z_1$ is in, and as no element in Z_1 is in the partially acyclic discarded set, then the argument cannot be decisively out w.r.t. the partially acyclic range. Thus, it is not in the partially acyclic discarded set by Proposition 2.150. We can therefore show that $Z_2 \cap E^{p+} = \emptyset$. We can continue this line of reasoning until we reach Z_n and the result that $Z_n \cap E^{p+} = \emptyset$. Consequently, y_n cannot be decisively out w.r.t. the partially acyclic range either and $y_n = b \notin E^{p+}$. We reach a contradiction with the assumptions. Therefore, $E^{p+} \subseteq E^+$. We can thus finally conclude that $E^+ = E^{p+}$. □

Lemma 8.41. *Let EFC be a strongly consistent EAFC and $D^{EFC} = (A, L, C)$ its corresponding ADF obtained through Translation 49. A conflict–free set of arguments $E \subseteq A$*

defends an argument $a \in A$ in EFC iff a is decisively in w.r.t. the partially acyclic range v^p_E of E in D^{EFC}.

Proof. In Theorem 8.39 we have shown that the conflict–free extensions of EFC and D^{EFC} coincide. In Lemma 8.40, we have proved that the set of arguments defeated by E with a reinstatement set on E in EFC equals the partially acyclic discarded set of E in D^{EFC}. Now, we need to prove that an argument $a \in A$ is defended by E in EFC iff it is decisively in w.r.t. the partially acyclic range interpretation v^p_E of E in D^{EFC}.

Let us start with left to right direction. If an argument a is defended by E, then every argument $b \in E$ s.t. b defeats$_E$ a, is in turn defeated with reinstatement by E. Therefore, a is defended iff every argument $b \in A$ defeating it is in E^+. Let us now consider an argument c s.t. $(c, a) \in R$, but c does not defeat$_E$ a. This means that there is a suitable defense attack carried out by a set $F \subseteq E$. We can now shift to D^{EFC}. Every attacker of a, be it b style (i.e. it becomes a defeater) or c style (i.e. does not become a defeater), has a corresponding att formula in the condition of a and this formula is not equivalent to $\top$ due to the strong consistency of EFC. If it is a formula att^b_a, then we can observe that as b is mapped to **f** by the partially acyclic discarded range, the formula evaluates to true under this range and will remain such independently of what is assigned to the remaining arguments in the formula. If it is a formula att^c_a, then the disjunction of conjunctions corresponding to the defense attackers evaluates to true and thus the whole att^c_a is true. Moreover, it will stay such, no matter what new arguments come into play. Consequently, the condition of a is in under the partially acyclic range and will remain in for any of its completions to A. Thus, a is decisively in w.r.t. the partially acyclic range of E.

Let us continue with the right to left direction. If an argument a is decisively in w.r.t. the partially acyclic range, then its condition is in w.r.t. every completion of the range to A. This means that every $att^b_a = \neg b \vee (\bigwedge B_1 \vee ... \bigwedge B_m)$ subformula of the acceptance condition evaluates to true under the acyclic range and remains such under every completion. Therefore, it is either b that has to be assigned **f** by the range or at least one set B_i has all arguments assigned **t** by the range. If it is the first case, then by Lemma 8.40, $b \in E^+$ and if the attack from b is a defeat, then a is defended from b by E in EFC. If it is the latter, then we can observe that the attack from b on a does not become a defeat. Since the att subformulas account for all attackers of a, we can conclude that E defends a. □

Theorem 8.42. *Let EFC be a strongly consistent EAFC and $D^{EFC} = (A, L, C)$ its corresponding ADF obtained through Translation 49. A set of arguments $E \subseteq A$ is a conflict–free extension of EFC iff it is conflict–free in D^{EFC}. E is a stable extension of EFC iff it is a model of D^{EFC}. E is a grounded extension of EFC iff it is the acyclic grounded extension of D^{EFC}. Finally, E is a σ–extension of EFC, where $\sigma \in \{$admissible, complete, preferred$\}$, iff it is a ca_2–σ–extension of D^{EFC}.*

Proof. With the help of Theorem 8.39, Lemmas 8.40 and 8.41, it can be shown that $E \subseteq A$ is a σ–extension of EFC, where $\sigma \in \{$admissible, complete preferred$\}$ iff it is a $ca_2 - \sigma$–extension of D^{EFC}. What remains to be proved is the relation between stable extensions and models, and the grounded and acyclic grounded extensions.

Let $E \subseteq A$ be a stable extension of EFC. This means it is conflict–free and defeats$_E$ every argument $a \in A \setminus E$. We can observe that every defeat$_E$ originating from E will be a trivial reinstatement set for any of these defeats$_E$. Therefore, from Theorem 8.39 and Lemma 8.40, it follows that E is conflict–free in D^{EFC} and that every argument $a \in A \setminus E$ is in the partially acyclic discarded set. By Proposition 2.150 it holds that for every such a, $C_a(E \cap par(a)) = out$. Therefore, E is a model of D^{EFC}. As observed in Example 103, E does not need to be stable in D^{EFC}.

Let $E \subseteq A$ be a model of D^{EFC}. By Theorem 8.39, it is conflict–free in EFC. By Lemma 2.159, from the fact that E is a model it follows that every argument in $A \setminus E$ is in the partially acyclic discarded set. Consequently, it is also in E^+ in EFC, and by the definition of this set is defeated$_E$ by E. Therefore, E is stable in EFC.

In order to show that the grounded extension in EFC and the acyclic grounded in D^{EFC} correspond, we can use the iterating from the empty set approach (see Definition 2.142 and [66]). Let us start with $E = E' = \emptyset$. The set E is conflict–free in EFC and E' is pd–acyclic conflict–free in D^{EFC}. They are also (aa–) admissible in their respective frameworks. Since E' is pd–acyclic conflict–free, then the partially acyclic range of E' is in fact acyclic by Lemma 2.132. Therefore, if we perform an iteration and add to E the arguments it defends in EFC and to E' those that are decisively in w.r.t. the acyclic range of E' in D^{EFC}, then it is still the case that $E = E'$. Moreover, by Lemma 2.154, E' is still aa–admissible and thus pd–acyclic conflict–free. From the admissibility of E' follows the admissibility of E. We can now repeat the iteration and again observe that $E = E'$. We can continue in this manner until there are no arguments left (and as we are working with finite frameworks, this is warrantied) and observe that $E = E'$ and E is grounded in EFC while E' acyclic grounded in D^{EFC}. □

Theorem 8.47. *Let $EFC = (A, R, D)$ be an EAFC and $D^{EFC} = (A', L, C)$ its corresponding ADF obtained through Translation 50. Let E^b denote the (possibly empty) set of bypass arguments generated by E in A'.*

If a set of arguments $E \subseteq A$ is a conflict–free extension of EFC then $E' = E \cup E^b$ is conflict–free in D^{EFC}. If $E' \subseteq A'$ is conflict–free in D^{EF}, then $E = E' \cap A$ is conflict–free in EFC.

If a set of arguments $E \subseteq A$ is a stable extension of EFC then $E' = E \cup E^b$ is a model of D^{EFC}. If $E' \subseteq A'$ is a model of D^{EF}, then $E = E' \cap A$ is stable in EFC.

If a set of arguments $E \subseteq A$ is the grounded extension of EFC then $E' = E \cup E^b$ is the acyclic grounded extension of D^{EFC}. If $E' \subseteq A'$ is the acyclic grounded extension of D^{EF}, then $E = E' \cap A$ is grounded in EFC.

If $E \subseteq A$ is a σ–extension of EFC, where $\sigma \in \{$admissible, complete, preferred$\}$, then $E' = E \cup E^b$ is a ca_2–σ–extension of D^{EFC}. If $E' \subseteq A'$ ca_2–σ–extension of D^{EFC}, then $E = E' \cap A$ is a σ–extension of EFC.

Proof. Let $E \subseteq A$ be a conflict–free extension of EFC and $E' = E \cup E^b$ a set in D^{EFC}. Conflict–freeness of E means that given an argument $a \in E$, it is either not attacked at all by any argument in E or every attack carried out by a member of E is defense attacked by a subset of E. If it is the first case, then from the functional version of the acceptance conditions we can observe that $C_a(E' \cap par(a)) = in$. If it is the latter, then there is a defense attacking subset $B \subseteq E$ for a given attack. Therefore, there is a suitable set $B' \subseteq (E \cup E^b)$, and by construction the condition of a is again in. Finally, as every bypass argument is accompanied by its source, its condition is also satisfied. Therefore, E' is conflict–free in D^{EFC}.

Let now $E' \subseteq A$ be a conflict–free extension of D^{EFC}. This means that for any argument $a \in E'$, $C_a(E' \cap par(a)) = in$. If $a \in A^b$, then it means that the original argument for which a was created is in E. If $a \in A$, then by analyzing the acceptance condition we can observe that either there is no argument $b \in E$ s.t. $(b, a) \in R$, or for any such attack there is a transformed defense attack in E'. However, since bypasses cannot appear without sources, we can revert the defense attacking set to its original form and conclude that there is a subset of E carrying out a defense attack on (b, a). Thus, no defeats$_E$ are present in E and the set is conflict–free in EFC.

Let $E \subseteq A$ be a conflict–free extension of EFC. We define the set E^+ as the collection of those arguments $b \in A$ s.t. an argument $a \in E$ defeats$_E$ b and there is a reinstatement set for this defeat on E. Clearly, by conflict–freeness of E, $E \cap E^+ = \emptyset$. We will show that this set is equal to the partially acyclic discarded set of $E' = E \cup E^b$ without the bypasses in D^{EFC}, i.e. $E^+ = E'^{p+} \cap A$.

Let $b \in E^+$ in EFC. Assume it does not qualify for E'^{p+} in D^{EFC}; this means that b has a partially acyclic evaluation (F', G', B') on A' s.t. $B' \cap E' = \emptyset$ and $F' \subseteq E'$. Let $G' = \{a_0, ..., a_n\}$ be the sequence of this evaluation. Let us now analyze the case in which b has a partially acyclic evaluation (F', G', B') on A s.t. $B' \cap E = \emptyset$ and $F' \subseteq E'$. Again, we can focus on those F' that are completed with bypasses (possibly with the exception of the last argument). Let $G' = \{a_0, a_0^b, ..., a_n\}$ be the pd–sequence of the evaluation. By construction, the minimal decisively in interpretation v_{a_0} for a_0 used in the creation of the evaluation has a $\mathbf{t}$ part contained in F' and thus in E'. If $v_{a_0}^{\mathbf{t}} = \emptyset$, then it is not a bypass argument. Moreover, due to the fact that $v_{a_0}^{\mathbf{f}} \cap E' = \emptyset$, it cannot be the case that E attacks a_0 in EFC. Therefore, there is no defeat either, and a_0 cannot be in E^+. If $v_{a_0}^{\mathbf{t}} \neq \emptyset$, then we can distinguish two cases; one where a_0 is a bypass argument and one in which it is a standard argument. If it is the first case, then due to the fact that $F' \subseteq E'$, the original argument behind a_0 is in E and therefore cannot be in E^+ in EFC. Let us therefore focus on the other. The minimal decisively in interpretation v_{a_0} that has been used in the construction of (F', G', B') has a $\mathbf{t}$ part that is a subset of $F' \subseteq E'$. From the construction of the interpretations and conditions, this means that a given attacker of a_0 is either not present in E, or that the transformed defense attacking set K' for it is a subset of E'. Based on the construction of E' and the general fact that a bypass argument cannot appear in a conflict–free extension without its origin, the real defense attacker set K has to be a subset of E. Therefore, a_0 is not defeated by any argument in E and does not qualify for E^+ in EFC.

Let us continue onto a_1 and assume that v_{a_1} is its associated minimal decisively in interpretation. We know that $v_{a_1}^{\mathbf{t}} \subseteq E' \cup \{a_0\}$ and that $v_{a_1}^{\mathbf{f}} \cap E' = \emptyset$. If a_1 is in fact a bypass argument, then its original is either a_0 or is contained in E', and thus in E. Consequently, it will not be in E^+ in EFC. Let us assume that it is not a bypass argument. From the construction of the interpretations and conditions, this means that a given attacker of a_1 is either not present in E, or that the transformed defense attacking set for it is contained in $E' \cup \{a_0\}$. Therefore, if a_0 is a bypass argument, then the original defense attacking set is in E, and if it is not, it is a subset of $E \cup \{a_0\}$. We can recall that if a_0 is a standard argument, then it is not defeated by any argument in E. Consequently, either no argument in E defeats a_1 (i.e. argument is not attacked at all or a defense attack is present) or for no defeat by E on a_1 there is a reinstatement set on E. Consequently, a_1 does not qualify for E^+ in EFC. We can continue reasoning in this manner till we reach $a_n = b$ and the conclusion that if b has an unblocked partially acyclic evaluation (F', G', B') s.t. $F' \subseteq E'$ in D^{EFC}, then it cannot be in E^+ in EFC. Hence, $E^+ \subseteq E'^{p+} \cap A$.

Let us focus on the other way around. We now need to prove that there is no argument $b \in E'^{p+} \cap A$ in D^{EFC} that is not in E^+ in EFC. Assume it is not the case, i.e. no argument in E defeats$_E$ b or no such defeat has a reinstatement set on E, even though $b \in E'^{p+}$. Let us focus on the first case. If there is no defeat, then there is either no attack on b from E in the first place, or for every attack there is a subset of E carrying out an appropriate defense attack. Consequently, from the analysis of conflict–freeness we can observe that the acceptance condition of b w.r.t. $E' \cap par(b)$ should be mapped to in. Thus, by Proposition 2.150, b could not have been in E'^{p+} and we reach a contradiction with the assumptions. Let us now focus on the case where there is a defeat on b by an argument $d \in E$, but it lacks a reinstatement set on E. By Theorem 2.64, there exists a sequence of distinct defense attacks $((z_1, (x_1, y_1)), ..., (z_n, (x_n, y_n)))$ s.t. $(x_n, y_n) = (d, b)$, each (x_i, y_i) attack is unique, no argument in E defeats$_E$ any element $z \in z_1$, and for every other $(z_i, (x_i, y_i))$ in the sequence, either no argument $h \in E$ defeats$_E$ any element $z \in z_i$ or for every such defeat there is an argument $l \in A$ s.t. $(l, (h, z) \in \{(z_1, (x_1, y_1)), ..., (z_{i-1}, (x_{i-1}, y_{i-1}))\}$.

Let us start with the set z_1. By previous reasoning, we can observe that if E does not defeat$_E$ any argument in z_1, then the conditions of the arguments in z_1 are in fact satisfied by E' and thus no element of z_1 is in the partially acyclic discarded set. Furthermore, we can easily construct an unblocked partially acyclic evaluation any $z \in z_1$ with the pd–set in E' and extend it to z^b (if it exists). Therefore, $z^b \notin E'^{p+}$. Let us now consider z_2 and let $z \in z_2$ be an argument. If it is not defeated$_E$ by E, then we come back to the previous case and can show that z cannot be in the partially acyclic discarded set. If it is defeated$_E$, then the condition of z is out w.r.t. E'. However, we can observe that by the construction, the condition of z w.r.t. $E' \cup z_1 \cup z_1^b$ is in, and as no element in $z_1 \cup z_1^b$ is in the discarded set, then the argument cannot be decisively out w.r.t. the partially acyclic range. Thus, it is not in the partially acyclic discarded set by Proposition 2.150. We can therefore show that $z_2 \cap E'^{p+} = \emptyset$ and that $z_2^b \cap E'^{p+} = \emptyset$. We can continue this line of reasoning until we reach z_n and the result that $z_n \cap E'^{p+} = \emptyset$ and $z_n^b \cap E'^{p+} = \emptyset$. Consequently, y_n cannot be decisively out w.r.t. the partially acyclic range either and $y_n = b \notin E'^{p+}$. We reach a contradiction with the assumptions and can conclude that $(E^{p+} \cap A) \subseteq E^+$.

We have now shown that the set of arguments defeated by E with a reinstatement set on E in EFC equals the partially acyclic discarded set (without bypasses) of $E' = E \cup E^b$ in D^{EFC}. The bypass arguments cannot appear in **f** parts of minimal decisively in interpretations; they also do not possess a minimal decisively in interpretation with a non empty **f** part. Therefore, as long as two conflict–free sets of D^{EFC} differ only by the bypass arguments they contain, then their partially acyclic discarded sets are the same. Furthermore, if an argument is in the discarded set, then so is its source, and E'^{p+} can be also described with $E^+ \cup (E^+)^b$. We now need to prove that an argument $a \in A$ is defended by E in EFC iff it is decisively in w.r.t. the partially acyclic range interpretation of $E' = E \cup E^b$ in D^{EFC}.

Let us start with left to right direction. If an argument a is defended by E, then every argument $b \in E$ s.t. b defeats$_E$ a, is in turn defeated with reinstatement by E. Therefore, a is defended if every argument $b \in A$ defeating it is in E^+. Let us now consider an argument c s.t. $(c, a) \in R$, but c does not defeat$E$ a. This means that there is a suitable defense attack carried out by a set $F \subseteq E$. We can now shift to D^{EFC}. Every attacker of a, be it b or c style, has a corresponding att formula in the condition and thanks to our transformation, it is not equivalent to $\top$ due to inconsistencies. If it is a formula att_a^b, then we can observe that as b is mapped to **f** by the partially acyclic discarded range, the formula evaluates to true under this range and will remain such independently of what is assigned to the remaining arguments in the formula. If it is a formula att_a^c, then the disjunction of conjunctions corresponding to the transformed defense attackers evaluates to true due to the presence of bypass arguments and thus the whole att_a^c is true. Moreover, it will stay such, no matter what new arguments come into play. Consequently, the condition of a is in under the partially acyclic range and will remain in for any of its completions to A. Thus, a is decisively in w.r.t. the range. We can also observe that if a is decisively in w.r.t. the partially acyclic range of E', then a^b is decisively in w.r.t. the partially acyclic range of $E' \cup \{a\}$.

Let us continue with the right to left direction. If an argument a is decisively in w.r.t. the partially acyclic range, then its condition is in w.r.t. every completion of the range to A. This means that every $att_a^b = \neg b \vee (\bigwedge B'_1 \vee ... \bigwedge B'_m)$ subformula of the acceptance condition evaluates to true under the acyclic range and remains such under every completion. Therefore, due to the fact that b does not appear in any of the B' sets, it is either b that has to be assigned **f** by the range or at least one set B'_i has all arguments assigned **t** by the range. If it is the first case, then $b \in E^+$ and if the attack from b is a defeat, then a is defended from b by E in EFC. If it is the latter, then we can observe that the attack from b on a does not become a defeat – since the set carrying out defense attack represented by B' is in E', then so is its original version. As the att subformulas account for all attackers of a, we can conclude that E defends a.

Using the proved properties it can be shown that if $E \subseteq A$ is a σ–extension of EFC, where $\sigma \in \{$admissible, complete preferred$\}$, then $E \cup E^b$ $ca_2 - \sigma$–extension of D^{EFC} and that if $E' \subseteq A'$ is a $ca_2 - \sigma$–extension of D^{EFC}, then $E' \cap A$ is a σ–extension of EFC. Furthermore, we can observe that due to the fact that decisiveness of an argument propagates to its bypass and a bypass cannot appear without its source, there is a one to one relation between the complete (and thus, preferred) extensions of EFC and D^{EFC}. This is not necessarily the case for the admissible extensions – not always a bypass has to and needs to be included. What remains to be proved is the relation between stable extensions and models, and the grounded and acyclic grounded extensions.

Let $E \subseteq A$ be a stable extension of EFC. This means it is conflict–free and defeats$_E$ every argument $a \in A \setminus E$. We can observe that every defeat$_E$ originating from E will be a trivial reinstatement set for any of these defeats$_E$. Therefore, from the previous parts of this proof it follows that $E' = E \cup B$ is conflict–free in D^{EFC} and that every argument $a \in A \setminus E$ is in the partially acyclic discarded set. Furthermore, so is every argument a^b, and thus $E' = A' \setminus E'^{p+}$. By Proposition 2.150 it holds that for every such a, $C_a(E \cap par(a)) = out$. Therefore, E' is a model of D^{EFC}. As observed in Example 103, E' does not need to be stable in D^{EFC}.

Let $E' \subseteq A'$ be a stable extension of D^{EFC}. By previous parts of the proof, $E = E' \cap A$ is conflict–free in EFC. By Lemma 2.159, from the fact that E' is a model it follows that every argument in $A' \setminus E'$ is in the partially acyclic discarded set. Consequently, every argument in $A \setminus E$ is in E^+ in EFC, and by the definition of this set is defeated$_E$ by E. Therefore, E is stable in EFC.

In order to show that the grounded extension in EFC and the acyclic grounded in D^{EFC} correspond, we can use the iterating from the empty set approach. Let us start with $E = E' = \emptyset$. The set E is conflict–free in EF and E' is pd–acyclic conflict–free in D^{EFC}. They are also (aa–) admissible in their respective frameworks. Since E' is pd–acyclic conflict–free, then the partially acyclic range of E' is in fact acyclic by Lemma 2.132. Therefore, if we perform an iteration and add to E the arguments it defends in EFC and to E' those that are decisively in w.r.t. the acyclic range of E' in D^{EFC}, then it is still the case that $E = E'$. Moreover, by Lemma 2.154, E' is still aa–admissible and thus pd–acyclic conflict–free. From the admissibility of E' follows the admissibility of E. Let us now repeat the iteration on E' by limiting ourselves to bypass arguments. From the previous parts of the proof it should be clear that it is the $(E')^b = E^b$ arguments that will be added. Again, E' is still aa–admissible and pd–acyclic conflict–free. We can now repeat the iterations (including the extra step for E') and observe that $E' = E \cup \{b\}$. We can continue in this manner until there are no arguments left (and as we are working with finite frameworks, this is warrantied) and observe that $E' = E \cup \{b\}$ and E is grounded in EFC while E' is acyclic grounded in D^{EFC}. $\square$

15.7 Translating BAFs: Proof Appendix

Theorem 9.2. *Let $BF = (A, R, S)$ be a deductive BAF and $R' = \{R^{sup}, R^{med}_{R^{sup}}\} \subseteq R^{ind}$ the collection of supported and super–mediated attacks in BF. Let $apd - F^{BF} = (A, R \cup \bigcup R')$ be the associated attack propagation AF obtained through Translation 53 and $E \subseteq A$ a complete extension of $apd - F^{BF}$. Then, E is closed under S in BF.*

Proof. Let $E \subseteq A$ be a complete extension of $apd - F^{BF}$. Let us assume it is not closed under S, i.e. there exist arguments $a \in E, b \in A \setminus E$ s.t. aSb in BF. As b is not in

E, this means E does not defend b in $apd - F^{BF}$, i.e. there exists an argument $c \in A$ s.t. $(c, b) \in R \cup \bigcup R'$, but no $d \in E$ s.t. $(d, c) \in R \cup \bigcup R'$. Let us analyze the case where $(c, b) \in R$. As $(a, b) \in S$, there exists a (super) mediated attack from c to a and thus $(c, a) \in R \bigcup R'$. Consequently, if there is no argument d attacking c, we breach the admissibility of E. If $(c, b) \in R^{sup}$, then there exists a super mediated attack from c to a again and it contradicts the admissibility of E. Finally, let $(c, b) \in R^{med}_{R^{sup}}$. This means there exists an argument $e \in A$ s.t. b supports e and c (supported) attacks e. Therefore, there is a chain of support from a to e as well, and again it has to be the case that c super mediated attacks a. We breach the admissibility of E. Hence, we can conclude that if E is complete in $apd - F^{BF}$, then it is closed in BF. □

Theorem 9.3. *Let $BF = (A, R, S)$ be a BAF and $R' = \{R^{sec}\} \subseteq R^{ind}$ the collection of secondary attacks in BF. Let $apd - F^{BF} = (A, R \cup \bigcup R')$ be the associated attack propagation AF obtained through Translation 53 and $E \subseteq A$ a complete extension of $apd - F^{BF}$. Then, E is inverse closed under S in BF.*

Proof. Let $E \subseteq A$ be a complete extension of $apd - F^{BF}$. Let us assume it is not inverse closed under S, i.e. there exist arguments $a \in A \setminus E, b \in E$ s.t. aSb in BF. As a is not in E, this means that E does not defend a in $apd - F^{BF}$, i.e. there exists an argument $c \in A$ s.t. $(c, a) \in R \cup \bigcup R'$, but no $d \in E$ s.t. $(d, c) \in R \cup \bigcup R'$. If $(c, a) \in R \cup R^{sec}$, then there exists a secondary attack from c to b as well and thus $(c, b) \in \bigcup R'$. Consequently, if there is no argument d attacking c, we breach the admissibility of E. Additionally, we can observe that if $c = a$, then d would be secondary attacking b and thus breaching the conflict–freeness of E. Therefore, we can conclude that if E is complete in $apd - F^{BF}$, then it is inverse closed under S in BF. □

Theorem 9.4. *Let $BF = (A, R, S)$ be a BAF, $R' \subseteq R^{ind}$ a collection of indirect attacks in BF and $iclo - F^{BF} = (A', R'')$ its associated inverse closure attack propagation–defender AF w.r.t. R' obtained through Translation 54. A set of arguments $E \subseteq A$ is i–admissible (i–preferred) in BF w.r.t. (R', R') iff it is admissible (preferred) in $iclo - F^{BF}$.*

Proof. Let $E \subseteq A$ be an i–admissible extension of BF w.r.t. (R', R'). This means it is +conflict–free w.r.t. R' in BF, defends all of its members w.r.t. R' and for every $b \in E, a \in A$ s.t. aSb, $a \in E$. E can be easily shown to be conflict–free in $iclo - F^{BF}$ – all the attacks not covered by $R \cup \bigcup R'$ are not within $A \times A$. Let us now show that every argument $a \in A$ is defended by E in $iclo - F^{BF}$. Let $b \in A'$ be an attacker of a in R''. Due to i–admissibility of E, if $b \in A$, then the (b, a) conflict has to be in the $R \cup \bigcup R'$ part of R'', and thus b is a (possibly) indirect attacker of a in BF. Therefore, there exists $c \in E$ s.t. $(c, b) \in R \cup \bigcup R'$ by the i–admissibility of E in BF, and E can defend a against any attacker $b \in A$ in $iclo - F^{BF}$. Let us now focus on the case where $b \in S$. This means it is of the form (d, a) for an argument $d \in A$. Since E is inverse closed in BF, it has to be the case that $d \in E$. Consequently, d attacks b in R'', and again E can defend a against it. We can therefore conclude that E is an admissible extension of $iclo - F^{BF}$.

Let $E \subseteq A'$ be an admissible extension of $iclo - F^{BF}$. We can observe that all arguments in S are self–attackers. Consequently, $E \subseteq A$. Since $R \cup \bigcup R' \subseteq R''$, then if E is conflict–free in $iclo - F^{BF}$, it has to be conflict–free w.r.t. R' in BF. Let $a \in E, b \in A'$ be arguments s.t. $(b, a) \in R''$. Therefore, by admissibility of E in $iclo - F^{BF}$, there has to be $c \in E$ s.t. $(c, b) \in R''$. If $b \in S$, then by the construction of R'', cSa. Hence, it can be shown that E satisfies the inverse closure requirements in BF. If $b \in A$, then $(b, a) \in R \cup \bigcup R'$ and $(c, b) \in R \cup \bigcup R'$ (note that $c \in A$). Consequently, E defends a in BF. We can therefore conclude that E is i–admissible in BF.

In both cases, the i–preferred and preferred semantics are defined as the maximal i–admissible and admissible extensions. Hence, by the previous parts of this proof, E is i–preferred in BF iff it is preferred in $iclo - F^{BF}$. □

Theorem 9.9. *Let $BF = (A, R, S)$ be a deductive BAF and $R' = \{R^{sup}, R^{med}_{R^{sup}}\} \subseteq R^{ind}$ the collection of supported and super–mediated attacks in BF. Let $apd - F^{BF} = (A, R \cup \bigcup R')$ the associated attack propagation AF and $cd - F^{BF} = (A', R'')$ the associated coalition AF obtained through Translations 53 and 57. If set $E = \{a_1, ..., a_n\} \subseteq A$ is a σ–extension of $apd - F^{BF}$, where $\sigma \in \{$complete, preferred, grounded, stable$\}$, then $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ is a σ–extension of $cd - F^{BF}$. If set $E' \subseteq A'$ is a σ–extension of $cd - F^{BF}$, then $E = \bigcup E'$ is a σ–extension of $apd - F^{BF}$.*

Proof. In order to prove this theorem, we will be relying on Theorem 9.2, which shows that if E is complete in $apd - F^{BF}$ and $a \in E$ supports an argument b in BF, then $b \in E$.

Let $E = \{a_1, ..., a_n\} \subseteq A$ be a complete extension of $apd - F^{BF}$. Assume that $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ is not conflict–free in $cd - F^{BF}$. By using Theorem 9.2 we can show that $\bigcup E' = E$. Therefore, if there exist $C(a_i), C(a_j) \in E'$ s.t. $C(a_i)R''C(a_j)$, then there have to be arguments $a, b \in E$ s.t. aRb. We breach the conflict–freeness of E in $apd - F^{BF}$.

Let us now assume that E' is not admissible in $cd - F^{BF}$. Thus, there exist $C' \in A', C(a_j) \in E'$ s.t. $C'R''C(a_j)$ and no $C(a_k) \in E'$ for which $C(a_k)R''C'$. Consequently, there are arguments $a \in C' \subseteq A, b \in E$ s.t. $(a, b) \in R$. However, by the admissibility of E, there must be an argument $c \in E$ s.t. $(c, a) \in R \cup \bigcup R'$. If $(c, a) \in R$, then $C(c)R''C'$ and as $C(c) \in E'$, we reach a contradiction. If $(c, a) \in R^{sup}$, then there exists $d \in A$ s.t. $(d, a) \in R$ and c supports d. By completeness of E in $apd - F^{BF}$, $d \in E$. Therefore, $C(d) \in E'$ and $C(d)R''C'$. We reach a contradiction. If $(c, a) \in R^{med}_{R^{sup}}$, then there exists $e \in A$ s.t. a supports e and c (supported) attacks e. Thus, by previous analysis, there is an argument in $d \in E$ directly attacking e and as e has to be present in C', we can conclude that $C(d)R''C'$. Therefore, E' is admissible in $cd - F^{BF}$.

Finally, let us assume that E' is not complete in $cd - F^{BF}$. Thus, there exists a coalition argument $C' \in A' \setminus E'$ that is defended by E' in $apd - F^{BF}$, i.e. for every argument $C'' \in A'$ s.t. $C''R''C'$, there exists $C(a_i) \in E'$ for which $C(a_i)R''C''$. We can observe it has to be the case that there is at least one argument $a \in C'$ s.t. $a \notin E$ – otherwise, C' would have been included in E' by construction. By completeness of E, it cannot be the case that E defends a in $apd - F^{BF}$. This means there is an argument $b \in A$ s.t.

$(b, a) \in R \cup \bigcup R'$ and no argument $c \in E$ s.t. $(c, b) \in R \cup \bigcup R'$. By the previous analysis it thus holds that $C(b)R''C'$. Moreover, if no argument $c \in E$ attacks $C(b)$, then by the fact that $\bigcup E' = E$, there cannot be any $C(a_i) \in E'$ s.t. $C(a_i)R''C(b)$. We reach a contradiction and can therefore conclude that E' is a complete extension of $cd - F^{BF}$.

Let $E' \subseteq A'$ be a complete extension of $cd - F^{BF}$. Let us assume that $E = \bigcup E'$ is not conflict–free in $apd - F^{BF}$. Thus, there exist arguments $a, b, c, d \in E$ s.t. 1. $(a, b) \in R$, or 2. a supports c and $(c, d) \in R$, or 3. a supports c, b supports d and $(c, d) \in R$, or finally 4. b supports d and $(a, d) \in R$. Let $C \in E'$ be a coalition that contains a and $C' \in E'$ the coalition that contains b. If we are dealing with first case, there is a conflict between the coalitions containing a and b in R'' and we breach the conflict–freeness of E'. If it is the second case, then $c \in C$ and we come back to the first case. If it is the third case, then $c \in C$ and $d \in C'$ and we again have an attack in E'. If it is the final (fourth) case, then $d \in C'$ and we again reach a contradiction with the assumptions. Thus, E has to be conflict–free in $apd - F^{BF}$.

Let us now assume that E is not admissible, i.e. there exist arguments $a \in A, b \in E$ s.t. $(a, b) \in R \cup \bigcup R'$ and no $c \in E$ s.t. $(c, a) \in R \cup \bigcup R'$. From the previous analysis it follows that a coalition $C(a) \in A'$ for a has to attack a coalition $C' \in E'$ containing b. Therefore, by the admissibility of E', there must exist a coalition $C'' \in E'$ s.t. $C''R''C(a)$. By the construction of coalitions, this means there is an argument $d \in E$ that either (directly) attacks a or an argument supported by a. Hence, either $(d, a) \in R$ or there is a mediated attack from d to a. We reach a contradiction and can conclude that E is admissible in $apd - F^{BF}$.

Finally, let us assume that E is not complete in $apd - F^{BF}$. Consequently, it has to be the case there is an argument $b \in A \setminus E$ that is defended by E in $apd - F^{BF}$. At the same time, by the completeness of E', it cannot be the case that E' defends $C(b)$. Therefore, there exists a coalition $C \in A'$ s.t. $CR''C(b)$ and no coalition $C' \in E'$ s.t. $C'R''C$. By the previous analysis, we can observe that there must be an argument $c \in C$ that is attacking b in $R \cup \bigcup R'$. If there was an argument $d \in E$ directly or indirectly attacking c, then again from the previous parts of this proof we can conclude that the coalition in E' containing d would attack C in R''. We reach a contradiction. Thus, E is complete in $apd - F^{BF}$.

We can observe that if two coalitions $C, C' \subseteq A$ are defended by a given set, then so is any coalition $C'' \in A'$ s.t. $C'' \subseteq C \cup C'$. This simply follows from the construction of $cd - F^{BF}$. Therefore, it cannot be the case that there exist two different extensions $E, E' \subseteq A'$ of $cd - F^{BF}$ s.t. $\bigcup E = \bigcup E'$. Thus, it can be shown that there is a one–to–one relation between the complete extensions of $cd - F^{BF}$ and $apd - F^{BF}$.

Due to the one–to–one correspondence between the complete extensions of $cd - F^{BF}$ and $apd - F^{BF}$ and the fact that a subset relation between two complete extensions of one structure implies the same subset relation between the corresponding extensions of the other, we can use Theorem 2.10 to prove the same correspondence between the preferred and grounded extensions as in the complete case. Using Theorem 2.9 and the previous analysis on attacks, the same relation between the stable extensions of both framework can be shown. □

Theorem 9.10. *Let $BF = (A, R, S)$ be a deductive BAF, $R' = \{R^{sup}, R^{med}_{R^{sup}}\} \subseteq R^{ind}$ the collection of supported and super–mediated attacks in BF and $cd - F^{BF} = (A', R'')$ the associated coalition AF obtained through Translation 57. The following holds:*

- *if set $E = \{a_1, ..., a_n\} \subseteq A$ is +conflict–free w.r.t. R' and closed under S in BF, then $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ is a conflict–free extension of $cd - F^{BF}$.*
- *if set $E = \{a_1, ..., a_n\} \subseteq A$ is a c–admissible (c–preferred) extension of BF w.r.t. (R', R'), then $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ is an admissible (preferred) extension of $cd - F^{BF}$.*
- *if set $E = \{a_1, ..., a_n\} \subseteq A$ is a d–complete extension of BF w.r.t. (R', R'), then $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ is a complete extension of $cd - F^{BF}$.*
- *if set $E = \{a_1, ..., a_n\} \subseteq A$ is a d–grounded (stable) extension of BF w.r.t. R', then $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ is a grounded (stable) extension of $cd - F^{BF}$.*
- *if set $E' \subseteq A'$ is a conflict–free extension of $cd - F^{BF}$, then $E = \bigcup E'$ is +conflict–free w.r.t. R' and closed under S in BF.*
- *if set $E' \subseteq A'$ is an admissible (preferred) extension of $cd - F^{BF}$, then $E = \bigcup E'$ is a c–admissible (c–preferred) extension of BF w.r.t. (R', R').*
- *if set $E' \subseteq A'$ is a complete extension of $cd - F^{BF}$, then $E = \bigcup E'$ is a d–complete extension of BF w.r.t. (R', R').*
- *if set $E' \subseteq A'$ is a grounded (stable) extension of $cd - F^{BF}$, then $E = \bigcup E'$ is a d–grounded (stable) extension of BF w.r.t. R'.*

Proof. Let $E = \{a_1, ..., a_n\}$ be +conflict–free w.r.t. R' and closed under S in BF and let $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ be the corresponding set in $cd - F^{BF}$. Due to the fact that E is closed under S, we can observe that $E = \bigcup E'$. If E' is not conflict–free, then by the construction of $cd - F^{BF}$, it can be shown that E has to contain two arguments directly attacking each other (see proof of Theorem 9.9). Thus, we contradict the +conflict–freeness of E, and can conclude that E' is conflict–free in $cd - F^{BF}$.

Let $E' = \{C(a_1), C(a_2), ..., C(a_n)\}$ be a conflict–free extension of $cd - F^{BF}$ and let $E = \bigcup E'$ be the corresponding set in BF. By the construction of $cd-F^{BF}$ and coalitions, it can be shown that E has to be closed under S. Let us assume it is not +conflict–free in BF w.r.t. R'. It means there are two arguments $a, b \in E$ s.t. a directly, supported or super mediated attacks b. However, as seen in the proof of Theorem 9.9, the fact that E is closed under S leads to the conclusion that there must be some direct attack (c, d) for $c, d \in E$. Consequently, there is a conflict between the coalitions in E' that brought c and d to E, and we reach a contradiction. We can conclude that E is +conflict–free and closed under S in BF.

In order to show admissibility, we can use the presented conflict–freeness analysis and the proof of Theorem 9.9. The results for complete, grounded and stable semantics follow from Theorems 9.1 and 9.9. The results for preferred semantics follow from Theorems 9.1, 9.2 and 9.9. $\square$

Theorem 9.12. *Let $BF = (A, R, S)$ be a BAF specialized for necessary support, $R' = \{R^{sec}, R^{ext}\}$ the collection of secondary and extended attacks in BF and $dn - F^{BF} = (A', R'')$ the defender AF associated with BF obtained through Translation 58. If $E \subseteq A$ is an i–admissible extension of BF w.r.t. (R', R'), then there exists an admissible extension $E' \subseteq A'$ s.t. $E' \cap A = E$. If $E' \subseteq A'$ is an admissible extension of $df - F^{BF}$, then $E = E' \cap A$ might not be an i–admissible extension of BF w.r.t. (R', R').*

Proof. Let $E \subseteq A$ be an i–admissible extension of BF w.r.t. (R', R'). With $E^+ = \{a \mid a \in A, \exists b \in E \text{ s.t. } (b, a) \in R \cup \bigcup R'\}$ be the set of arguments (directly or indirectly) attacked by E w.r.t. R'. It is easy to see that if E is conflict–free w.r.t. R', then $E \cap E^+ = \emptyset$. By $S(a) \subseteq S$ we will denote the set of all (direct) supports carried out by a. Let now $E' = E \cup \bigcup\{S(a) \mid a \in E^+\}$ be a set of arguments in $dn - F^{BF}$.

Let us assume E' is not conflict–free in $dn - F^{BF}$ and let $a, b \in E'$ be two arguments s.t. $(a, b) \in R''$. Due to the fact that $R \subseteq R''$, it cannot be the case that $a, b \in E' \cap A$ – otherwise, we would breach the +conflict–freeness of E. By the construction of E', it is also easy to see that it cannot be the case that $a \in E' \cap A$ and $b \in E' \cap A'$. This leaves us with the case that $a \in E' \cap A'$ and $b \in E' \cap A$. In other words, a represents a support link targeted at b. However, this means that the source of this support is in E^+; therefore, by the properties of secondary attack, b is also in E^+ and therefore cannot be in E due to conflict–freeness. We reach a contradiction and can conclude that E' is conflict–free in $dn - F^{BF}$.

Let us now assume that E' is not admissible in $dn - F^{BF}$. This means there are arguments $b \in A'$, $a \in E'$ s.t. $(b, a) \in R''$ and no argument $c \in E'$ s.t. $(c, b) \in R''$. Let us first focus on the case where both a and b are normal arguments in A. Thus, $(b, a) \in R$, and it has to be the case that E defends a in BF. Consequently, there is an argument $d \in E$ s.t. $(d, b) \in R \cup \bigcup R''$. If $(d, b) \in R$, then $(d, b) \in R''$, and we reach a contradiction. If $(d, b) \in R^{sec}$, then there is an argument $e \in A$ s.t. eSb and $e \in E^+$. Therefore, the (e, b) support is in E', and attacks b. We reach a contradiction. If $(d, b) \in R^{ext}$, then there is an argument $f \in A$ s.t. f supports (directly or indirectly) b and $(f, b) \in R$. As E is inverse closed under support, $f \in E$. Thus $f \in E'$ and $(f, b) \in R''$. We reach a contradiction yet again. Let us now focus on the case where a is a normal argument and b represents the support relation (x, a) targeted at a by an argument $x \in A$. If b is not attacked by E', then it means that $x \notin E'$ and $x \notin E$. Since xSa, it cannot be the case that E is inverse closed under support. We reach a contradiction. Finally, let us assume that b is a normal argument and a represents the support relation (b, x) carried out at b on an argument $x \in A$. This means that E' does not attack b in $dn - F^{BF}$. Therefore, no support argument targeted at b is in E' and no argument y s.t. yRb can be in E' either. If it is the first case, then no supporter of b is attacked, and thus E cannot carry out a secondary attack at b. From the latter, it cannot carry out a direct one either, and due to the fact that E is inverse closed, it cannot carry out an extended attack as well. Thus, E cannot defend a in BF and we reach a contradiction with the i–admissibility of E. We can finally conclude that E' is admissible in $dn - F^{BF}$.

Please see Example 111 for a proof that not every admissible extension of $dn - F^{BF}$ produces an i–admissible extension of BAF w.r.t. (R', R'). □

Theorem 9.14. *Let $BF = (A, R, S)$ be a support acyclic BAF, $R' = \{R^{sec}\}$ the collection of secondary attacks in BF and $FN^{BF} = (A, R, N)$ the AFN associated with BF obtained through Translation 59. Then, a set $E \subseteq A$ is:*

- *+conflict–free w.r.t. $\emptyset$ in BF iff it is conflict–free in FN^{BF}.*
- *inverse closed under S in BF iff it is coherent in FN^{BF}.*
- *+conflict–free w.r.t. R' and inverse closed under S in BF iff it is strongly coherent in FN^{BF}.*
- *an i–admissible extension of BF w.r.t. (R', R') iff it is admissible in FN^{BF}.*
- *an i–preferred extension of BF w.r.t. (R', R') iff it is preferred in FN^{BF}.*
- *a d–complete extension of BF w.r.t. (R', R') iff it is complete in FN^{BF}.*
- *a d–grounded extension of BF w.r.t. R' iff it is grounded in FN^{BF}.*
- *a stable extension of BF w.r.t. R' iff it is stable in FN^{BF}.*

Proof. For the sake of simplicity, we will not be making the parametrization of BAF semantics explicit in the proof.

- Conflict–freeness in AFNs is defined on the direct attacks only, and so is +conflict–freeness w.r.t. $\emptyset$ in BAFs. Thus, it is easy to show that these extensions coincide.
- If E is inverse closed, then for every argument $a \in E$, if bSa, then $b \in E$. Therefore, for every argument $a \in E$ and every set $B \subseteq A$ s.t. BNa, $B \cap E \neq \emptyset$. From Theorems 9.13 and 4.32 it thus follows that E is coherent in FN^{BF}. The other way around is straightforward.
- If E +conflict–free w.r.t. R' and inverse closed under S, then by the previous point it is coherent in FN^{BF}. As there are no arguments $a, b \in E$ s.t. $(a, b) \in R \cup \bigcup R'$, then it is also trivially conflict–free and thus strongly coherent. Let now E be strongly coherent in FN^{BF}. By the previous point, it is inverse closed under S. Assume it is not +conflict–free w.r.t. R'. We can observe it can only be the case that there are $a, b \in E$ s.t. $(a, b) \in R^{sec}$. This means that there is an argument c s.t. c supports b and $(a, c) \in R$. However, as the set is inverse closed, then $c \in E$. Therefore, we breach the conflict–freeness of E in FN^{BF}. We can conclude that E is +conflict–free and inverse closed under S in BF.

- Let E be i–admissible in BF. Let us assume it is not admissible in FN^{BF}. By the previous points, it suffices to show that there are arguments $a \in E, b \in A$ s.t. $(b, a) \in R$ and a coherent set B containing b s.t. no element in it is attacked by E in R in FN^{BF}. We can observe that the same (b, a) attack is in BF and due to i–admissibility, it has to be the case that there is $c \in E$ s.t. $(c, b) \in R \cup R^{sec}$. If $(c, b) \in R$, then E can easily defend a in FN^{BF} and we reach a contradiction. If $(c, b) \in R^{sec}$, then there exists an argument $d \in A$ s.t. $(c, d) \in R$ and d supports b in S. Due to the fact that FN^{BF} is binary and strongly valid, it is easy to observe that b will possess only one minimal powerful sequence and that this sequence will contain d. Therefore, E can attack this sequence in FN^{BF} and we can finally conclude that the set is admissible in FN^{BF}.

  Let E be admissible in FN^{BF}. Let us assume it is not i–admissible in BF. Again, by the previous points, it suffices to show that there are arguments $a \in E, b \in A$ s.t. $(b, a) \in R \cup R^{sec}$ and no argument $c \in E$ s.t. $(c, b) \in R \cup R^{sec}$. Let us first assume that $(b, a) \in R$. This means that the attack occurs also on the AFN side and by the admissibility of E in FN^{BF}, E has to attack every coherent set for b. Without the loss of generality, we can focus on the minimal coherent sets, and by the construction of FN^{BF} we know that the single minimal powerful sequence for b consists all of those arguments directly or indirectly supporting b in S (plus b itself). Thus, if E contains an argument d attacking any of these arguments in FN^{BF}, then d carries out a directed or secondary attack at b in BF. Thus, E defends a in BF and we reach a contradiction. Let us now assume that $(b, a) \in R^{sec}$. Due to the fact that E is inverse closed for S, there exists an argument e in E s.t. e supports b and $(b, a) \in R$. We can thus repeat the previous analysis to show that E directed or secondary attacks b. We reach a contradiction yet again and can thus conclude that E is i–admissible in BF.

- Since admissible and i–admissible extensions coincide, so do the preferred and i–preferred ones.

- Let E be a d–complete extension of BF. By definition, it is also d–admissible, and by Theorem 9.3, also i–admissible. Thus, by previous analysis, E is an admissible extension of FN^{BF}. If it is not complete, then there is an argument $a \in A \setminus E$ that is defended by E. This means that for every set of arguments $C \subseteq A$ s.t. CNa, $C \cap E \neq \emptyset$, and that for every argument $b \in A$ s.t. $(b, a) \in R$, E attacks all coherent sets for b. By the construction of FN^{BF}, it holds that for every $c \in A$ s.t. cSa, $c \in E$. By the previous analysis, we can also observe that E contains arguments directly or secondary attacking all direct attackers of a in BF. Moreover, due to the fact that the support arguments are in E and E is i–admissible, we cover also the indirect attackers of a in BF as well. Thus, we contradict the d–completeness of E in BF, and it has to be the case that E is complete in FN^{BF}.

  Let E be a complete extension of FN^{BF}. By the previous analysis, E is i–

admissible in BF, and thus d–admissible as well. If it is not d–complete, then there exists an argument $b \in A \setminus E$ defended by E in BF. Thus, from the previous analysis we can observe that if there is an argument $c \in A$ s.t. $(c, b) \in R$, then every coherent set for c is attacked by E in FN^{BF}. What is important is the fact that by the proof of Theorem 9.3, if b is defended by E in BF, then so is every argument supporting it. Due to the fact that the support subgraph is directed acyclic, we can form a topological ordering of supporters $(b_0, ..., b_n)$ of b in BF that behaves like the minimal powerful sequence for b in FN^{BF}. As b_0 requires no support and by the explanation above has to be defended against attacks in FN^{BF}, E defends b_0 in FN^{BF}. Therefore, $b_0 \in E$, and b_1 is both sufficiently supported and defended from attacks by E. Consequently, E defends b_1 in FN^{BF}. We can continue this line of reasoning until we reach $b_n = b$ and conclude that E defends b. As $b \notin E$, we reach a contradiction with the completeness of E in FN^{BF}. Thus, E is d–complete in BF.

- Follows from the relation between the complete and d–complete extensions and Theorems 2.80 and 2.95.
- Follows straightforwardly from the previous parts of this proof and Theorem 2.94.

□

Theorem 9.15. *Let $BF = (A, R, S)$ be a support acyclic BAF and let $ES^{BF} = (A \cup \{\eta\}, R', E)$ be its associated EAS obtained through Translation 60. ES^{BF} is attack binary, support singular and all–supported. It is in minimal, weakly, relation and strongly valid forms.*

Proof. The fact that the target framework is attack binary, support singular and all–supported follows straightforwardly from the translation. It is also in minimal normal form based on Lemma 4.73. Since ES^{BF} is singular, we can observe that there exists only one support assigning function f in accordance with Definition 4.35. Due to the fact that (A, S) is directed acyclic, there exists a topological ordering of the arguments. This ordering (with η added at the beginning) will be a topological ordering for the arguments in ES^{BF}. Therefore, by Theorem 4.38, our framework is strongly valid. By Theorem 4.36, it is also weakly and relation valid. □

Theorem 9.16. *Let $BF = (A, R, S)$ be a support acyclic BAF, $ES^{BF} = (A \cup \{\eta\}, R', N)$ the EAS associated with BF obtained through Translation 60 and $R'' = \{R^{sec}\}$ the collection of secondary attacks in BF. Then, a set $X \subseteq A$ is:*

- *+conflict–free w.r.t. $\emptyset$ in BF if it is conflict–free in ES^{BF}.*
- *inverse closed under S in BF if $X \cup \{\eta\}$ is self–supporting in ES^{BF}.*
- *+conflict–free w.r.t. R'' and inverse closed under S in BF if $X \cup \{\eta\}$ is strongly self–supporting in ES^{BF}.*

- *an i–admissible extension of BF w.r.t. (R'', R'') if $X \cup \{\eta\}$ is admissible in ES^{BF}.*
- *an i–preferred extension of BF w.r.t. (R'', R'') iff $X \cup \{\eta\}$ is preferred in ES^{BF}.*
- *a d–complete extension of BF w.r.t. (R'', R'') iff $X \cup \{\eta\}$ is complete in ES^{BF}.*
- *a d–grounded extension of BF w.r.t. R'' iff $X \cup \{\eta\}$ is grounded in ES^{BF}.*
- *a stable extension of BF w.r.t. R'' iff $X \cup \{\eta\}$ is stable in ES^{BF}.*

Additionally, a set $X' \subseteq A \cup \{\eta\}$ is:

- *conflict–free in ES^{BF} if $X' \cap A$ is +conflict–free w.r.t. $\emptyset$ in BF.*
- *self–supporting in ES^{BF} if $X' \cap A$ is inverse closed under S in BF.*
- *strongly self–supporting in ES^{BF} if $X' \cap A$ is +conflict–free w.r.t. R'' and inverse closed under S in BF.*
- *admissible in ES^{BF} if $X' \cap A$ is an i–admissible extension of BF w.r.t. (R'', R'').*

Proof. For the sake of simplicity, we will not be making the parametrization of BAF semantics explicit in the proof.

Let $X \subseteq A$ be a +conflict–free w.r.t. $\emptyset$ extension of BF. This means there are no arguments $a, b \in X$ s.t. $(a, b) \in R$. Therefore, there are no $a, b \in X$ s.t. $(\{a\}, b) \in R'$, and as ES^{BF} is attack binary, X is conflict–free in ES^{BF}. Due to the fact that η does not participate in any conflicts, we can observe that $X \cup \{\eta\}$ is conflict–free in ES^{BF} as well.

Let $X' \subseteq A \cup \{\eta\}$ be conflict–free in ES^{BF}. This means there are no $Y \subseteq X', a \in X'$ s.t. $(Y, a) \in R'$. Therefore, by the construction of R', it cannot be the case that there are $a, b \in X' \cap A$ s.t. $(a, b) \in R$. Thus, $X' \cap A$ is +conflict–free in BF w.r.t. $\emptyset$.

If $X \subseteq A$ is inverse closed in BF, then for every argument $a \in X$, if bSa, then $b \in X$. Therefore, for every argument $a \in X$ s.t. $\exists b \in A, (b, a) \in S$, there is a set $Y \subseteq X$ s.t. $(Y, a) \in E$. For every other $a \in X$, $(\{\eta\}, a) \in E$. As η itself requires no support through E, we can observe that every argument $X \cup \{\eta\}$ is sufficiently supported by the set. Thus, $X \cup \{\eta\}$ is self–supporting by Theorem 4.32. Please note that if $X = \emptyset$, then X is self–supporting in ES^{BF} as well.

If $X' \subseteq A \cup \{\eta\}$ is self–supporting in ES^{BF}, then for every non–η argument $a \in X'$, there is a subset $Y \subseteq X'$ s.t. $(Y, a) \in E$. Due to the fact that ES^{BF} is singular and all–supported, it is precisely one set. This means that for every set $Y \subseteq A$ s.t. YEa, $Y \subseteq X'$. Therefore, for every argument $b \in A$ s.t. bSa, $b \in X' \cap A$. Hence, $X' \cap A$ is inverse closed in BF.

Let $X \subseteq A$ be +conflict–free w.r.t. R'' and inverse closed in BF. By the previous parts of this proof, $X' = X \cup \{\eta\}$ is self–supporting. As there are no arguments $a, b \in X$ s.t. $(a, b) \in R \bigcup R''$, then X' is trivially conflict–free and thus strongly self–supporting in ES^{BF}.

Let now $X' \subseteq A \cup \{\eta\}$ be strongly self–supporting in ES^{BF}. By the previous parts of this proof, $X = X' \cap A$ is inverse closed under S in BF. However, assume it is not +conflict–free w.r.t. R''. We can observe it can only be the case that there are $a, b \in E$ s.t. $(a, b) \in R^{sec}$. This means that there is an argument c s.t. c supports b and $(a, c) \in R$. However, as the set is inverse closed, then $c \in X'$. Therefore, we breach the conflict–freeness of X' in ES^{BF}. We can conclude that X is +conflict–free w.r.t. R'' and inverse closed under S in BF.

Let $X \subseteq A$ be i–admissible in BF. Let us assume that $X' = X \cup \{\eta\}$ is not admissible in ES^{BF}. Consequently, there exists an argument $a \in X'$ and a minimal e–supported attack $Y \subseteq A \cup \{\eta\}$ on it s.t. no argument in Y is attacked by X'. We can observe it cannot be the case that $a = \eta$. By the construction of ES^{BF}, every subset of $B \subseteq Y$ that is actually carrying out the conflict in R' consists of a single argument only, i.e. $B = \{b\}$ for some $b \in A$. Due to the fact that η cannot participate in conflicts, the same attack will appear in R in BF. Additionally, since Y contains all supporters of b and none of them are attacked, it cannot be the case that X carries out a direct or secondary attack on b. Due to the fact that $a \in X$, we breach the i–admissibility of X in BF and reach a contradiction. Hence, X' is admissible in ES^{BF}.

Let $X' \subseteq A \cup \{\eta\}$ be admissible in ES^{BF}. Let us assume that $X = X' \cap A$ is not i–admissible in BF. Again, by the previous points, it suffices to show that there are arguments $a \in X, b \in A$ s.t. $(b, a) \in R \cup R^{sec}$ and no argument $c \in X$ s.t. $(c, b) \in R \cup R^{sec}$. Let us first assume that $(b, a) \in R$. This means that the attack occurs also on the EAS side and by the admissibility of X' in ES^{BF}, X' has to attack every self–supporting set for b. Without the loss of generality, we can focus on the minimal self–supporting sets, and by the construction of ES^{BF} we know that the single minimal evidential sequence for b consists all of those arguments directly or indirectly supporting b in S (plus b itself). Thus, if X' contains an argument d attacking any of these arguments in ES^{BF}, then d carries out a directed or secondary attack at b in BF. Since d cannot be η, then $d \in X$. This means that X defends a in BF and we reach a contradiction. Let us now assume that $(b, a) \in R^{sec}$. Due to the fact that X is inverse closed for S, there exists an argument e in X s.t. e supports b and $(b, a) \in R$. We can thus repeat the previous analysis to show that X directed or secondary attacks b. We reach a contradiction yet again and can thus conclude that X is i–admissible in BF.

Based on the previous parts of this proof it is easy to see that if $X \subseteq A$ is i–preferred in BF, then $X' = X \cup \{\eta\}$ is preferred in ES^{BF}. Moreover, if $X' \subseteq A \cup \{\eta\}$ is preferred in ES^{BF}, then $X = X' \cap A$ is i–preferred in BF.

Let $X \subseteq A$ be a d–complete extension of BF. By definition, it is also d–admissible, and by Theorem 9.3, also i–admissible. Thus, we can use the previous analysis to show that $X' = X \cup \{\eta\}$ is an admissible extension of ES^{BF}. If it is not complete, then there is an argument $a \in (A \cup \{\eta\}) \setminus X'$ that is defended by X'. This means that there exists a set of arguments $C \subseteq A \cup \{\eta\}$ s.t. CEa and $C \subseteq X'$, and that for every minimal e–supported attack $B \subseteq A \cup \{\eta\}$ on a, X' a member of B. This, by the construction of ES^{BF} and previous analysis, means that for every $c \in A$ s.t. cSa, $c \in X$, and that for every argument b directly or secondary attacking a, there is an argument in X directly or secondary attacking b. Thus, we contradict the d–completeness of X in BF, and it has to be the case that X' is complete in ES^{BF}.

Let $X' \subseteq A \cup \{\eta\}$ be a complete extension of ES^{BF}. By the previous analysis, $X = X' \cap A$ is i–admissible in BF, and thus d–admissible as well. If it is not d–complete, then there exists an argument $b \in A \setminus X$ defended by X in BF. In other words, every direct or secondary attacker of b is direct or secondary attacked by X. We can reuse the previous parts of this proof to show that every minimal e–supported attack on b is attacked by X'. What is left is the question of support. By the proof Theorem 9.3, if b is defended by X in BF, then so is every argument supporting it. Due to the fact that the support subgraph is directed acyclic, we can form a topological ordering of supporters $(b_0, ..., b_n)$ of b in BF s.t. $(\eta, b_0, ..., b_n)$ behaves like the minimal evidential sequence for b in ES^{BF}. Clearly, $\eta \in X'$. Consequently, X' sufficiently supports b_0 and by the previous parts of this analysis, attacks all minimal e–supported attacks carried out on it. Hence, by the completeness of X', $b_0 \in X'$. We can continue this line of reasoning until we reach $b_n = b$ and conclude that X' defends b in ES^{BF}. As $b \notin X'$, we reach a contradiction with the completeness of X' in ES^{BF}. Thus, X' is d–complete in BF.

It is easy to see that the relation between the complete extensions of BF and ES^{BF} is one–to–one. The argument η is easily defended by any set and thus will be present in every complete extension of ES^{BF}. Consequently, the issue of both $\emptyset$ and $\{\eta\}$ being admissible in ES^{BF} and corresponding to $\emptyset$ in BF is resolved. The correspondence between the complete and d–complete extensions of the two frameworks now follows easily from Theorems 2.80 and 2.112. The relation between the stable extensions can be proved easily as well. □

15.8 Translating AFNs: Proof Appendix

Theorem 10.2. *Let $FN = (A, R, N)$ be an AFN and $F^{FN} = (A', R')$ its corresponding AF built from Translation 61. If $E' \subseteq A'$ is conflict–free in F^{FN}, then $\bigcup E'$ is conflict–free in FN, but not vice versa. A set $E \subseteq A$ is a strongly coherent extension of FN iff F^{FN} admits a set $E' = \{C_1, ..., C_n\} \subseteq A'$ s.t. $E = \bigcup_{i=1}^{n} C_i$ as a conflict–free extension. A set $E \subseteq A$ is an admissible extension of FN iff F^{FN} admits a set $E' = \{C_1, ..., C_n\} \subseteq A'$ s.t. $E = \bigcup_{i=1}^{n} C_i$ as an admissible extension. For every complete extension of FN there exists exactly one corresponding complete extension of F^{FN}.*

Proof. Let $E \subseteq A$ be a conflict–free extension of FN. It does not have to be the case that every argument in E possesses a powerful sequence on A. Consequently, E might not be a subset of $\bigcup A'$, and thus it might not have a corresponding conflict–free extension in F^{FN}.

Let $E \subseteq A$ be a strongly coherent extension of FN. By definition, E is a coherent set for any of the arguments it contains. Thus, for every argument $a \in A$, there exists at least one $E_a \subseteq E$ that is a minimal coherent set for a. Let $E' = \{E_a\}_{a \in E}$ be a collection of such sets. We can observe that the union of all sets in E' is E. Assume that E' is not conflict–free. By the construction of F^{FN}, it means there are some sets $E_a, E_b \in E'$ s.t. $\exists x_1 \in E_a, x_2 \in E_b$ s.t. $x_1 R x_2$. However, as both $x_1, x_2 \in E$, this breaches the conflict–freeness of E. We reach a contradiction and can conclude that if E is strongly coherent, then E' is conflict–free.

Let $E' = \{E_1, ..., E_n\}$ be a conflict–free extension of F^{FN}. By construction, it means that every E_i is coherent, and that for no $a \in E_i, b \in E_j$, aRb. Therefore, $E = \bigcup_{i=1}^{n} E_i$ is also coherent and conflict–free. Consequently, E is strongly coherent in FN.

Let $E \subseteq A$ be an admissible extension of FN and $E' = \{E_a\}_{a \in E}$ the corresponding conflict–free set described in the previous part of this proof. Assume it is not admissible, i.e. there exists $E_x \in E'$ and $E_y \in A'$ s.t. $E_y R' E_x$ and no $E_z \in E'$ s.t. $E_z R' E_y$. By the construction of F^{FN}, this means that there is an argument $a \in E$ (occurring in E_x) that is attacked by $b \in A$ (occurring in E_y) and no $c \in E$ (as there is no $E_z \in E'$) that is attacking the coherent set for b represented by E_y. Therefore, E could not have defended a, and we reach a contradiction with the admissibility of E. Thus, if E is admissible in FN, then so is E' in F^{FN}.

Let $E' = \{E_1, ..., E_n\}$ be an admissible extension of F^{FN}. By the previous analysis, $E = \bigcup_{i=1}^{n} E_i$ is strongly coherent in F^{FN}. If it is not admissible, then there is an argument $a \in E$ s.t. $\exists b \in A$, bRa and there is a coherent set $B \subseteq A$ for b s.t. there is no $c \in E, d \in B$ for which cRd. Without the loss of generality, we can assume that B is a minimal coherent set for b. By the construction of F^{FN}, every $E_i \in E'$ s.t. $a \in E_i$ is attacked by B in R'. Moreover, it also holds there is no $E_j \in E'$ s.t. $E_j R' B$. Consequently, E' could not have defended the AF arguments in E' representing coherent sets for a. We reach a contradiction with the admissibility of E'. Therefore, if E' is admissible in F^{FN}, then so is E in FN.

Let $E' = \{E_1, ..., E_n\}$ and $F = \{F_1, ..., F_k\}$ be two complete extension of F^{FN} s.t. $E = \bigcup_{i=1}^{n} E_i = \bigcup_{i=1}^{k} F_i$ and $E' \neq F$. Let us assume there is an AF argument G s.t. $G \in E'$ but $G \notin F$. This means that E' does not defend G, even though F does. However, as $\bigcup_{i=1}^{n} E_i = \bigcup_{i=1}^{k} F_i$, then for every AFN argument $a \in G$, there is another AF argument in F containing it. Therefore, from the way attacks are created in F^{FN}, it can be shown that as F defends all of its arguments, it has to defend G as well. Thus, by completeness it has to be the case that $G \in F$, and we reach a contradiction with the assumptions. We can repeat the same analysis for $G \in F$ and $G \notin E'$ and come to the same conclusion. Therefore, it has to be the case that $E' = F$. Hence, there can only be one complete extension associated with a given AFN extension. □

Theorem 10.4. *Let $FN = (A, R, N)$ be an AFN and $E, E' \subseteq A$ two admissible extensions of FN. If for every $a \in E$, $b \in E'$ it is not the case that $(a, b) \in R$ and $(b, a) \in R$, then $E \cup E'$ is also admissible.*

Proof. First of all, it can be easily show that since both E and E' are coherent, then so is $E \cup E'$. Moreover, as the sets are conflict–free and there is no conflict between E and E' either, then $E \cup E'$ is also conflict–free. Thus, the union of our sets is strongly coherent.

We now need to focus on defense. Assume it is not the case, and there is an argument $a \in E \cup E'$ that is attacked by $b \in A$ and not all coherent sets of b are attacked by $E \cup E'$. The argument a has to appear in at least one of those sets. Clearly, if $E \cup E'$ does not attack all coherent sets of b, then neither does E nor E'. Therefore, at least one of them cannot be admissible, and we reach a contradiction with the conclusions. Hence, we can conclude that $E \cup E'$ is an admissible extension of FN. □

Theorem 10.5. *Let $FN = (A, R, N)$ be an AFN. For any two admissible extensions E_1 and E_2 of FN, if for every argument $a \in E_1, b \in E_2$ there exists an admissible extension E_3 of FN s.t. $a, b \in E_3$, then $E_1 \cup E_2$ is an admissible extension of FN.*

Proof. Assume it is not the case, i.e. even though the conditions are met, $E_1 \cup E_2$ is not an admissible extension of FN. If $E_1 \cup E_2$ is not admissible, even though both E_1 and E_2 are, then by Theorem 10.4 there exist arguments $a \in E_1$, $b \in E_2$ s.t. either $(a, b) \in R$ or $(b, a) \in R$. However, this means that E_3 cannot be conflict–free, let alone admissible. Thus, we reach a contradiction. □

Theorem 10.9. *Let $FN = (A, R, N)$ be an AFN and $SF^{FN} = (A', R'')$ its corresponding attack propagated SETAF obtained by Translation 63. If $E \subseteq A$ is strongly coherent in FN, then it is conflict–free in SF^{FN}. It does not necessarily hold for conflict–free semantics. If E is a σ–extension of FN, where $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$, then it is a σ–extension of SF^{FN}. If $E' \subseteq A'$ is a σ'–extensions of SF^{FN}, where $\sigma' \in \{$conflict–free, preferred, complete, grounded, stable$\}$, then it is also a σ'–extension of FN. It does not necessarily hold for admissible semantics. If E' is conflict–free in SF^{FN}, then it is not necessarily strongly coherent in FN.*

Proof. With the exception of conflict–free semantics, the extensions of FN and its weak validity form $FN^{wv} = (A', R', N')$ coincide by Theorem 4.18. Therefore, what we need to show is that the extensions of $FN^{wv} = (A', R', N')$ and $SF^{FN} = (A', R'')$ coincide as well.

We can observe that not every conflict–free extension of FN is conflict–free in FN^{wv}, even though all conflict–free extensions of FN^{wv} are conflict–free in FN (see Theorem 4.18). Let $E \subseteq A'$ be a conflict–free set of FN^{wv}. E is not necessarily conflict–free in SF^{FN}. Let us consider a trivial framework $(\{a, b, c\}, \{(a, b)\}, \{(\{b\}, c)\})$ where a attacks b and b supports c. The corresponding SETAF is $(\{a, b, c\}, \{(\{a\}, b), (\{a\}, c)\})$. We can observe that while $\{a, c\}$ is AFN conflict–free, it is not SETAF conflict–free.

Let $E \subseteq A'$ be a conflict–free extension of SF^{FN}. Since R'' contains the attacks from R', it is easy to see that E is also conflict–free in FN^{wv}. Please note it is not necessarily strongly coherent in FN^{wv} – the same explanations that will be provided in the case of admissible semantics apply. Therefore, we can conclude that not every conflict–free extension of FN is conflict–free in SF^{FN}, even though every conflict–free set of SF^{FN} will be conflict–free in FN.

Let $E \subseteq A'$ be a strongly coherent set of FN^{wv}. For every $a \in E$, E will be in $a's$ collection of coherent sets. Since E is conflict–free, none of the arguments in E will be in the set Z containing all and only attackers of E. Consequently, no nonempty subset of E will be a propagated attack on any of the elements of E, as it will not have a shared element with Z. We can thus conclude that E is conflict–free in SF^{FN}.

Let $E \subseteq A'$ be an admissible extension of FN^{wv}. We know it is conflict–free in SF^{FN}. Assume it is not admissible in SF^{FN}; this means there exists an argument $a \in E$ and a set of arguments $B \subseteq E'$ s.t. $(B, a) \in R''$, but no subset $E' \subseteq E$ and no argument $b \in B$ s.t. $(E', b) \in R''$. Therefore, by the construction of SF^{FN}, B can attack any coherent set of a, which includes E. Consequently, there is an argument $c \in B, d \in E$ s.t. $(c, d) \in R$. Unfortunately, E is not capable of attacking all coherent sets for c in R'. Therefore, we breach the admissibility of E in FN^{wv}. Thus, E is admissible in SF^{FN}.

Let $E \subseteq A'$ be an admissible extension of SF^{FN}. It is not necessarily admissible in FN^{wv}. Let $(\{a, b\}, \emptyset, \{(\{a\}, b)\})$ be a simple AFN containing only a supporting edge from a to b. The corresponding SETAF is $(\{a, b\}, \emptyset)$. We can observe that $\{b\}$ is admissible in our SETAF, even though it is not even coherent in AFN.

Let $E \subseteq A'$ be a complete extension in FN^{wv}. By the previous parts of this proof it follows that E is an admissible extension of SF^{FN}. Assume it is not complete; this means there exists an argument $b \in A'$ that is defended by E in SF^{FN}, but not in FN^{wv}. Let $(a_0, ..., a_n)$ be an arbitrary sequence for b. Since b is coherent in A', at least one such sequence will exist. Assume that a_0 is not in E; it must be thus the case that it is not defended by E. Since $E \cup \{a_0\}$ is trivially coherent, we can conclude that E does not defend a_0 from attacks. Therefore, there exists an argument $y \in A'$ s.t. $yR'a_0$ and it is not the case that E attacks all coherent sets containing y. Consequently, a_0 will be attacked by y_0 in R'', but there will be no propagated attack from any subset of E to y_0. Therefore, E cannot defend a_0 in SF^{FN}. If it is the case that a_0 is in E, then let us focus on a_1. Again, assume that a_1 is not in E. Due to the presence of a_0, $E \cup \{a_1\}$ is trivially coherent. Thus, it must be the case that a_1 is not defended from attacks by E and again by the analysis above we can find at least one attacker $y_1 \in A'$ not covered by E. We can continue this analysis until we reach $a_n = b$ and conclude that there must have been at least one argument y_i attacking an element in the sequence that has a coherent set not attacked by E. We can repeat this reasoning for any powerful sequence for b, each time coming to a conclusion that there must have been an attacker not covered by E. From these attackers we can build a propagated attack on b, and since none of its elements is attacked in R'' by any subset of E, it could not have been the case that E defended b in SF^{FN}. We reach a contradiction. Consequently, E is complete in SF^{FN}.

Let $E \subseteq A$ be a complete extension of SF^{FN}. We know it is conflict–free in FN^{wv}; we now need to show it is coherent, admissible and complete. Let $\{X_1^a, ..., X_n^a\}$ be the collection of all coherent sets on A' s.t. $a \in X_i^a$ and $Z_i^a = \{b \in A' \mid \exists c \in X_i^A, (b, c) \in R'\}$ be the set of all arguments attacking a given X_i^a in R'. Let $Z'_1, ..., Z'_m$ be the sets of arguments attacking a through R'' constructed as in Translation 63.

First of all, we will show that if an argument $a \in A'$ is defended by E in SF^{FN}, then so are the members of at least one coherent set containing a. Assume that it is not the case and that for every coherent set X_i^a for a, there exists a set Z_j' attacking a member of X_i^a in R'' s.t. there is no $E' \subseteq E, z \in Z_j'$ for which $(E', z) \in R''$. We can observe that $\bigcup_{i=1}^{n} X^a$ is a coherent set for a. Therefore, based on our translation, the union of such Z_j' sets forms an attacker set for a in R''. Since none of the Z_j' sets was attacked by E on any of its elements, it follows that E does attack any member of their union. This means that E cannot possibly defend a and we reach a contradiction our assumptions. Thus, if E defends a, then it also defends all members of at least one coherent set for a. Moreover, by completeness, E has to contain all of them. Hence, we can conclude that if E is complete in SF^{FN}, then it is coherent in FN^{wv}.

Let us now focus on admissibility. Assume that even though every argument in E is defended by E in SF^{FN}, there is an argument $a \in E$ not defended in FN^{wv}. Since the set is coherent, it can thus only be the case that there exists a coherent set $C \subseteq A'$ not attacked by E and containing an argument b s.t. $bR'a$. Since $bR'a$, then by Translation 63 $\{b\}R''a$ in SF^{FN}. If there is a coherent set for b not attacked by E, then it cannot be the case there exists a propagated attack from any subset of E to b in R'' and thus a could not have been defended in SF^{FN}. We reach a contradiction and can thus conclude that E is an admissible extension of FN^{wv}.

Finally, we can show that E is complete in FN^{wv}. If it is not complete, then it means there is an argument $a \notin E$ which is defended by E in FN^{wv}, but not in SF^{FN}. If it is defended by E in FN^{wv}, then every coherent set C for an argument b s.t. $bR'a$ contains an argument attacked by E. Moreover, it also holds that $E \cup \{a\}$ is a coherent set for a. Let $Z' \subseteq A'$ be a set of arguments attacking a in R''. By the construction of SF^{FN}, we can observe that at least one of those sets is $\{b\}$ and that E contains a subset attacking b in R''. Therefore, E defends a against any attacks carried out by sets containing b in SF^{FN}. Let us now consider other options; by construction, every Z' contains those arguments s.t. for every coherent C set for a, there is $c \in C$, $z \in Z'$ s.t. $(z, c) \in R'$. Since $E \cup \{a\}$ is a coherent set for a and we have excluded all the direct attackers of a in R', then there is an argument $e \in E$, $z \in Z'$ s.t. $(z, e) \in R'$ and thus $(\{z\}, e) \in R''$. As E is admissible, then there has to be a subset $E' \subseteq E$ s.t. $(E', z) \in R''$. Consequently, it can be shown that E defends a in SF^{FN}, and we reach a contradiction with the assumptions. Thus, E is complete in FN^{wv}.

Since we know that complete extensions coincide between FN^{wv} and SF^{FN}, by Theorems 2.24 and 2.95 we can conclude that the preferred and grounded coincide as well between the two frameworks. What remains to show is the relation between the stable extensions. Let $E \subseteq A'$ be a stable extension in FN^{wv}. By Lemma 2.94 we know it is strongly coherent and attacks every coherent set of any argument $a \notin E$. By the analysis above it is easy to see that the set will be conflict–free in SF^{FN} and for every such a there will exist an according propagated attack. Consequently, E will be stable in SF^{FN}.

Let E be a stable extension of SF^{FN}. Using Theorems 2.24 and 2.23 we know it is complete and thus strongly coherent in FN^{wv}. Since every argument $a \notin E$ is attacked in R'', then by construction of R'' we know that every coherent set containing a is attacked by E in FN^{wv}. Consequently, it will be in E^{att} and using Lemma 2.94 we can conclude E is stable in FN^{wv}. □

Theorem 10.11. *Let $FN = (A, R, N)$ be an AFN and $SF^{FN} = (A', R')$ its corresponding defender SETAF obtained by Translation 65. By $E_{np} = \{a' \mid$ there is no coherent set containing $a\} \cup \{a' \mid$ for every coherent set C for a, $\exists e \in E, c \in C \setminus \{a\}, (e, c) \in R\}$ we will denote primed arguments corresponding to a subset of E^{att}, in which every argument a either has no coherent set or every of its coherent sets is attacked by E on an argument different from a.*

If a set $E \subseteq A$ is conflict–free in FN, then it is conflict–free in SF^{FN}. The set $E \cup E_{np}$ is not necessarily conflict–free in SF^{FN}. If a set E is strongly coherent in FN, then $E \cup E_{np}$ is conflict–free in SF^{FN}. If E is a σ–extension of FN, where $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$, then $E' = E \cup E_{np}$ is a σ–extension of SF^{FN}.

If a set $E' \subseteq A'$ is a σ–extension of SF^{FN}, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, then $E = E' \cap A$ is a σ–extension of FN. If E' is conflict–free, $E = E' \cap A$ does not have to be strongly coherent in FN.

Proof. Let $E \subseteq A$ be a conflict–free extension of FN. Since all attacks in R' that occur between the arguments in A only correspond precisely to R, then E is easily conflict–free in SF^{FN}. However, $E' = E \cup E_{np}$ is not necessarily conflict–free in SF^{FN}. Take for example the framework $FN = (\{a, b, c\}, \{(a, b)\}, \{(\{b\}, c)\})$. The set $\{a, c\}$ is conflict–free in FN, despite the fact that a attacks the only supporter b of c. Thus, $\{a, c\}_{np} = \{c'\}$, and we obtain the set $\{a, c, c'\}$ on the SETAF side. Clearly, a primed argument cannot appear in a conflict–free extension along with its original version.

Let $E \subseteq A$ be a strongly coherent set of FN. Let us assume that $E' = E \cup E_{np}$ is not conflict–free in SF^{FN}, i.e. there exist a set of arguments $S \subseteq E'$ and an argument $b \in E'$ s.t. $SR'b$. By the construction of SF^{FN}, we can observe it cannot be the case that $b \in A$ and $S \subseteq A$. Therefore, either S or b is in fact (a set consisting of) a primed argument. Let us assume that $b \in A$; consequently, $S = \{b'\}$. By the construction of E', this means that every powerful sequence for b is attacked by E on a non–b element. However, as E contains the elements of at least one such sequence for b, we breach the conflict–freeness of E in FN. Let us thus assume that $S \subseteq E$ and $b = c' \in E_{np}$ is a primed version of an argument $c \in A$. By the construction of SF^{FN} this means that $\{c\} \cup S$ is a coherent set for c. However, due to the conflict–freeness of E, it cannot be the case that E attacks any member of S. This means that c' (and thus b) should not have appeared in E_{np} in the first place and we reach a contradiction with the construction of E'. Therefore, E' is conflict–free in SF^{FN}.

Let $E' \subseteq A'$ be a conflict–free extension of SF^{FN}. Since every attack from R is contained in R', it follows easily that $E = E' \cap A$ is a conflict–free set of FN. Nevertheless, it does not have to be coherent, and therefore strongly coherent. In the example given in the first paragraph on the proof, a set $\{c\}$ would be considered conflict–free in the target SETAF. However, due to the absence of b, it is not coherent in the source AFN.

Let $E \subseteq A$ be an admissible extension of FN. By definition, it is strongly coherent. Thus, by the previous part of this proof, we know that $E' = E \cup E_{np}$ is conflict–free in SF^{FN}. What remains to be shown is that it defends all of its members. Let a be an argument in E. If it requires no support to stand in FN, then there is no auxiliary argument attacking it in R'. If it does, then from the coherence of E in FN, it follows easily that $\exists X \subseteq E$ s.t. $XR'a'$ in SF^{FN}. Thus, we can conclude that E' defends its elements from E from attacks by primed arguments. From the way E_{np} is constructed and the fact that every attack from R is represented in R', it is also easy to see that the elements of E_{np} are also defended by E' in SF^{FN}. Therefore, what is left is to prove that a is defended by E' from attacks carried out by arguments in E. Since a is defended by E in FN, then for every b s.t. bRa, every coherent set C containing b is attacked by E. If b is not coherent in A, then it is attacked by an initial argument b' in R', which by construction of E_{np} is in E'. Thus, let us focus on the case that b is coherent in A. If all of its coherent sets are attacked through b itself, i.e. $\exists e \in E$ s.t. eRb, then obviously $e \in E'$ and $\{e\}R'b$ and a is defended. If all of coherent sets of b are attacked through a non–b element, then by construction E_{np} contains b' and $\{b'\}R'b$. Again, a is defended by E'. We can thus conclude that E' is admissible in SF^{FN}.

Let $E' \subseteq A'$ be an admissible extension of SF^{FN}. We know that $E = E' \cap A$ is at least conflict–free in FN. Let us assume that E is not coherent in FN, i.e. there exists $a \in E$ that does not have a powerful sequence on E. If a does not have a powerful sequence on A to start with, then by Translation 65 it would be attacked by $\{a'\}$ in R' and the same time there would be no set of arguments attacking a'. Consequently, E' could have been admissible in SF^{FN}. Therefore, there exists at least one minimal powerful sequence for a. If a did not require support through N, then it is trivially coherent in E. Otherwise, we know it is attacked by $\{a'\}$ in A' and that every set of arguments attacking a' corresponds to the members of a powerful sequence for a that precede a. Thus, if there is no powerful sequence on E for a, then a could not have been defended by E' against $\{a'\}$ in SF^{FN}. This leads us to a conclusion that E is coherent in FN.

Let us now assume that E is not admissible in FN. As it is coherent, it means that there exist arguments $a \in E, b \in A$ s.t. bRa and there is a powerful sequence $(b_0, .., b_n)$ for b which is not attacked by E. Since b_0 requires no support through N, there is no argument b_0' attacking b_0 in SF^{FN}. Thus, if no argument in E attacks b_0, then no subset of E' attacks b_0 either. Let us focus on b_1; if it requires no support through N, we can repeat the b_0 analysis. If it does, then we know there exists an argument b_1' in A' attacking b_1. As $\{b_0\}R'b_1'$ and there is no argument in E (and no set in E') attacking b_0, we can conclude that b_1' could not have been in E' due to admissibility of E' in SF^{FN}. Therefore, E' does not attack b_1. We can repeat this analysis till we reach $b_n = b$ and conclude that there is no subset of E' attacking b. Consequently, a could not have been defended by E' in the first place and we reach a contradiction with the admissibility of the set in SF^{FN}. Thus, E is admissible in FN.

Let $E \subseteq A$ be complete in FN. By the previous parts of this proof, we know that $E' = E \cup E_{np}$ is admissible in SF^{FN}. If it is not complete, it means there exists an argument $a \in A' \setminus E'$ defended by E' in SF^{FN}. Assume that $a = b'$ is a primed argument for $b \in A$. Since b' is not in E', then $b' \notin E_{np}$. Thus, E does not attack a non–b member in every powerful sequence of b. If b' is not attacked at all in R', then by the construction of E' it has to be in E' already. Hence, there is at least one powerful sequence $(b_0, ..., b_n)$ for b s.t. $\{b_0, ..., b_{n-1}\}$ attacks b' and no argument in E attacks any of the elements in $\{b_0, ..., b_{n-1}\}$ in FN. However, since we claim E' defends b' in SF^{FN}, there must be a subset of E' that attacks some b_i for $0 \leq i \leq n-1$.

Let us start with b_0. By the requirements of the powerful sequences, b_0 possesses no auxiliary argument. Thus, if it is not attacked by a member of E in R, then it cannot be the case that there is a subset of E' attacking b_0 either. Let us thus focus on b_1. If it has no primed attacker, we can repeat the previous analysis. Thus, assume it is attacked by an argument b'_1. Again, by the construction of the sequence, b'_1 is attacked by $\{b_0\}$ in R'. Since we have established that E' does not attack b_0, it cannot be the case that it defends (and thus contains) b'_1. Thus, if b_1 is not attacked by E, it is not attacked by E' either.

We can continue this line of reasoning until we reach the conclusion that E' could not have defended b' without E attacking a member of every powerful sequence for b. Therefore, if E' defends a primed argument, then this argument is already in E'.

Let us thus assume that a is an argument in A. Therefore, a is defended by E' in SF^{FN}, but not by E in FN. If E does not defend a, then either $E \cup \{a\}$ is not coherent or there exists $b \in A$ and a coherent set $C \subseteq A$ containing b s.t. bRa and E does not attack any element in C. Let us focus on the first case. If a requires no support, then $E \cup \{a\}$ is trivially coherent. Thus, it has to be the case there is a set of arguments supporting it through N, and hence there exists a primed argument $a' \in A'$. If $E \cup \{a\}$ is not coherent, then E cannot contain the members of any powerful sequence for a preceding a, and thus E' cannot defend a against a'. We reach a contradiction. Let us focus on the other case. If bRa, then $\{b\}R'a$ and thus E' has to be able to attack b in SF^{FN}. However, we can reuse the previous analysis to show that if no member of C is attacked by E, then b cannot be attacked by E', which contradicts our assumptions. Thus, we can finally conclude that E' is complete in SF^{FN}.

Let $E' \subseteq A'$ be a complete extension of SF^{FN}. We know that $E = E' \cap A$ is at least admissible in FN. What remains to be shown is that there is no argument $a \in A \setminus E$ defended by E in FN. Assume it is not the case and that E defends a, even though E' does not. First of all, if E defends a in FN, then $E \cup \{a\}$ is coherent. If a requires no support through N in the first place, then it does not posses a primed argument attacking it in SF^{FN}. If it does require support, then there is a powerful sequence for a s.t. elements of the sequence preceding it are contained in E, and thus in E'. Consequently, E' attacks the auxiliary argument a' for a. Thus, if E' does not defend a against some attack, this attack can only come from an argument in A and thus from R. Let us thus assume that there is an argument $b \in A$ s.t. $\{b\}R'a$, but E' does not attack b. We know that E attacks every powerful sequence of b in FN. If at least one of these attacks are carried out towards b itself, then $\exists c \in E$ s.t. cRb and thus $\{c\}R'b$. Hence, E must be defending b. Let us thus assume that none of the attacks is on b. However, since every attack in R is in R', then obviously E' defends the auxiliary argument b' of b. Consequently, $b' \in E'$ and E' attacks b, thus again defending a. We reach a contradiction. We can finally conclude that E is complete in FN.

Let us now show that there is a one–to–one relation between the complete extensions of FN and SF^{FN}. Assume it is not the case. Thus, there exist two different complete extensions E' and E'' of SF^{FN} s.t. $E' \cap A = E'' \cap A$. This means they can only differ by primed arguments. Assume there is an argument $a' \in E'$, but not in E''. By construction, a' can only be attacked by subsets of A representing powerful sequences for a. Thus, there exists a set $B \subseteq A$ s.t. $BR'a'$ and E' attacks an argument in B, but E'' does not. Let $B = \{a_0, ..., a_{n-1}\}$ be a part of a powerful sequence for a without a. Due to the properties of a_0, there is no primed argument for a_0. Thus, if E' attacks a_0, then so does E'', and we reach a contradiction. Therefore, E' does not attack a_0, and cannot defend a'_1. Hence, $a'_1 \notin E'$. Consequently, if E' attacks a_1, it can only be through arguments from A, and in this case E'' would attack a_1 as well. Again, we reach a contradiction. We can continue in this manner until we reach a_{n_1} and the conclusion that a' could not have been in E'. Therefore, $E' \subseteq E''$. In a similar manner we can show that $E'' \subseteq E'$. Hence, there is a one–to–one relation between the complete extensions of FN and SF^{FN}.

By using the results above and Theorems 2.24 and 2.95, we can easily prove the relation between preferred and grounded extensions of FN and SF^{FN} stated in the theorem.

Let us focus on the stable semantics. Assume that E is stable in FN. Then $E' = E \cup E_{np}$ is at least complete, and thus conflict–free in SF^{FN}. From the fact that every argument $e \in A \setminus E$ is in E^{att}, we can easily prove that e is attacked by E' as well. What remains to be shown is that every auxiliary argument not in E' is also attacked by E' in SF^{FN}. Let a' be an arbitrary auxiliary argument outside of the extension. If a is in E', then by the fact that E' is complete it has to be the case that E' attacks a'. If a is not in E', then by the correspondence with E it means that all of its coherent sets were attacked. Should all members of its powerful sequences be attacked on a non–a member, then a' must be in E'. Thus, we are left with a case were all the sequences are attacked on a only. However, by the completeness of E in FN this means that E must have included the members of these sequences (excluding a). Thus, E' had sufficient means to attack a' again. Consequently, E' is stable in SF^{FN}.

Let us now focus on the other way around and show that if $E' \subseteq A'$ is stable in SF^{FN}, then so is $E = E' \cap A$ in FN. By Theorems 2.23 and 2.24 we know that E' is complete in SF^{FN}. Consequently, E is complete in FN. Due to the one–to–one relation between the complete extensions, we can observe that E' must be of the form $E \cup E_{np}$. Let us now assume that there is an argument $a \in A \setminus E$ which is not in E^{att}. This means that not every coherent set of this argument is attacked by E. Thus, it cannot be the case that E' contains a' – it just wouldn't be able to defend it. Hence, E' is not be able to attack a in SF^{FN} (all the other attacks are after all the same as in FN) and it could not have been stable in the first place. We reach a contradiction. Therefore, E is stable in FN. □

Theorem 10.12. *Let $FN = (A, R, N)$ be a strongly valid AFN and $SF^{FN} = (A', R')$ its corresponding defender SETAF obtained by Translation 66. By $E_{np} = \{a' \mid$ for every* *coherent set C for a, $\exists e \in E, c \in C \setminus \{a\}, (e, c) \in R\}$ we will denote primed arguments corresponding to a subset of E^{att} in which for every argument a and any coherent set for it, there is a member of this set attacked by E different from a.*

If a set $E \subseteq A$ is conflict–free in FN, it is conflict–free in SF^{FN}. The set $E \cup E_{np}$ is not necessarily conflict–free in SF^{FN}. If a set E is strongly coherent in FN, then $E \cup E_{np}$ is conflict–free in SF^{FN}. If E is a σ–extension of FN, where $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$, then $E' = E \cup E_{np}$ is a σ–extension of SF^{FN}.

If a set $E' \subseteq A'$ is a σ–extension of SF^{FN}, where $\sigma \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, then $E = E' \cap A$ is a σ–extension of FN. If E' is conflict–free, the set $E = E' \cap A$ does not have to be strongly coherent in FN.

Proof. The analysis of conflict–freeness is the same as in Theorem 10.11.

Let $E \subseteq A$ be a strongly coherent set of FN. Let us assume that $E' = E \cup E_{np}$ is not conflict–free in SF^{FN}, i.e. there exist a set $S \subseteq E'$ and argument $b \in E'$ s.t. $SR'b$. By the construction of SF^{FN}, we can observe it can only be the case that either S or b is in fact (a set consisting of) a primed argument. Let us assume that b is an argument in A and $S \subseteq A'$. Since $SR'b$, it has to be the case that $S = \{b'\}$. However, we can observe that E is a coherent set for b, and by conflict–freeness of E it cannot be the case that E attacks any of its members. Therefore, we reach a contradiction with the construction of E_{np} and it cannot be the case that $b' \in E'$. Let us therefore assume that $S \subseteq A$ and $b = c'$ for an argument $c \in A$. It can be easily shown that $E \cup \{c\}$ is a coherent set for c. Hence again due to conflict–freeness of E, we can observe that c' should not have appeared in E'. Again, we reach a contradiction, and can conclude that E' is conflict–free in SF^{FN}.

Let $E' \subseteq A'$ be a conflict–free extension of SF^{FN}. Since every attack from R is contained in R', it follows easily that $E = E' \cap A$ is a conflict–free set of FN. Nevertheless, it does not have to be coherent, and therefore strongly coherent. We can consider a simple AFN $(\{a, b\}, \emptyset, \{(\{a\}, b)\})$. The corresponding SETAF is $(\{a, b, b'\}, \{(\{b'\}, b), (\{a\}, b')\})$. Although the set $\{b\}$ is SETAF conflict–free, it is not AFN coherent.

Let $E \subseteq A$ be an admissible extension of FN and $E' = E \cup E_{np}$ its corresponding set in SF^{FN}. By the previous parts of this proof, E' is conflict–free in SF^{FN}. What remains to be shown is that E' defends every $a \in E'$. Let us first assume that $a \in E$, i.e. a is a standard argument. Since E is admissible in FN, then it attacks every coherent set of an argument $b \in A$ s.t. bRa. We can also observe that due to strong validity, every attacker in R will possess at least one coherent set. Therefore, either there exists $c \in E$ s.t. cRb (and therefore $\{c\}R'b$), or the attacks on coherent sets are not carried out directly against b and thus $b' \in E_{np}$. Hence, in any case there is $X \subseteq E'$ s.t. $XR'b$ when $a, b \in A$ are normal arguments and $\{b\}R'a$. We can also observe that as E is a coherent set for a, then there exists a subset of E supporting a through N. This means we can construct a subset of E attacking a' in SF^{FN} (assuming a' is constructed at all). Therefore, if E is admissible in FN, then E' defends all of its arguments that are from E itself. What remains to be analyzed is the acceptability of arguments from E_{np}. Let $a = b'$ be a primed argument for $b \in A$; assume it is not defended by E'. Consequently, there exists a set of arguments

$X \subseteq A$ s.t. $XR'b'$ and the same time there is no $x \in X'$ and no set of arguments $X' \subseteq E'$ s.t. $X'R'x$. By the construction of SF^{FN}, such an X' would either correspond to an attacker from R or be the primed version of x (assuming it exists). Hence, we can show that it cannot be the case that E attacks every coherent set for x in FN. As it holds for every $x \in X$, we can collect these unattacked coherent sets into one single set $C \subseteq A$. By using Theorem 4.32, it can be observed that $C \cup \{b\}$ would be a coherent set for b and no argument in C is attacked by E in FN. Hence, we reach a contradiction with the construction of E' and b' should not have been in the set in the first place. We can therefore conclude that E' is an admissible extension of SF^{FN}.

Let $E' \subseteq A'$ be an admissible extension of SF^{FN} and $E = E' \cap A$ its corresponding set in FN. By the previous parts of this proof we know that E is conflict–free in FN. Let us now show it is also coherent. Let $a \in E' \cap A$ be a standard argument. If there is no $a' \in A'$, then (a) is a trivial powerful sequence for a and thus the argument is coherent in E in FN. If there exists $a' \in A'$, then due to the admissibility of E' in SF^{FN}, there will exist a set $X \subseteq E' \cap A$ of standard arguments s.t. $XR'a'$. Therefore, by the construction of SF^{FN}, for every set $C \subseteq A$ s.t. CNa, $C \cap E \neq \emptyset$. The same analysis can be performed for every argument in E. Hence, by Theorem 4.32, E is coherent in FN and as it is also conflict–free, we obtain strong coherence.

Let us now assume that E is not admissible in FN, i.e. there exist arguments $a \in E, b \in A$ s.t. bRa and there is a powerful sequence $(b_0, .., b_n)$ for b which is not attacked by E. Since b_0 requires no support through N, there is no argument b'_0 attacking b_0 in SF^{FN}. Thus, if E does not attack b_0, then no subset of E' attacks b_0 either. Let us focus on b_1; if it requires no support through N, we can repeat the b_0 analysis. If it does, then we know there exists an argument b'_1 in A' attacking b_1. As $\{b_0\}R'b'_1$ and there is no argument in E (and no subset of E') attacking b_0, we can conclude that b'_1 could not have been in E' due to admissibility of E' in SF^{FN}. Therefore, E' does not attack b_1. We can repeat this analysis till we reach $b_n = b$ and conclude that there is no subset of E' attacking b. Consequently, a could not have been defended by E' in the first place and we reach a contradiction with the admissibility of the set in SF^{FN}. Thus, E is admissible in FN.

Let $E \subseteq A$ be a complete extension of FN and $E' = E \cup E_{np}$ its associated admissible extension of SF^{FN}. We will show it is also complete. Assume it is not the case; this means there exists an argument $a \in A' \setminus E'$ defended by E' in SF^{FN}.

We can first consider the case that $a = b'$ is a primed argument for $b \in A$. Since $b' \notin E'$, from the construction of E' it follows that we can find a powerful sequence $(b_0, ..., b_n)$ for b that is not attacked on a non–b member by E. Let us start with b_0. By the requirements of the powerful sequences, b_0 possesses no auxiliary argument. Thus, if it is not attacked by a member of E in R, then it cannot be the case that there is a subset of E' attacking b_0 either. Let us thus focus on b_1. If it has no primed attacker, we can repeat the previous analysis. Thus, assume it is attacked by an argument b'_1. Again, by the construction of the sequence, b'_1 is attacked by $\{b_0\}$ in R'. Since we have established that E' does not attack b_0, it cannot be the case that it defends (and thus contains) b'_1. Thus, if b_1 is not attacked by E, it is not attacked by E' either. We can continue this line of reasoning until we reach b_{n-1} and the conclusion that E' could not have defended b' if there exists a sequence for b that is not attacked by E on a non–b member. Therefore, if E' defends a primed argument, then this argument is already in E'.

We can now consider the case where $a \in A$ is a normal argument. Therefore, a is defended by E' in SF^{FN}, but not by E in FN. If E does not defend a, then either $E \cup \{a\}$ is not coherent, or there exists $b \in A$ and a coherent set $C \subseteq A$ containing b s.t. bRa and E does not attack any element in C.

Let us focus on the missing support case. If a requires no support, then $E \cup \{a\}$ is trivially coherent, and we reach a contradiction. If there is a set of arguments supporting a through N, then there exists a primed argument $a' \in A'$. If $E \cup \{a\}$ is not coherent, then E cannot contain the members of any powerful sequence for a preceding a. Thus, we can use Theorem 4.32 to show that there exists a set of arguments F supporting a in N s.t. $F \cap E = \emptyset$. Consequently, by the construction of SF^{FN}, no subset of E' attacks a'. Hence, a is not defended by E' and we reach a contradiction.

Let us focus on the unattacked attacker case. If bRa, then $\{b\}R'a$ and thus E' has to be able to attack b in SF^{FN}. If not every powerful sequence for b is attacked by C, then not every powerful sequence for b is attacked a non–b element, and thus $b' \notin E'$ (if it exists in the first place). If no $c \in E$ attacks b in R, then no $\{c\}$ attacks b in R' either. In conclusion, no subset of E' can attack b, and we reach a contradiction. Thus, we can finally conclude that E' is complete in SF^{FN}.

Let $E' \subseteq A'$ be a complete extension of SF^{FN}. By the previous parts of this proof we know that $E = E' \cap A$ is at least admissible in FN. What remains to be shown is that there is no argument $a \in A \setminus E$ defended by E in FN. Assume it is not the case and that E defends a, even though E' does not.

We can observe that if E defends a in FN, then $E \cup \{a\}$ is coherent. If a requires no support through N in the first place, then it does not posses a primed argument attacking it in SF^{FN}. If it does require support, then for every $F \subseteq A$ s.t. FNa, $F \cap E \neq \emptyset$. Thus, there is a subset of E (and thus E') capable of attacking the primed argument for a. Consequently, E' defends a against its primed argument, assuming it exists.

We can now see that if E' does not defend a against some attack, this attack can only come from R and be carried out by a standard argument. Let us thus assume that there is an argument $b \in A$ s.t. $\{b\}R'a$, but E' does not attack b. We know that E attacks every powerful sequence of b in FN. If at least one of these attacks are carried out towards b itself, then there exists $c \in E$ s.t. cRb and thus $\{c\}R'b$. Hence, E must be defending b. Let us thus assume that none of the attacks is on b. However, we can reuse the previous parts of this proof to show that E' defends the auxiliary argument b' of b. Consequently, $b' \in E'$ and E' attacks b, thus again defending a. We reach a contradiction. We can finally conclude that E is complete in FN.

Let us now show that there is a one–to–one relation between the complete extensions of FN and SF^{FN}. Assume it is not the case. Thus, there exist two different complete extensions E' and E'' of SF^{FN} s.t. $E' \cap A = E'' \cap A$. This means they can only differ by primed arguments. Assume there is an argument $a' \in E'$, but not in E''. By construction,

a' is attacked only by (minimal) subsets of A that share a common element with every set that supports a through N. Thus, there exists such a set $B \subseteq A$ s.t. $BR'a'$ and E' attacks an argument from B, but E'' does not. Due to the fact that we are dealing with strongly valid AFN, we can observe that every such attacking set B is a part of a set that would sufficiently support a, and by Theorem 4.32 is coherent. In other words, B is a part of a powerful sequence for a. We can observe that an attacker for B cannot be a standard argument (i.e. attack cannot originate from R), otherwise both sets E' and E'' would be capable of it. Hence, there is another primed argument corresponding to the argument in B attacked by E'. We can can find a set B' attacking this primed argument which in turn is attacked by E', but not by E''. By using previous observations, we can show that both B and B' are a part of a powerful sequence for a. We can continue this analysis until we reach a set of arguments in which no argument possesses a primed attacker, and as we are dealing with a strongly valid framework, this is bound to happen. Therefore, if E' attacks this argument, it is only by using attacks from standard arguments. However, if this is the case, then E'' attacks this argument as well, and we reach a contradiction. Therefore, $E' \subseteq E''$. In a similar manner we can show that $E'' \subseteq E'$. Hence, there is a one–to–one relation between the complete extensions of FN and SF^{FN}.

By using the results above and Theorems 2.24 and 2.95, we can easily prove the relation between preferred and grounded extensions of FN and SF^{FN} stated in the theorem.

Let us focus on the stable semantics. Assume that E is stable in FN. Then $E' = E \cup E_{np}$ is at least complete, and thus conflict–free in SF^{FN}. From the fact that every argument $e \in A \setminus E$ is in E^{att}, we can use previous parts of the proof to show that e is attacked by E' as well. What remains to be shown is that every auxiliary argument not in E' is also attacked by E' in SF^{FN}. Let a' be an arbitrary auxiliary argument outside of the extension. If the original argument a is in E', then by the fact that E' is complete (and thus defends a) it has to be the case that E' attacks a'. If a is not in E', then by the correspondence with E it means that all of its coherent sets were attacked. Should all members of its powerful sequences be attacked on a non–a member, then a' must be in E' and we reach a contradiction. Thus, we are left with the case where all the sequences are attacked on a only. However, by the completeness of E in FN this means that E must have included the members of these sequences (excluding a). By the construction of the sequences, we can show that E' had sufficient means to attack a' again. Consequently, E' is stable in SF^{FN}.

Let us now focus on the other way around and show that if $E' \subseteq A'$ is stable in SF^{FN}, then so is $E = E' \cap A$ in FN. By Theorems 2.23 and 2.24 we know that E' is complete in SF^{FN}. Consequently, E is complete in FN. Due to the one–to–one relation between the complete extensions, we can observe that E' must be of the form $E \cup E_{np}$. Let us now assume that there is an argument $a \in A \setminus E$ which is not in E^{att}. This means that not every coherent set of this argument is attacked by E. Thus, it cannot be the case that E' contains a' – it just wouldn't be able to defend it. Hence, E' is not be able to attack a in SF^{FN} (all the other attacks are after all the same as in FN) and it could not have been stable in the first place. We reach a contradiction. Therefore, E is stable in FN. □

Theorem 10.13. *Let $FN = (A, R, N)$ be a strongly valid and support binary AFN and $BF^{FN} = (A, R, S)$ its associated BAF obtained through Translation 67. Then BF^{FN} is support acyclic.*

Proof. Let $N' = \{(a, b) \mid (\{a\}, b) \in N\}$ be the binary version of N. Due to the fact that FN is support binary, we can observe that for every argument $a \in A$, the set $suf(a)$ from Definition 4.30 will consist of exactly one set of arguments equal to $sup(a)$. Thus, there exists only one support assignment function for FN from Definition 4.30. Since FN is strongly valid, A can be ordered into a powerful sequence w.r.t. this function. We can observe that this ordering will define a topological ordering on A w.r.t. N'. Thus, (A, N') has to be a directed acyclic graph, and therefore the support subgraph of BF^{FN} is directed acyclic. $\square$

Theorem 10.15. *Let $FN = (A, R, N)$ be an AFN and $ES^{FN} = (A, R, E)$ its associated EAS obtained through Translation 68. Then, ES^{FN} is attack binary. If FN is weakly (relation, strongly) valid, then so is ES^{FN}. If FN is strongly consistent, then so is ES^{FN}. ES^{FN} does not have to be in minimal form, even if FN is.*

Proof. The next theorem (Theorem 10.16) explains how powerful and evidential sequences of our frameworks are connected. By using this result it is easy to show that if FN is weakly valid, then so is ES^{FN}. By analyzing the sequences we can also show relation validity; if for every argument b appearing in a supporting set in N for an argument a we can find a powerful sequence in which it precedes a, then for every supporter in E for a we can find a suitable evidential sequence. Consequently, if FN is relation valid, then so is ES^{FN}. Strong validity can be shown simply by analyzing the definitions of the forms and Translation 68; the sup and suf sets we have used in establishing strong validity for AFNs are the same construction we have used in creating support sets in E in Translation 68 (the continuation merely removes some redundancies).

Let us now look at the strong consistency form. Let $N(a) = \{b \mid \exists C \subseteq A, b \in C \text{ s.t. } CNa\}$ and $R(a) = \{b \mid bRa\}$ be the arguments supporting and attacking a in FN. Let $E(a) = \{b \mid \exists C \subseteq A, b \in C \text{ s.t. } CEa\}$ and $R'(a) = \{b \mid \exists C \subseteq A, b \in C \text{ s.t. } CR'a\}$ be the arguments supporting and attacking a in FN. We can observe that $R(a) = R'(a)$ and that $\{\eta\} \cap R'(a) = \emptyset$. Moreover if $N(a) \neq \emptyset$, then $N(a) = E(a)$, and if $N(a) = \emptyset$, then $E(a) = \{\eta\}$. Consequently, we can show that if $R(a) \cap N(a) = \emptyset$, then $R'(a) \cap E(a) = \emptyset$. Hence, if FN is strongly consistent, then so is ES^{FN}.

Let us now consider a simple AFN $(\{a, b, c, d\}, \emptyset, \{(\{a, b\}, d), (\{a, c\}, d)\})$. We can observe it is in minimal normal form. The associated EAS is $(\{a, b, c, d, \eta\}, \emptyset, \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, c), (\{a\}, d), (\{a, c\}, d), (\{a, b\}, d), (\{b, c\}, d)\})$. We can observe it is not in minimal normal form anymore. $\square$

Theorem 10.16. *Let $FN = (A, R, N)$ be an AFN and $ES^{FN} = (A', R', E)$ its corresponding EAS obtained through Translation 68. Let $a \in A$ be an argument. If $(a_0, ..., a_n)$ is a powerful sequence for a on $S \subseteq A$ in FN, then $(\eta, a_0, ..., a_n)$ is an evidential sequence* *for a on $S \cup \{\eta\}$ in ES^{FN}. If $(\eta, a_0, ..., a_n)$ is an evidential sequence for a on $S \subseteq A'$ in ES^{FN}, then $(a_0, ..., a_n)$ is a powerful sequence for a on $S \setminus \{\eta\}$ in FN. If a set $S \subseteq A$ is coherent in FN, then $S \cup \{\eta\}$ is self–supporting in ES^{FN}. If $S' \subseteq A$ is self–supporting in ES^{FN}, then $S' \cap A$ is coherent in FN.*

Proof. Let $(a_0, ..., a_n)$, where $a_n = a$, be a powerful sequence for a on $S \cup \{a\}$. Then $(\eta, a_0, .., a_n)$ is an evidential sequence for a on $S \cup \{a, \eta\}$. Since a_0 requires no support in FN, then by Translation 68 it is supported by η in ES^{FN} and the evidential condition is satisfied. Let a_i be an arbitrary, nonzero element of the powerful sequence. For any set X s.t. XNa_i, we know that $X \cap \{a_0, .., a_{i-1}\} \neq \emptyset$. Thus, if at least one such supporting set X exists, it is easy to see by Translation 68 (and its continuation) that there is $X' \subseteq \{a_0, .., a_{i-1}\}$ s.t. $X'Ea_i$ and that the evidential condition is satisfied. If no supporting set X exists, then we have that a_i is supported by η, and the condition is again satisfied. Thus, we have a valid evidential sequence on $S \cup \{a, \eta\}$ for a.

Showing that if $(\eta, a_0, .., a_n)$ is an evidential sequence for a then $(a_0, .., a_n)$ is powerful follows quite similarly. We can only note on the requirement on a_0 (and basically all other arguments supported purely by η): if an argument is supported by η in ES^{FN}, then by the construction of ES^{FN}, it required no support in FN. Thus the powerful condition on a_0 is met.

Based on these results, the correspondence between the self–supporting and coherent sets of ES^{FN} and FN is straightforward. $\square$

Theorem 10.17. *Let $FN = (A, R, N)$ be an AFN and $ES^{FN} = (A', R', E)$ its corresponding EAS obtained through Translation 68. If $a \in A$ is defended by a coherent set $S \subseteq A$ in FN, then it is acceptable w.r.t. $S \cup \{\eta\}$ in ES^{FN}. If $a \in A$ is acceptable w.r.t. a self–supporting set $S' \subseteq A'$ in ES^{FN}, then it is defended by $S \cap A$ in FN.*

Proof. Let $a \in A$ be defended by a coherent set $S \subseteq A$ in FN. From Theorem 10.16 it follows that $S \cup \{\eta\}$ is self–supporting. Moreover, if a is defended by S in FN, then $S \cup \{a\}$ is coherent as well, and thus $S \cup \{a, \eta\}$ is also self–supporting in ES^{FN}. Thus, a has an evidential sequence on $S \cup \{a, \eta\}$, and by Theorem 2.99, is e–supported by $S \cup \{\eta\}$. Now let b be an argument in A s.t. bRa. Since a is defended by S in FN, then for every coherent set $C \subseteq A$ s.t. $b \in C$, $\exists c \in S, d \in C$ s.t. cRd. Thus, after Translation 68, we have that for $\{b\}R'a$ and for every (and thus also minimal) self–supporting set $C \cup \{\eta\}$ containing b, $\exists H \subseteq S, d \in C$ s.t. HRd. It is easy to see that $C \cup \{\eta\}$ is an e–supported attack against a and since $S \cup \{\eta\}$ is assumed to be self–supporting, any attack it carries out against an element of $C \cup \{\eta\}$ is also e–supported. As all attacks in ES^{FN} come from FN and η cannot attack or be attacked in the framework, we can conclude that a is acceptable w.r.t. $S \cup \{\eta\}$ in ES^{FN}.

Let $a \in A$ be acceptable w.r.t. a self supporting set $S' \subseteq A'$ in ES^{FN}. We can observe that if S' is self–supporting and e–supports a, then $S' \cup \{a\}$ is also self–supporting. Thus, by Theorem 10.16, $S' \cap A$ is coherent. From acceptability of a it follows that given any set $C \subseteq A'$ that carries out a minimal e–supported attack on a, S' support attacks a member

of C. Since S' is self–supporting, any attack carried out by it will be e–supported. By Lemma 2.108, C is self–supporting, and thus $C \setminus \{\eta\}$ is coherent in FN. Since C attacks a, there exists $C' \subseteq C$ s.t. $C'R'a$. By Translation 68 all such sets C' consist of exactly one element and $\{c\}R'a$ in ES^{FN} iff cRa in FN (observe that η does not carry out attacks). The attack by S' against C follows a similar analysis. Please note that although technically we attack only minimal e–supported attacks on a, it is easy to see that it cannot be the case that there exists an unattacked e–supported attack on a. Every such attack either contains a minimal one, or is one – either case, it still remains attacked. Consequently, it holds that for every coherent set $C \setminus \{\eta\}$ s.t. CRa, $S \setminus \{\eta\}$ contains a suitable attacker in FN. We can conclude that if a is acceptable w.r.t. s' in ES^{FN}, then it is defended by $S' \cap A$ in FN.
□

Theorem 10.18. *Let $FN = (A, R, N)$ be an AFN and $ES^{FN} = (A', R', E)$ its corresponding EAS obtained through Translation 68. If a set $S \subseteq A$ is (strongly) coherent in FN, then $S \cup \{\eta\}$ is (strongly) self–supporting in ES^{FN}. If $S \subseteq A$ is (strongly) self–supporting in ES^{FN}, then $S \cap A$ is (strongly) coherent in FN. If $S \subseteq A$ is a σ–extension in FN, where $\sigma \in \{$conflict–free, admissible, complete, preferred, grounded, stable$\}$ then $S \cup \{\eta\}$ is a σ–extension in ES^{FN}. If $S \subseteq A$ is a σ–extension of ES^{FN}, then $S \cap A$ is a σ–extension of FN.*

Proof. The coherence and self–support analysis has already been carried out in Theorem 10.16. Let us therefore continue with conflict–freeness. It is easy to see by Translation 68 that if a given set $S \subseteq A$ is conflict–free in FN, then both S and $S \cup \{\eta\}$ are conflict–free in ES^{FN}. Similarly, if a set $S' \subseteq A'$ is conflict–free in ES^{FN}, then $S' \cap A$ is conflict–free in FN. This is due to the fact that the attack relation only undergoes a minor change to account for a shift from binary to group attack and that η does participate in any attacks.

The relation between the strongly coherent and strongly self–supporting extensions follows straightforwardly from the results above. We can use Lemma 2.109, Theorems 10.16 and 10.17 in order to prove the correspondence between the admissible extensions. Please observe that $\emptyset$ and $\{\eta\}$ are trivially conflict–free, (strongly) self–supporting and admissible in ES^{FN}, even though they correspond to $\emptyset$ in FN.

We can use the relation between the admissible extensions and Theorem 10.17 in order to show that if $S \subseteq A$ is complete in FN, then $S \cup \{\eta\}$ is complete in ES^{FN}, and that if $S' \subseteq A'$ is complete in ES^{FN}, then $S' \cap A$ is complete in FN. We can observe that η is acceptable w.r.t. $\emptyset$ and will always be present in any complete extension in ES^{FN}. It is therefore easy to show that the complete extensions of both frameworks are in a one–to–one relation.

Based on the relation between the complete extensions and the fact that the grounded extensions are the least w.r.t. set inclusion complete both in FN and ES^{FN} (Theorems 2.112 and 2.95), we can show that the grounded extensions correspond as well. The same holds for the preferred semantics.

We are left with the stable semantics. Let us first show that if $S \subseteq A$ is stable in FN, then $S \cup \{\eta\}$ is stable in ES^{FN}. Based on the definition of AFN stable semantics and the previous parts of this proof, it holds that $S \cup \{\eta\}$ is complete in ES^{FN}. Therefore, it is also strongly self–supporting. Let S^+ be the deactivated set and $a \in S^+$. If a is in the set because there exists $b \in S$ s.t. bRa, then naturally $S \cup \{\eta\}$ carries out an e–supported attack on a, independently of whether a is e–supported by A' or not in ES^{FN}. Let us focus on the case when a is in the deactivated set due to lack of support. If a is not powerful in A, then by Theorem 10.16 it is not e–supported in A' in ES^{FN} and thus does not affect $S \cup \{\eta\}$. Therefore, let us assume there exists at least one powerful sequence $(a_0, ..., a_n, a)$ for A. Without the loss of generality, we can assume this sequence is minimal. Since a is in the deactivated set, part of this sequence is not present in S. Let $0 \leq i \leq n$ be the position of the first argument in the sequence that does not belong to S. If it is $i = 0$, then since a_0 requires no support and is in S^+, it has to be the case that S (and thus also $S \cup \{\eta\}$ contains an attacker of a_0. For other $t \neq 0$, since all the required support for a_i is in S but $a_i \in S^+$, then again it has to be the case that S (and thus also $S \cup \{\eta\}$ contains an attacker of a_i. This minimal powerful sequence for a in FN gives rise to a minimal evidential sequence in ES^{FN} (see proof of Theorem 10.16), from which by Theorem 2.99 we can obtain a minimal set e–supporting a. Since S can attack all sequences for a, then by Translation 68 so can $S \cup \{\eta\}$. Moreover, as $S \cup \{\eta\}$ is a self–supporting set, the attacks are e–supported. Consequently, the stability conditions of ES^{FN} are satisfied for $S \cup \{\eta\}$.

Let us show that if $S' \subseteq A'$ is complete in ES^{FN}, then $S = S' \cap A$ is complete in FN. By Theorem 2.112, every stable extension in ES^{FN} is also complete. Therefore, based on the previous parts of this proof, S is complete in FN. Assume now that S' is stable in ES^{FN}, but S is not stable in FN. This means there exists an argument $a \in A \setminus S$ that is not in the deactivated set. Consequently, it has to be the case that a is not attacked by S and either requires no support or is supported by S. Based on the previous parts of this proof this means that a is not e–support attacked by S' and has to be e–supported by S' in ES^{FN}). Since a is not in the stable extension in ES^{FN}, it has to be the case that all of the sets minimally e–supporting it are attacked. However, if S' attacks all sets minimally e–supporting a, then naturally all coherent sets containing a are also attacked in FN. Therefore, if a requires no support in FN (and therefore (a) is a powerful sequence for it), then we reach a contradiction with it not being attacked directly. If S supports a, then since S is strongly coherent in FN, it cannot be the case that at the same time all powerful sequences of a are attacked (through an element different than a). Thus, S has to be stable in FN. □

Theorem 10.19. *Let $FN = (A, R, N)$ be a strongly consistent AFN and $D^{FN} = (A, L, C)$ its corresponding ADF obtained through Translation 69. D^{FN} is a BADF. It is also in cleansed form. If FN is in minimal form, then D^{FN} is redundancy–free. If FN is weakly valid, then so is D^{FN}. If it is minimal and relation valid, then D^{FN} is relation valid. If FN is strongly valid, then D^{FN} is an AADF$^+$. If it is in addition minimal, then D^{FN} is strongly valid.*

Proof. Let us assume that D^{FN} is not a BADF. This means there exists a link $(a, b) \in L$ in D^{FN} that is neither supporting nor attacking. Consequently, there exists $E \subseteq par(b)$

s.t. $C_b(E) = in$ and $C_b(E \cup \{a\}) = out$ and a set $E' \subseteq par(b)$ s.t. $C_b(E') = out$ and $C_b(E' \cup \{a\}) = in$. Based on Translation 69, we can observe that if $C_b(E) = in$, then $E \cap F \neq \emptyset$ for every set $F \subseteq A$ s.t. FNb and there is no argument $e \in E$ s.t. eRb. Thus, if $C_b(E \cup \{a\}) = out$, then it can only be the case that aRb. Therefore, there cannot exist a set of arguments E' s.t. $C_b(E' \cup \{a\}) = in$, as by definition in every such case $C_b(E' \cup \{a\}) = out$. Hence, D^{FN} is a BADF.

W can observe that if an argument is consistent in FN, then it has an acceptance condition in D^{FN} that maps at least one set of arguments to in. In other words, the condition is not (or not equivalent to) falsum. Consequently, if there is a set that maps the condition to in, then there exists at least one decisively in interpretation for this argument. Thus, all constructed pd–functions are sound on A and we can easily create a standard evaluation containing all arguments in A (and therefore a standard evaluation for every $a \in A$). We can conclude that D^{FN} is in cleansed form.

Let s focus on the redundancy–free form and let us assume that FN is in minimal form. Let $a, b \in A$ be argument s.t. aRb and $E \subseteq A$ a minimal set of arguments s.t. $C \cap E \neq \emptyset$ where $C \subseteq A$ is a set of arguments s.t. CNb. Due to strong consistency of FN, we can observe that $E \cap \{a\} = \emptyset$. By the construction of D^{FN}, it can be seen that $C_b(E) = in$ and $C_b(E \cup \{a\}) = out$. Thus, the (a, b) link cannot be supporting, and as a result, it is not redundant.

Let now $b \in A$ and $F \subseteq A$ be an argument and a set of arguments s.t. FNb. Let $a \in F$ be a supporter of b. Due to strong consistency of FN, we can observe that it cannot be the case that aRb. Therefore, by the construction of D^{FN}, $C_b(F) = in$. Moreover, since FN is minimal, no subset of F supports b. Hence, $C_b(F \setminus \{a\}) = out$. Consequently, the (a, b) link cannot be attacking and thus, is not redundant. We can finally conclude that D^{FN} is in redundancy–free form if FN is in minimal form.

In order to see that if FN is weakly valid, then so is D, please consult the proof of Lemma 10.20.

Let us focus on relation validity and assume that FN is both in minimal and relation valid forms. Based on the previous parts of the proof, we know that D^{FN} is redundancy–free and weakly valid. With $N(a) = \{b \mid \exists F \subseteq A, b \in F \text{ s.t. } FNa\}$ we will denote the arguments supporting a. Let $E \subseteq A$ be a set of arguments s.t. $E \subseteq N(a)$ and for every $F \subseteq A$ s.t. FNa, $F \cap E \neq \emptyset$. Due to the fact that FN is relation valid, it holds that there exists a coherent subset $S \subseteq A \setminus \{a\}$ s.t. $E \subseteq S$. A coherent set can be represented as a powerful sequence in FN and this powerful sequence will correspond to an acyclic pd–evaluation in D^{FN} (see Lemma 10.20). Due to the strong consistency of FN, we can observe that E does not contain any attacker of a. Furthermore, based on the construction of D^{FN}, we can also see that any subset of parents of a evaluating C_a to in will correspond to a subset of $N(a)$ that has an element in common with every set supporting a in N. Therefore, it holds that $C_a(E) = in$ and there exists a minimal decisively in interpretation v_a for a s.t. $v_a^{\mathbf{t}} \subseteq E$. Since we could have represented S as a powerful sequence and $E \subseteq S$, we can extend this sequence with a in order to obtain a powerful sequence for a itself. Similarly, by using the analysis in Lemma 10.20, we can extend the acyclic pd–

evaluation for S by a and v_a in order to obtain an acyclic pd–evaluation for a. Due to the aforementioned relation between the sets satisfying the conditions of arguments in D^{FN} and the support in FN, we can show that every minimal decisively in interpretation for a can be used in constructing an acyclic pd–evaluation for it. Thus, D^{FN} is relation valid.

Let FN be now strongly valid. Assume D^{FN} is not an AADF$^+$. This means there exists a pd–function and a standard evaluation (F, B) created with it that we cannot transform into an acyclic one. Based on the previous analysis we can observe that given an argument a and the decisively in interpretation v_a assigned to it by the pd–function, $v_a^{\mathbf{t}}$ is a subset of $N(a)$ in FN that has an element in common with every set supporting a in N. Thus, from our pd–function we can derive a function meeting the construction requirements from Definition 4.30. Since we cannot order an evaluation based on this pd–function into an acyclic evaluation, it can be shown that we cannot create a powerful sequence with the associated function in FN. Thus, we reach a contradiction with strong validity of FN. We can conclude that D^{FN} has to be an AADF$^+$.

Let now FN be minimal and strongly valid. This means that our D^{FN} is redundancy–free and an AADF$^+$. In addition, based on the previous parts of this proof, it is in cleansed form. Consequently, by Theorem 4.43, D^{FN} is strongly valid. □

Lemma 10.20. *Let $FN = (A, R, N)$ be a strongly consistent AFN, $D^{FN} = (A, L, C)$ its corresponding ADF obtained through Translation 69 and $E \subseteq A$ a set of arguments. For a given powerful sequence for an argument $e \in E$ we can construct a corresponding acyclic pd–evaluation and vice versa. E is coherent in FN iff it is pd–acyclic in D^{FN}.*

Proof. Let $E \subseteq A$ be a set of arguments, $e \in E$ and $a_0, ..., a_n$ a powerful sequence for e. We will show it satisfies the pd–sequence requirements.

First of all, the $a_n = e$ condition is satisfied. Secondly, we have that for a_0 there is no $B \subseteq A$ s.t. BNa_0. This means that that a_0 faces only binary attack and its condition basically consists only of the att part. Based on the explanations we presented in Section 5.7 concerning AFs, a_0 has a single minimal decisively in interpretation that maps every attacker of a_0 to $\mathbf{f}$. The $\mathbf{t}$ part is empty and thus the interpretation satisfies the pd–evaluation criterion of a_0.

Finally, in the powerful sequence we have that for every nonzero a_i, it holds that for each $B \subseteq A$ s.t. BNa_i, $B \cap \{a_0, ..., a_{i-1}\} \neq \emptyset$. Let $E_i = \{a_0, ..., a_{i-1}\} \cap par(a_i)$. Since FN is strongly consistent, no argument in E_i is an attacker of a_i. Thus, by the construction of D^{FN} it holds that $C_{a_i}(E_i) = in$. An interpretation assigning $\mathbf{t}$ to E_i and $\mathbf{f}$ to $A \setminus E_i$ will be a decisively in interpretation for a_i. Thus, we can extract a minimal interpretation v from it, which will assign $\mathbf{t}$ to a subset $E_i' \subseteq E_i$ and $\mathbf{f}$ to all those arguments $b \in A$ s.t. bRa_i. Based on this, we can conclude that v satisfies the pd–sequence condition. Therefore, we obtain an acyclic pd–evaluation $((a_0, ..., a_n), \bigcup_0^n \{a_i\}^-)$ for e on E corresponding to the powerful sequence $(a_0, ..., a_n)$.

Let $E \subseteq A$ be a set of arguments, $e \in E$ and $((a_0, ..., a_n), B)$ an acyclic pd–evaluation for e. We will show that the sequence part satisfies the powerful conditions. Again, the $a_n = e$ condition is easily met. Since the decisively in interpretation for a_0 consists only from negative mappings, which by Translation 69 come from the attackers of a_0. As a_0 is strongly consistent, none of those attackers is also a supporter, and thus we can conclude that there exists no supporting set for a_0 and that another powerful requirement is met. Now, we know that for every nonzero a_i and its minimal decisively in interpretation v_i, $v_i^{\mathbf{t}} \subseteq \{a_0, ..., a_{i-1}\}$. By construction of the arguments we know that $\forall Z \subseteq A$ s.t. ZNa_i, $v_i^{\mathbf{t}} \cap Z \neq \emptyset$. Consequently, $Z \cap \{a_0, ..., a_{i-1}\} \neq \emptyset$ and the final powerful requirement is satisfied. Therefore, the pd–sequence of the evaluation produces a powerful sequence.

The correspondence between coherent and pd–acyclic sets follows straightforwardly from this analysis. □

Lemma 10.21. *Let $FN = (A, R, N)$ be a strongly consistent AFN, $D^{FN} = (A, L, C)$ its corresponding ADF obtained through Translation 69. A set of arguments $E \subseteq A$ is strongly coherent in FN iff it is a pd–acyclic conflict–free extension of D^{FN}.*

Proof. Let us assume that E is strongly coherent in FN, but not pd–acyclic conflict–free in D^{FN}. By Lemma 10.20 we know that every argument in E possesses a pd–acyclic evaluation on E. What remains to be shown is that every argument has an evaluation on E that is also unblocked (as a result of this, the condition of every argument will also be satisfied). By Lemma 10.20 we can create an evaluation corresponding to the powerful sequence of e on E. The blocking set of such an evaluation corresponds exactly to the union of attackers of all its sequence members. As all the members of the pd–sequence of this evaluation are in E, it has to be the case that an element of the blocking set is accepted. However, it would clearly breach the conflict–freeness of E in FN and we reach a contradiction. Therefore, E is pd–acyclic conflict–free in D^{FN}.

Let us now assume that E is pd–acyclic conflict–free in D^{FN}, but not strongly coherent in FN. By Lemma 10.20, E is at least coherent. If E is not conflict–free in FN, it means that $\exists x, y \in E$ s.t. xRy. However, by the strong consistency of FN and Translation 69, it would mean that $C_y(E \cap par(y)) = out$. Consequently, E could not have been conflict–free in D^{FN}, and as every pd–acyclic conflict–free extension is also just conflict–free, we reach a contradiction. Hence, if E is pd–acyclic conflict–free in D^{FN}, then it is strongly coherent in FN. □

Lemma 10.22. *Let $FN = (A, R, N)$ be a strongly consistent AFN, $D^{FN} = (A, L, C)$ its corresponding ADF obtained through Translation 69. Let $E \subseteq A$ be strongly coherent in FN and thus pd–acyclic conflict–free in D^{FN}. Then E^{att} coincides with the acyclic discarded set of E.*

Proof. If every coherent set containing a is attacked by E, it means that every powerful sequence for a is attacked by E. By Lemma 10.20, we have that every powerful sequence corresponds to an acyclic pd–evaluation. As seen in the proof, attackers of the members of this sequence form the blocking set of the evaluation. Thus, if E attacks a member of the powerful sequence, it means that an argument from the blocking set of the evaluation is in E. Therefore, the evaluation is blocked, and whatever is in E^{att} is in E^{a+}.

Now let us assume there is an argument $a \in E^{a+}$, but not in E^{att}. This means that a has an unattacked powerful sequence, but every acyclic pd–evaluations (F, B) for a is blocked through the blocking set (see Lemma 2.128). By Lemma 10.20 we can construct a pd–evaluation corresponding to the unattacked sequence. Since the blocking set of the evaluation is composed of the attackers of members of the powerful sequence, it cannot be the case that there is no $b \in E$ attacking the sequence and at the same time $E \cap B \neq \emptyset$. We reach a contradiction. Therefore, whatever is in E^{a+} is also in E^{att}. □

Theorem 10.23. *Let $FN = (A, R, N)$ be a strongly consistent AFN and $D^{FN} = (A, L, C)$ its corresponding ADF obtained through Translation 69. Let $E \subseteq A$ be strongly coherent in FN and thus pd–acyclic conflict–free in D^{FN}. Then E defends an argument $a \in A$ in FN iff this argument is decisively in w.r.t. v_E^a in D^{FN}.*

Proof. We will use the formulation of defense in AFNs from Lemma 2.93.

Let us assume that a is defended in FN, but is not decisively in w.r.t. v_E^a. This means there exists at least one completion v' of the range interpretation that outs the acceptance condition of a. Let $E' = v'^{\mathbf{t}}$. According to Translation 69, the condition of a is not satisfied if there exists $b \in E'$ s.t. bRa or there exists $C \subseteq A$ s.t. CNa and $C \cap E' = \emptyset$. If it is the first case, then from the fact that $E^{att} = E^{a+}$ by Lemma 10.22, it follows that there is an attacker b of a not included in E^{att}. Thus, a could have not been defended in FN. If it is the latter case, it means that there exists $C \subseteq A$ s.t. $CNa \land C \cap E = \emptyset$ as well. Consequently, $E \cup \{a\}$ could not have been coherent. We reach a contradiction. Therefore, if an argument a is defended by E in FN, then it is decisively in w.r.t. v_E^a in D^{FN}.

Let us now assume that a is decisively in w.r.t. v_E^a, but is not defended in FN. This means that either there is an argument $b \in A$ s.t. bRa and $b \notin E^{att}$, or $E \cup \{a\}$ is not coherent. By Translation 69 and Lemma 10.22, it is easy to see that if it were the first case, then a could not have been decisively in w.r.t. the acyclic range of E. Let us thus assume that the issue lies in the coherence. Since we know that E is strongly coherent, a is the only argument that would not have a powerful sequence on $E \cup \{a\}$. This means that either there is no powerful sequence for a to start with, or there is a set $C \subseteq A$ s.t. CNa and $C \cap E = \emptyset$. If it is the first case, then by Lemma 10.20 there is no acyclic pd–evaluation for a in D. Consequently, it has to mapped to false by v_E^a and is therefore decisively out w.r.t. it by Proposition 2.150. We reach a contradiction with the assumption it is decisively in. If it is the latter case, then by the Translation 69 the acceptance condition of a could not have been satisfied by E. Hence, a could not have been decisively in w.r.t. v_E^a and we reach a contradiction. We can therefore conclude that if a is decisively in w.r.t. v_E^a in D^{FN}, then it is defended by E in FN. □

Theorem 10.24. *Let $FN = (A, R, N)$ be a strongly consistent AFN, $D^{FN} = (A, L, C)$ its corresponding ADF obtained through Translation 69. A set of arguments $E \subseteq A$ is coherent in FN iff it is pd–acyclic in D^{FN}. E is strongly coherent in FN iff it is pd–acyclic conflict–free in D^{FN}. E is a σ–extension of FN, where $\sigma \in \{$admissible,* *complete, preferred$\}$ iff it is an aa–σ–extension of D^{FN}. E is stable in FN iff it is stable in D^{FN}. E is grounded in FN iff it is acyclic grounded in D^{FN}.*

Proof. Let E be an admissible extension in FN. By Lemma 10.21 and Theorem 10.23 we know that it is pd–acyclic conflict–free in D^{FN} and that all arguments in E are decisively in w.r.t. v_E^a. Since the members of the blocking sets correspond to the attackers of the arguments, they are naturally falsified in the range interpretation. Consequently, all aa–admissible criterions are satisfied. The other way around follows straightforwardly from the theorems.

We now know that the admissible extensions of FN and D^{FN} coincide. Thus, the maximal w.r.t. set inclusion admissible sets are the same, and E is preferred in FN iff it is aa–preferred in D^{FN}.

The completeness follows straightforwardly from admissibility and Theorem 10.23. We can use Theorems 2.95 and 2.158 in order to show that E is grounded in FN iff it is acyclic grounded in D^{FN}.

What remains to be shown is the correspondence of stable semantics. Let E be AFN stable. By Lemma 10.21 we know that E is then at least pd–acyclic conflict–free in D^{FN}. It is easy to see by the definition of the deactivated set and Translation 69, that the acceptance condition of every argument $a \notin E$ will be out. Thus, E satisfies the model criterion and we can conclude that it is ADF stable.

Let now E be ADF stable.Since E is also a model, then we know by Lemma 2.159 that $E^{a+} = A \setminus E$. We know it is pd–acyclic conflict–free, thus at least strongly coherent in FN by Lemma10.21. By this and Lemma 10.22 we can conclude that E^{a+} coincides with E^{att}. Thus, by Lemma 2.94 E is AFN stable. □

15.9 Translating EASs: Proof Appendix

Theorem 11.3. *Let $ES = (A, R, E)$ be an EAS and $SF^{ES} = (A', R')$ its corresponding SETAF obtained by Translation 72. If $S \subseteq A$ is conflict–free in ES, then there might not exist a conflict–free extension $S' \subseteq A'$ of SF^{ES} s.t. $S = \bigcup S'$. If $S \subseteq A$ is strongly self–supporting in ES, then there exists a conflict–free extension $S' \subseteq A'$ of SF^{ES} s.t. $S = \bigcup S'$. If S is a σ–extension of ES, where $\sigma \in \{$admissible, complete, preferred, grounded stable$\}$, then there exists a σ–extension $S' \subseteq A'$ of SF^{ES} s.t. $S = \bigcup S'$.*

If $S' \subseteq A'$ is conflict–free in SF^{ES}, then $\bigcup S'$ is strongly self–supporting (and thus also conflict–free) in ES. If $S' \subseteq A'$ is a σ–extension of SF^{ES}, then $\bigcup S'$ is a σ–extension of ES.

Proof. The fact that not every conflict–free extension of ES has an associated conflict–free extension in SF^{ES} comes simply from the fact that arguments not possessing evidential sequences can participate in conflict–free extensions, but not in self–supporting ones, which form the arguments in SF^{ES}.

Let $S \subseteq A$ be a strongly self–supporting set of ES. Let $S' = \{X \mid X$ is a minimal self–supporting set for an argument $a \in S$ s.t. $X \subseteq S\}$ be its associated set in SF^{ES}. We can observe that $\bigcup S' = S$; every argument $a \in S$ possesses an evidential sequence on S and thus a suitable self–supporting set for a can be extracted from it. Let us assume that S' is not conflict–free in SF^{ES}. This means there is an argument $X \in S'$ and a subset $F \subseteq S'$ s.t. $FR'X$. However, by the construction of R', this means there exists an argument $x \in X$ (and thus in S) and a set $F' \subseteq \bigcup F$ (and thus also a subset of S) s.t. $F'Rx$. Hence, S could not have been conflict–free in ES and we reach a contradiction. We can therefore conclude that if S is strongly self–supporting in ES, then it is conflict–free in SF^{ES}.

Let $S' \subseteq A'$ be a conflict–free extension of SF^{ES} and $S = \bigcup S'$ the associated set in ES. It can be easily shown that S is self–supporting. What remains to be proved is that it is also conflict–free. Let us assume it is not the case, i.e. there exists $X \subseteq S$ and $y \in S$ s.t. XRy. From the fact that $X \subseteq S$, there must be a set of SETAF arguments $X' \subseteq S'$ s.t. $X \subseteq \bigcup X'$. Similarly, there has to exist an argument $Y \in S'$ s.t. $y \in Y$. However, by the construction of R', this means that $X'R'Y$. Hence, S' cannot be conflict–free in SF^{ES} and we reach a contradiction. We can conclude that S is strongly self–supporting in ES.

Let $S \subseteq A$ be an admissible set of ES and S' its associated conflict–free extension of SF^{ES} constructed in the way described in the first paragraph. Let $a \in A$ be an argument acceptable w.r.t. S. This means that $S \cup \{a\}$ is a self–supporting set for a; from it we can extract a minimal self–supporting set $X \subseteq S \cup \{a\}$ for a, which becomes an argument $X \in A'$. Let us assume that S' does not defend X. Consequently, there exists a set of SETAF arguments $Y \subseteq A'$ s.t. $YR'X$, but there is no subset $V \subseteq S'$ and no SETAF argument $y \in Y$ s.t. $VR'y$. Based on the construction of SF^{ES}, we can extract a set of EAS arguments $Y' \subseteq \bigcup Y$ and an argument $x \in X$ s.t. $Y'Rx$. Due to the fact that every argument in Y represents a self–supporting set, $\bigcup Y$ is an e–supported attack against x. Since $x \in S \cup \{a\}$, then S defends x in ES and there has to be a subset of S attacking an argument in $\bigcup Y$. Thus, there is a subset of S' attacking an argument in Y. We reach a contradiction. Hence, if S defends an argument $a \in A$, then S' defends an argument containing a as well. Thus, it can be shown that S' is admissible in SF^{ES}.

Let now $S' \subseteq A'$ be an admissible extension of SF^{ES} and $S = \bigcup S'$ the associated strongly self–supporting set of ES. Let $X \in S'$ be a SETAF argument defended by S' in SF^{ES}. However, assume that there is an argument $x \in X$ that is not acceptable w.r.t. S in ES. Due to the fact that S is self–supporting, we can observe that S e–supports every argument in S, including x. Thus, it is the attacks that we need to be concerned with. Let $T \subseteq A$ be a minimal e–supported attack against x. By the construction of R', we can find a set $T' \subseteq A'$ s.t. $T = \bigcup T'$ and $T'R'X$. Due to admissibility of S', we can find a subset of S' attacking a member of T'. Consequently, we can find a subset of S attacking an argument in T as well. Thus, we reach a contradiction, and S in fact defends x. Hence, if S' is admissible in SF^{ES}, then $S = \bigcup S'$ is admissible in ES.

Let $S \subseteq A$ a complete extension of ES and S' its associated admissible extension of SF^{ES} constructed in the way described in the first paragraph. Let us assume that SF^{ES} is not complete; this means there exists an argument $X' \in A' \setminus S'$ that is defended by S'. By using the previous analysis, we can show that if S' defends X', then every argument $x \in X'$ is defended against attacks by S in ES. The issue lies with coherence. Since

X' is a self–supporting set, it can be represented as an evidential sequence $(a_0, ..., a_n)$. Due to the fact that $a_0 = \eta$, a_0 is trivially contained in S. Based on the construction of the sequence, we can observe that $\{a_0\}Ea_1$. Therefore, $S \cup \{a_1\}$ is a coherent set, and as it is defended against attacks as well, it holds that a_1 is acceptable w.r.t. S. By the completeness of S, it has to be the case that $a_1 \in S$. We can continue this analysis until a_n and the conclusion that $a_n \in S$. This, by the construction of S', means that $X' \in S'$ and we reach a contradiction. Hence, if S is complete in ES, then so is S' in SF^{ES}.

Let $S' \subseteq A'$ be a complete extension of SF^{ES} and $S = \bigcup S'$ its associated admissible extension of ES. Assume that S is not complete; this means there exists an argument $x \in A \setminus S$ that is acceptable w.r.t. S. This means that $S \cup \{a\}$ is a self–supporting set for x. We can extract from it a minimal set $X \subseteq S \cup \{a\}$ for x that will become an argument in A'. We can observe that $X \notin S'$ (otherwise, x would have appeared in S). Based on the previous analysis of conflicts, we can show that due to the acceptability of x w.r.t. S, X has to be defended by S'. Since $X \notin S'$, we reach a contradiction with completeness of S'. We can therefore conclude that if S' is complete, then so is S.

We will now show that there is a one–to–one correspondence between the EAS and SETAF complete extensions. Assume it is not the case and that there exist two complete extensions $S', S'' \subseteq A'$ of SF^{ES} s.t. $\bigcup S' = \bigcup S''$. This means that there exists an argument $X \in A'$ that is contained in $S' \setminus S''$ or $S'' \setminus S'$. Without the loss of generality, let us focus on the first case. This means that S' defends X and S'' does not. Let $Y \subseteq A'$ be a set of SETAF arguments attacking X. By the construction of R', this means that there is an argument $x \in X$ which is attacked by a subset of $\bigcup Y$. However, we can observe that since $\bigcup S' = \bigcup S''$, there has to be an argument $X' \in S''$ s.t. $x \in X'$. Consequently, as S'' has to defend X', then it attacks an argument in Y. Hence, S'' defends X, and we reach a contradiction. Therefore, $S' \subseteq S''$. We can perform a similar analysis for the other way and show that $S'' \subseteq S'$. Thus, $S' = S''$ and the correspondence between the EAS and SETAF complete extensions has to be one–to–one.

It is easy to show that given two complete extensions $S \subset T$ in ES, their associated complete extensions S' and T' in SF^{ES} are also of the form $S' \subset T'$. Thus, by combining the one–to–one relation between the complete extensions, Theorems 2.10 and 2.112, we can prove the stated relation between the preferred and grounded extensions of ES and SF^{ES}.

What remains to be shown is the relation between the stable extensions. Let $S \subseteq A$ be a stable extension of ES and $S' \subseteq A'$ the associated complete extension of SF (see Theorems 2.9, 2.10 and 2.112) created as in the first paragraph of this proof. Assume it is not stable; this means there exists an argument $X' \in A' \setminus S'$ that is not attacked by S'. Thus, it can be shown that there is no subset of S attacking any of the arguments $x \in X'$. Based on the completeness and construction of S', we can observe that X' is not a subset of S, i.e. there exists at least one argument $y \in X'$ s.t. $y \notin S$. We can observe that X' is a self–supporting set for y; from it we can extract a minimal one. Consequently, due to the fact that no argument in this set is attacked by S, we reach a contradiction with the stability of S. Hence, we can conclude that if S is stable in ES, then so is S' in SF^{ES}.

Let $S' \subseteq A'$ be a stable extension of SF^{ES} and $S = \bigcup S'$ its associated complete extension of ES. Let us assume that S is not stable; this means there exists an argument $x \in A \setminus S$ and a minimal self–supporting set $X \subseteq A$ for x s.t. no argument in X is attacked by S. We can observe that X will become an argument in A'. Based on the previous analysis we can observe that if there is no subset of S attacking an argument in X, then there is no subset of S' attacking X either. Thus, we reach a contradiction with the stability of S' in SF^{ES}. We can thus conclude that if S' is stable in SF^{ES}, then so is $S = \bigcup S'$ in ES. □

Theorem 11.4. *Let $ES = (A, R, E)$ be an EAS and $SF^{ES} = (A', R'')$ its corresponding attack propagated SETAF obtained by Translation 73. If $S \subseteq A$ is strongly self–supporting in ES, then it is conflict–free in SF^{ES}. It does not necessarily hold for conflict–free semantics. If S is a σ–extension of ES, where $\sigma \in \{$admissible, complete, preferred, grounded, stable$\}$, then it is a σ–extension of SF^{ES}. If $S' \subseteq A'$ is a σ'–extensions of SF^{ES},where $\sigma' \in \{$conflict–free, complete, preferred, grounded, stable$\}$, then it is also a σ'–extension of ES. Not every conflict–free extension of SF^{ES} is strongly self–supporting in ES.*

Proof. By Theorem 4.20, the extensions of ES and ES^{wv} under semantics that are self–supporting coincide, what needs to be shown is that the extensions of $ES^{wv} = (A', R', E')$ and $SF^{ES} = (A', R'')$ coincide as well.

Let us consider a simple weakly valid framework $(\{\eta, a, b, c\}, \{(\{a\}, b)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{b\}, c)\})$. We can observe that $\{a\}$ has the power to attack all self–supporting sets for c. Thus, in the target SETAF, $\{a, c\}$ is not a conflict–free extension. Unfortunately, it is such in our source EAS.

Let $S \subseteq A'$ be a strongly self–supporting set of ES^{wv}. For every $a \in S$, S will be in $a's$ collection of self–supporting sets. Since S is conflict–free, no subset of S will be in the set Z containing all and only attackers of S. Consequently, no nonempty subset of S will be a propagated attack on any of the elements of S, as no set in Z will be its subset. We can thus conclude that S is conflict–free in SF^{ES}.

Let $S \subseteq A'$ be a conflict–free extension of SF^{ES}. Since R'' contains the attacks from R', it is easy to see that S is also conflict–free in ES^{wv}.

Let $S \subseteq A'$ be an admissible extension of ES^{wv} and $a \in A'$ an argument acceptable w.r.t. S. Therefore, S e–supports a, and $S \cup \{a\}$ is in the collection of self–supporting sets of a. Let Z be the collection of all sets of arguments attacking $S \cup \{a\}$, X the collection of all sets of arguments attacking S and Y the collection of sets of arguments attacking a in ES^{wv}. It is easy to see that $Z = X \cup Y$. We know that any propagated attack on a will have at least one set in Z as a subset. If this set originates in Y, then since a is acceptable w.r.t. S we know that every minimal self–supporting set containing this set (i.e. a minimal e–supported attack) is attacked by S. Consequently, there will exist an attack in R'' from a set $S' \subseteq S$ to this set. If this set originates in X, then as S is admissible and all of its members are acceptable w.r.t. it, then again there will be a propagated attack in R'' from a set $S'' \subseteq S$ to this set. Thus, any propagated attack on a will be in turn attacked and we can conclude that an argument acceptable w.r.t. S in ES^{wv} is defended in SF^{ES}. Hence, S is admissible in SF^{ES}.

We can again consider the framework from the second paragraph of this proof. We can observe that a will be an initial argument in the target SETAF. Hence, $\{a\}$ is an admissible extension. Unfortunately, due to the absence if η, it is neither self–supporting nor admissible in the source EAS.

Let $S \subseteq A'$ be a complete extension in ES^{wv}. By the previous parts of this proof, we know that S is admissible in SF^{ES} and that whatever is acceptable w.r.t. S in ES^{wv}, is also defended in SF^{ES}. What remains is to be proved is that there is no argument $b \in A'$ which is defended by S in SF^{ES}, but is not acceptable w.r.t. S in ES^{wv}.

Let us assume it is not the case and that b is defended by S in SF^{ES}, but it is either not e–supported by S or there exists an e–supported attack on b not countered by S in ES. Let $(a_0, ..., a_n)$ be an arbitrary evidential sequence for b. We can observe that b is e–supported by A' (i.e. we are dealing with a weakly valid framework) and therefore at least one such sequence will exist. Since $a_0 = \eta$ and S is admissible, it must be the case that $a_0 \in S$. Let us thus assume that a_1 is not in S; it must be then the case that it is not acceptable w.r.t. S. From evidential conditions it follows that $\{\eta\}Ea_1$ and as $\eta \in S$, then S e–supports a_1. Consequently, there must exist a minimal e–supported attack $T \subseteq A$ on a_1 which is not attacked by S. Let $X_1, ..., X_n$ be the subsets of T that carry out the attack, i.e. X_iRa. By construction, X_i is a possible propagated attack against a_1. Since T is a possible self–supporting set for any of its elements and no subset of S carries out an attack against any argument in T, no subset of S will be a propagated attack against any argument in T and, in particular, in any X_i. If it is the case that a_1 in S, we can continue with a_2 and repeat the same reasoning until we reach $a_n = b$ and the conclusion that there must have been a minimal e–supported attack on some argument in the sequence not attacked by S. We can repeat this reasoning for any evidential sequence for b, each time coming to a conclusion that there must have been an e–supported attacker not covered by S. By combining the subsets of the attackers that perform the actual attack through R', we can build a propagated attack on b, and since none of its elements is attacked in R'' by any subset of S, it could thus not have been the case that S defended b in SF^{ES}. We reach a contradiction. Consequently, S is complete in SF^{ES}.

Let now $S \subseteq A$ be a complete extension of SF^{ES}. Before we continue, we will show that if an argument $a \in A'$ is defended by S in SF^{ES}, then so is at least one self–supporting set containing a.

Let $\{Z'_1, ..., Z'_n\}$ be the collection of sets of arguments attacking a through R''. If there exists no such set, then by the construction of SF^{ES} and the fact that every argument has at least one self–supporting set, it means that for at least one self–supporting set C of a, the collection of sets attacking arguments in C is empty. Consequently, C is not attacked, ans as C is a self–supporting set of any of its members and it is not attacked, then it cannot be the case that there is a propagated attack against any of its members. We thus obtain a trivial case of defense for the arguments in C.

Let us therefore assume there is at least one Z'_i. Since a in defended in SF^{ES}, for every attack Z'_i on a there exists a propagated attack from some subset of S to a member of Z'_i. The set of such attacked arguments will have at least one common element with every attacker set of at least one self–supporting set of a in ES. If it were not the case and for every self–supporting set there existed at least one attacker set Z in R' that would not have a member attacked by S in R'', then we could always create a propagated attack from such sets. Moreover, S would not defend a from this attack. Therefore, let C be the self–supporting set of a whose attackers sets are covered by S and let $(a_0, ..., a_n)$ be the evidential sequence for a in this set. Without the loss of generality, we can just focus on this sequence – if all attackers of C are attacked, then so are the ones of a minimal evidential sequence for a contained in C. What we need to show is that the fact that all R' attackers of the sequence are attacked by S through R'' is sufficient for all of its R'' attackers to be attacked through R".

Since $a_0 = \eta$ and η is an unattacked argument both in SF^{ES} and ES^{wv}, it is trivially defended by any set of arguments. Let us thus start with a_1 and its set attackers $B_{11}, ..., B_{1n_1}$. Since $\{a_0, a_1\}$ is the single minimal self–supporting set for a_0, then for every propagated attack F on a_0, $\exists B_{1i}$ s.t. $B_{1i} \subseteq F$. Since for any set $B_{11}, ..., B_{1n_1}$ we can find a subset of S attacking it through R'', then naturally every attack on a_1 through R'' is also attacked by S. Thus, a_0 is defended.

Let us now focus on a_2 and its set attackers $B_{21}, ..., B_{2n_2}$. The set $\{a_0, a_1, a_2\}$ is a self–supporting set for a_2 and $X = B_{11}, ..., B_{1n_1}, B_{21}, ..., B_{2n_2}$ is the collection of the sets attacking it. Consequently, for every propagated attack F on a_2, should it exist, it holds that $\exists B \in X$ s.t. $B \subseteq F$. Since for any set in X we can find a subset of S attacking it through R'', then every propagated attack on a_2 through R'' is also attacked and a_2 is defended. Should there be no propagated attack against a_2, by discussion above a_2 is trivially defended by S.

We can continue repeating this proof until we cover all members of the sequence, showing that they were all defended by S. Consequently, if an argument $a \in A'$ is defended by S in SF^{ES}, then so is at least one evidential sequence/self–supporting set containing it.

Let us now assume that $S \subseteq A$ is a complete extension of SF^{ES}. We know it is at least conflict–free in ES^{wv}. As S contains all arguments it defends, then if it contains an argument a, it also contains at least one evidential sequence for it. Consequently, S e–supports all of its members. Let us now focus on attacks. Assume that even though every argument in S is defended by S in SF^{ES}, there exists a minimal e–supported attack T on an argument $a \in S$ which is not attacked by S. Let $T' \subseteq T$ be the subset carrying out the attack through R', i.e. $T'R'a$. By the construction of SF^{ES} it is a propagated attack on a in SF^{ES}. Since T' is not attacked by S, then it cannot be the case that any subset of S carries out a propagated attack against any member of T' and thus a could not have been defended in SF^{ES}. We reach a contradiction. Hence, S is at least admissible in ES^{wv}. If it is not complete, then there is an argument $a \notin S$ acceptable w.r.t. S in ES^{wv}, but not defended in SF^{ES}. However, by the proof that every argument acceptable w.r.t. an admissible set in ES^{wv} is defended by the set in SF^{ES} we can see that cannot be the case.

Thus, S is complete in ES^{wv}.

Since we know that complete extensions coincide between ES^{wv} and SF^{ES}, by Theorem 2.24 and 2.112 we can conclude that so do preferred and grounded. What remains to show is the stable semantics. Let $S \subseteq A'$ be a stable extension in ES^{wv}. It is thus self–supporting conflict–free and attacks every evidential sequence of an argument $a \notin S$. By the analysis above it is easy to see that the set will be conflict–free in SF^{ES} and for every such a there will exist an according propagated attack. Consequently, S will be stable in SF^{ES}.

Let S be a stable extension of SF^{ES}. By the analysis above and Theorems 2.24 and 2.23 we know it is complete and thus self–supporting in ES^{wv}. Since every argument $a \notin S$ is attacked in R'', then by construction of R'' we know that every self–supporting set containing a is attacked by S in ES^{wv}. From this follows easily that S is stable in ES^{wv}. $\square$

Theorem 11.6. *Let $ES = (A, R, E)$ be an EAS and SF^{ES} its corresponding defender SETAF obtained by Translation 75. By $S_{np} = \{a' \mid$ there is no self–supporting set containing $a\} \cup \{a' \mid$ for every self–supporting set C for a, $\exists S' \subseteq S, c \in C \setminus \{a\}$ s.t. $(S', c) \in R\}$ we will denote the primed arguments corresponding to a subset of S^+ in which every argument a either has no self–supporting set or every such set is attacked by S on an argument different from a.*

If a set $S \subseteq A$ is conflict–free in ES, then it is conflict–free in SF^{ES}. The set $S \cup S_{np}$ is not necessarily conflict–free in SF^{ES}. If a set $S \subseteq A$ is strongly self–supporting in ES, then $S \cup S_{np}$ is conflict–free in SF^{ES}. If S is a σ–extension of ES, where $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$, then $S \cup S_{np}$ is a σ–extension of SF^{ES}.

If a set $S' \subseteq A'$ is a σ'–extension of SF^{ES}, where $\sigma' \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, then $S = S' \cap A$ is a σ'–extension of ES. If S' is conflict–free in SF^{ES}, then $S = S' \cap A$ might not be strongly self–supporting in ES.

Proof. Let $S \subseteq A$ be a conflict–free extension of ES. We can observe that all attacks in R' that occur only between the arguments in A correspond precisely to R. Therefore, S is easily conflict–free in SF^{ES}. However, $S' = S \cup S_{np}$ is not necessarily conflict–free in SF^{ES}. Take for example the framework $(\{a, b, c, \eta\}, \{(a, b)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{b\}, c)\})$. The set $\{a, c\}$ is conflict–free in it, despite the fact that a attacks the only supporter b of c. Thus, $\{a, c\}_{np} = \{c'\}$, and we obtain the set $\{a, c, c'\}$ on the SETAF side. Clearly, a primed argument cannot appear in a conflict–free extension along with its original version.

Let $S \subseteq A$ be a strongly self–supporting set of ES. Let us assume that $S' = S \cup np(S)$ is not conflict–free. Since S is itself conflict–free in SF^{ES}, this means that for some argument $a' \in np(S)$, either $a \in S$ or there exists $X \subseteq S$ s.t. $XR'a'$. However, a' is added to $np(S)$ if and only if S attacks all evidential sequences of a on non–a elements. If it were the case that $a \in S$, then S would contain an evidential sequence for a and thus would have to attack itself. We thus breach the conflict–freeness of S in ES. If $X \subseteq S$, then by the construction of SF^{ES}, $X \cup \{a\}$ can be ordered into an evidential sequence for

a. Hence, S would have to attack an argument in X, and we reach a contradiction again. Thus, S' is conflict–free in SF^{ES}.

Let $S' \subseteq A'$ be a conflict–free extension of SF^{ES}. Since every attack from R is contained in R', it follows easily that $S = S' \cap A$ is a conflict–free set of ES. However, not every conflict–free extension of SF^{ES} has to correspond to a strongly self–supporting set in ES. We can consider a trivial framework $(\{\eta, a\}, \emptyset, \{(\{\eta\}, a)\})$; the associated SETAF is $(\{\eta, a, a'\}, \{(\{\eta\}, a'), (\{a'\}, a)\})$. We can observe that $\{a\}$ is a conflict–free extension of our target framework. However, due to the absence of η, it is neither self–supporting nor strongly–self supporting in our source EAS.

Let $S \subseteq A$ be an admissible extension of ES. From the previous parts of this proof we know that $S' = S \cup np(S)$ is conflict–free in SF^{ES}. What remains to be shown is that S' defends all of its members in SF^{ES}. Let a be an argument in S. We can naturally exclude η from this analysis. Based on the construction of SF^{ES}, we can observe that $a' \in A'$ and that $\{a'\}R'a$. From the fact that S is self–supporting, it follows that there exists $X \subseteq S$ s.t. $XR'a'$. Since $S \subseteq S'$, it holds that S' defends the arguments it contains from attacks carried out by primed arguments. What we now need to show is that a is defended by S' from attacks carried out by arguments in S. Since a is acceptable w.r.t. S in ES, then every minimal e–supported attack T on a is attacked by S. Let $T' \subseteq T$ be a subset of T s.t. $T'Ra$ (and thus $T'R'a$). There exists an argument $t \in T'$ s.t. all of its minimal evidential sequences are attacked – if it were not the case, then every t would have an unattacked sequence which we could combine into a single unattacked e–supported attack on a. Thus, all evidential sequences for t are attacked; if it is by an attack on t itself, then obviously S' attacks t as well and a is defended. If it is not through t, then due to the presented construction, $np(S)$ contains t' and again a is defended by S'. Moreover, if a set of arguments attacking a contains an element not possessing an evidential sequence at all, then by the construction of S', the suitable primed argument is in S'. Finally, we need to analyze the status of arguments in $np(S)$. We can observe that if $a' \in np(S)$, then every evidential sequence for a is attacked by S on a non–a member. Thus, we can observe that every set of arguments attacking a' in R' is attacked by a subset of S (and thus a subset of S'). We can therefore finally conclude that S' is admissible in SF^{ES}.

Let $S' \subseteq A'$ be an admissible extension of SF^{ES}. By the previous parts of this proof we know that the set $S = S' \cap A$ is conflict–free in ES. However, let us assume that S is not self–supporting, i.e. there exists n argument $a \in S$ that does not have an evidential sequence on S. If a does not have an evidential sequence on A to start with, then by Translation 75 it is attacked by a' and the same time there is no set of arguments attacking a'. This contradicts the admissibility of S' in SF^{ES}. Consequently, there exists at least one (minimal) evidential sequence for a on A. If $a = \eta$, then it is trivially e–supported by S. Otherwise, we know that a has an attacker a' in A' and that every set of arguments attacking a' corresponds to the members of an evidential sequence for a that precede a. Thus, if there is no evidential sequence on S for a, then a could not have been defended by S' against a' in SF^{ES}. This leads us to a conclusion that S is self–supporting in ES.

Let us now assume that S is not admissible in ES. This means that there exists an argument $a \in S$ and a minimal e–supported attack T on a s.t. no element of T is attacked by S. Since T is a minimal e–supported attack, it is also a self–supporting set by Lemma 2.108. Thus, it can be represented as an evidential sequence; let $(b_0, .., b_n)$ be this sequence. Since $b_0 = \eta$, it cannot be attacked in any of the frameworks. Let us focus on b_1; we know that there exists an argument b'_1 in A' attacking b_1. As $\{b_0\}R'b'_1$ and there is no set in S (and no set in S') attacking b_0, b'_1 cannot be in S' due to admissibility. Moreover, no set in S attacks b_1 in ES either. Thus, it cannot be the case that S' attacks b_1 in SF^{ES}. Let us now consider b_2; we know it is supported either by $\{b_0\}$, $\{b_0, b_1\}$ or $\{b_1\}$ in E. However, based on the previous analysis, none of these sets can be attacked by S', and thus b'_2 cannot be in S' due to admissibility. This, in addition to the fact that no subset of S attacks b_2, means that S' cannot attack b_2 in SF^{ES}. We can repeat this analysis till we reach $b_n = b$ and conclude that there was no set in S' attacking T, and thus no set in S' attacking the set $T' \subseteq T$ that is responsible for carrying out the conflict in R'. Consequently, a could not have been defended by S' in the first place and we reach a contradiction. Thus, S is admissible in ES.

Let $S \subseteq A$ be complete in ES. By the previous parts of this proof we know that $S' = S \cup np(S)$ is admissible in SF^{ES}. If it is not complete, it means there exists an argument $a \in A' \setminus S'$ defended by S'. We will now distinguish between cases where a is a primed argument and where it is a standard one.

Let us first assume that a is primed version of an argument $b \in A$ and is defended by S' in SF^{ES}. However, we can observe that based on the construction on S', it must be the case that not every evidential sequence for b is attacked on a non–b element. Moreover, at least one sequence for b on A has to exist; otherwise, a would be in $np(S)$. Thus, let $(b_0, ..., b_n)$ be this unattacked evidential sequence on A for b. By the construction of SF^{ES}, it holds that $\{b_0, ..., b_{n-1}\}$ attacks a. We can observe that for any argument b_i in the sequence, $(b_0, ..., b_i)$ is an evidential sequence for this argument. Due to the fact that $b_0 = \eta$, it cannot be the case that either S or S' attack b_0. Since $\{b_0\}R'b'_1$, then due to the admissibility of S', $b'_1 \notin S'$. Moreover, as no subset of S attacks b_1, then we can conclude that S' cannot attack b_1. Based on the definition of the evidential sequence, it has to be the case that either $\{b_0\}$, $\{b_1\}$ or $\{b_0, b_1\}$ support b_2 in E. Hence, they also are the attackers of b'_2 in R', and as S' does not attack neither b_0 nor b_1, it cannot contain b'_2. Since no subset of S attacks b_2, we can thus show that S' cannot attack b_2 either. We can continue this analysis until we reach b_{n-1} and the conclusion that if S did not attack any b_i, then neither did S'. Consequently, S' could not have defended a and we reach a contradiction.

Let us thus assume that a is an argument in A and that it is defended by S' in SF^{ES}, but is not acceptable w.r.t. S in ES. If a is not acceptable w.r.t. S, then either S does not e–support a or there exists a minimal e–supported attack T on a not attacked by S. If it is the first case, then based on the fact that S is self–supporting, we can show that there is no evidential sequence for a s.t. the elements of this sequence preceding a are in S. Moreover, clearly $a \neq \eta$, as η has to be in S already. This means that S' cannot defend a against a' and we reach a contradiction. If it is the second case, then we can repeat previously performed analysis in order to show that if S does not attack any argument in T, then S' cannot attack any subset $T' \subseteq T$ carrying out the actual conflict. Consequently, a cannot be defended by S' in SF^{ES}, and we reach a contradiction. We can finally conclude that S' is complete in SF^{ES}.

Let $S' \subseteq A'$ be a complete extension of SF^{ES}. From the previous parts of this proof we know that $S = S' \cap A$ is admissible in ES. What remains to be shown is that there is no argument $a \in A \setminus S$ that is acceptable w.r.t. S in ES. Assume it is not the case. We can observe that if a is acceptable w.r.t. S, then S e–supports a. Thus, there is an evidential sequence for a s.t. elements of the sequence preceding a are contained in S. Hence, they are also contained in S', and based on the construction of SF^{ES}, it holds that S' attacks the auxiliary argument a' for a. Consequently, based on the fact that S' does not defend a, there must be an attack on a that is carried out by a set of standard arguments only. Let us thus assume that there is a set $B \subseteq A$ s.t. $BR'a$ and S' does not attack B. Based on the relation between R and R', it holds that BRa as well. Due to the fact that a is acceptable w.r.t. S in ES, either there is no e–supported attack T s.t. $B \subseteq T$ or every such T is attacked by S. If it is the first case, then it has to be the case that there exists $b \in B$ that does not possess an evidential sequence on A. Consequently, $b \neq \eta$, and there exists a primed argument $b' \in A'$ s.t. $\{b'\}R'b$. We can observe that b' will be an initial argument in SF^{ES}. Thus, by completeness of S', $b' \in S'$, and S' defends a from B. If it is the second case, then we can show that for at least one argument $b \in B$, S attacks every evidential sequence for b. If it were not the case, then we could use an unattacked sequence for every argument in B in order to construct an e–supported attack which S cannot attack. If S attacks b directly, then so does S'. If it attacks a non–b member in every sequence, then it is easy to see that S' defends (and thus contains) b'. This again means that S' can attack B. Hence, in all of the cases we can show that S' defends a and we reach a contradiction. We can finally conclude that S is complete in ES.

Let us now assume that the relation between the complete extensions in ES and SF^{ES} is not one–to–one. Based on the previous proofs this means that there exist two complete extensions $S', S'' \subseteq A'$ of SF^{ES} s.t. $S' \cap A = S'' \cap A$. This means there is a primed argument $a' \in A'$ s.t. either $a' \in S' \setminus S''$ or $a' \in S'' \setminus S'$. We can observe that if $a' \in S'$, then every set of arguments $\{a_0, ..., a_{n-1}\}$ s.t. $(a_0, ..., a_{n-1}, a)$ is an evidential sequence for a is attacked by S'. By using previous parts of this proof we can show this cannot be done without attacks from standard arguments. However, since $S' \cap A = S'' \cap A$, then S'' also can attack every such set $\{a_0, ..., a_{n-1}\}$. Thus, S'' defends a' and $a' \in S''$. We reach a contradiction with our assumptions and therefore $S' \subseteq S''$. We can how that $S'' \subseteq S'$ in a similar fashion. Consequently, $S' = S''$ and the relation between the complete extensions of ES and SF^{ES} is one–to–one.

By using Theorems 2.24 and 2.112 and the analysis above we can easily prove the relation between preferred and grounded extensions of these frameworks.

Let us focus on the stable semantics. Assume that $S \subseteq A$ is stable in ES. Based on Theorem 2.112 and the previous analysis, $S' = S \cup np(S)$ is complete (and thus conflict–free) in SF^{ES}. Now, since every argument $e \notin S$ has every evidential sequence attacked by S, then it is easy to see that every such e is attacked by S' as well, be it by the use of standard or primed arguments. What remains to be shown is that every auxiliary argument not in S' is also attacked by S'. Let a' be an arbitrary auxiliary argument outside of the extension. If a is in S', then by the fact that S' is complete it has to be the case that S' attacks a'. If a is not in S', then by the correspondence with S it means that every evidential sequence of a is attacked by S. If these attacks occur on non–a members, then a' must be in S'. Thus, we are left with a case were all the sequences are attacked on a only. However, by the completeness of S this means that S must have included the members of these sequences (excluding a). Thus, S' had sufficient means to attack a' again. Consequently, S' is stable in SF^{ES}.

Let us now focus on the other way around and show that if $S' \subseteq A'$ is stable in SF^{ES}, then so is $S = S' \cap A$ in ES. By Theorems 2.23 and 2.24 and the previous analysis, we know that both S' and S are complete. Let us now assume that S is not stable, i.e. there is an argument $a \in A \setminus S$ which has an evidential sequence unattacked by S. We can use previously done analysis to show that no subset of standard arguments of S' can attack a and that a' cannot be in S' (we would not be able to defend it). Thus, we reach a contradiction with the stability of S' and can conclude that S is stable in ES. □

Theorem 11.7. *Let $ES = (A, R, E)$ be a strongly valid EAS and SF^{ES} its corresponding defender SETAF obtained by Translation 76. By $S_{np} = \{a' \mid$ for every self–supporting set C for a, $\exists S' \subseteq S, c \in C \setminus \{a\}$ s.t. $(S', c) \in R\}$ we will denote the primed arguments corresponding to a subset of S^+ in which every self–supporting set for an argument a is attacked by S on an argument different from a.*

If a set $S \subseteq A$ is conflict–free in ES, then it is conflict–free in SF^{ES}. The set $S \cup S_{np}$ is not necessarily conflict–free in SF^{ES}. If a set $S \subseteq A$ is strongly self–supporting in ES, then $S \cup S_{np}$ is conflict–free in SF^{ES}. If S is a σ–extension of ES, where $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$, then $S \cup S_{np}$ is a σ–extension of SF^{ES}.

If a set $S' \subseteq A'$ is a σ'–extension of SF^{ES}, where $\sigma' \in \{$conflict–free, admissible, preferred, complete, grounded, stable$\}$, then $S = S' \cap A$ is a σ'–extension of ES. If S' is conflict–free in SF^{ES}, then $S = S' \cap A$ might not be strongly self–supporting in ES.

Proof. Let $S \subseteq A$ be a conflict–free extension of ES. We can observe that all attacks in R' that occur only between the arguments in A correspond precisely to R. Therefore, S is easily seen to be conflict–free in SF^{ES}. However, $S' = S \cup S_{np}$ is not necessarily conflict–free in SF^{ES}. Take for example the framework $(\{a, b, c, \eta\}, \{(a, b)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{b\}, c)\})$. The set $\{a, c\}$ is conflict–free in it, despite the fact that a attacks the only supporter b of c. Thus, $\{a, c\}_{np} = \{c'\}$, and we obtain the set $\{a, c, c'\}$ on the SETAF side. Clearly, a primed argument cannot appear in a conflict–free extension along with its original version.

Let $S \subseteq A$ be a strongly self–supporting set of ES. Let us assume that $S' = S \cup np(S)$ is not conflict–free. Since S is itself conflict–free in SF^{ES}, this means that for some argument $a' \in np(S)$, either $a \in S$ or there exists a set of standard arguments $X \subseteq S$ s.t. $XR'a'$. However, a' is added to $np(S)$ if and only if S attacks all evidential sequences of a on non–a elements. If it were the case that $a \in S$, then S would contain an evidential sequence for a, and thus a set $T \subseteq S$ s.t. TEa. However, since $a' \in np(S)$, it means we breach the conflict–freeness of S in ES. Let us focus on the case that there exists $X \subseteq S$ s.t. $XR'a'$. Due to the fact that S is a self–supporting set, every $x \in X$ contains an evidential sequence on S. We can recombine them into a single sequence. From the construction of SF^{ES} it holds that XEa; thus, we can extend the sequence with a in order to provide an evidential sequence for a s.t. every member of the sequence preceding a is in S. However, by the construction of $np(S)$, it means that S would have to attack itself and we reach a contradiction. Thus, S' is conflict–free in SF^{ES}.

Let $S' \subseteq A'$ be a conflict–free extension of SF^{ES}. Since every attack from R is contained in R', it follows easily that $S = S' \cap A$ is a conflict–free set of ES. However, not every conflict–free extension of SF^{ES} has to correspond to a strongly self–supporting set in ES. We can consider a trivial framework $(\{\eta, a\}, \emptyset, \{(\{\eta\}, a)\})$; the associated SETAF is $(\{\eta, a, a'\}, \{(\{\eta\}, a'), (\{a'\}, a)\})$. We can observe that $\{a\}$ is a conflict–free extension of our target framework. However, due to the absence of η, it is neither self–supporting nor strongly–self supporting in our source EAS.

Let $S \subseteq A$ be an admissible extension of ES. From the previous parts of this proof we know that $S' = S \cup np(S)$ is conflict–free in SF^{ES}. What remains to be shown is that S' defends all of its members in SF^{ES}. Let a be an argument in S. We can naturally exclude η from this analysis. Based on the construction of SF^{ES}, we can observe that $a' \in A'$ and $\{a'\}R'a$. From the fact that S is self–supporting in ES, it follows that there exists $X \subseteq S$ s.t. XEa and thus $X \subseteq S'$ s.t. $XR'a'$. Therefore, S' defends the arguments it contains from attacks carried out by primed arguments.

What we also need to show is that a is defended by S' from attacks carried out by arguments in S. Since a is acceptable w.r.t. S in ES, then every minimal e–supported attack T on a is attacked by S. Let $T' \subseteq T$ be a subset of T s.t. $T'Ra$ (and thus $T'R'a$). There exists an argument $t \in T'$ s.t. all of its minimal evidential sequences are attacked – if it were not the case, then every t would have an unattacked sequence which we could combine into a single unattacked e–supported attack on a. Thus, all evidential sequences for t are attacked; if it is by an attack on t itself, then obviously S' attacks t as well and a is defended against T. If it is not through t, then due to the presented construction, $np(S)$ contains t' and again a is defended by S'. We can observe that since ES is strongly valid, there exists no $T \subseteq A$ s.t. $TR'a$ in SF^{ES} that would not be a part of an e–supported attack on a in ES.

Finally, we need to analyze the status of arguments in $np(S)$. Let $a = b'$ be a primed argument for $b \in A$; assume it is not defended by S'. Consequently, there exists a set of arguments $X \subseteq A$ s.t. $XR'b'$ and at the same time there is no $x \in X'$ and no set of arguments $T' \subseteq S'$ s.t. $T'R'x$. By the construction of SF^{ES}, the set T' would either correspond to a set of arguments attacking x in R in ES or be the primed version of x (if it exists). Therefore, S cannot directly attack x, and due to the absence of x' in S', it cannot attack all evidential sequences for x on non–x members. Hence, we can show that it cannot be the case that S attacks every self–supporting set for x in ES. As it holds for every $x \in X$, we can collect these unattacked self–supporting sets into one single set $C \subseteq A$. By using Theorem 4.37, it can be observed that $C \cup \{b\}$ would be a self–supporting set for b and no argument in C is attacked by E in FN. Hence, we reach a contradiction with the construction of S' and b' should not have been in the set in the first place. We can therefore conclude that S' is admissible in SF^{ES}.

Let $S' \subseteq A'$ be an admissible extension of SF^{ES}. By the previous parts of this proof we know that the set $S = S' \cap A$ is conflict–free in ES. However, let us assume that S is not self–supporting, i.e. there exists n argument $a \in S$ that does not have an evidential sequence on S. We can observe that if $a = \eta$, then it has a trivial evidential sequence (η) on any set containing it. Thus, let us assume $a \neq \eta$. By Theorem 4.37, this means there is an argument $a \in S$ for which there is no $T \subseteq S$ s.t. TEa. However, by the construction of SF^{ES}, this means that S' cannot defend a against a'. This contradicts the admissibility of S' in SF^{ES}. Thus, for every argument $a \in S$, there exists a subset $T \subseteq S$ s.t. TEa, and by Theorem 4.37, S is self–supporting in ES.

Let us now assume that S is not admissible in ES. Based on the previous analysis, this means that there exists an argument $a \in S$ and a minimal e–supported attack T on a s.t. no element of T is attacked by S. Since T is a minimal e–supported attack, it is also a self–supporting set by Lemma 2.108. Thus, it can be represented as an evidential sequence; let $(b_0, .., b_n)$ be this sequence. Since $b_0 = \eta$, it cannot be attacked in any of the frameworks. Let us focus on b_1; we know that there exists an argument b'_1 in A' attacking b_1. As $\{b_0\}R'b'_1$ and there is no set in S (and no set in S') attacking b_0, b'_1 cannot be in S' due to admissibility. Moreover, no set in S attacks b_1 in ES either. Thus, it cannot be the case that S' attacks b_1 in SF^{ES}. Let us now consider b_2; we know it is supported either by $\{b_0\}$, $\{b_0, b_1\}$ or $\{b_1\}$ in E. However, based on the previous analysis, none of these sets can be attacked by S', and thus b'_2 cannot be in S' due to admissibility. This, in addition to the fact that no subset of S attacks b_2, means that S' cannot attack b_2 in SF^{ES}. We can repeat this analysis till we reach $b_n = b$ and conclude that there was no set in S' attacking T, and thus no set in S' attacking the set $T' \subseteq T$ that is responsible for carrying out the conflict in R'. Consequently, a could not have been defended by S' in the first place and we reach a contradiction. Thus, S is admissible in ES.

Let $S \subseteq A$ be complete in ES. By the previous parts of this proof we know that $S' = S \cup np(S)$ is admissible in SF^{ES}. If it is not complete, it means there exists an argument $a \in A' \setminus S'$ defended by S'. We will now distinguish between cases where a is a primed argument and where it is a standard one.

Let us first assume that a is the primed version of an argument $b \in A$ and is defended by S' in SF^{ES}. However, we can observe that based on the construction on S', it must be the case that not every evidential sequence for b is attacked on a non–b element. Due to the fact that we are dealing with a strongly valid framework, b must possess at least one sequence on A. We can observe that since S is self–supporting in ES, it is the case that $\eta \in S$ and thus $b \neq \eta$. Let now $(b_0, ..., b_n)$ be the unattacked evidential sequence on A for b. We can observe that there exists a nonempty set $T \subseteq \{b_0, ..., b_{n-1}\}$ s.t. TEb in ES and thus $TR'a$ in SF^{ES}. Let us start with b_0. By the requirements of the evidential sequences, $b_0 = \eta$, and is thus an initial argument both in ES and SF^{ES}. Consequently, it cannot be attacked neither by S nor S'. Let us thus focus on b_1. We can observe that no subset of S attacks b_1; therefore, if S' attacks b_1, it can only be through the primed argument b'_1. However, by the properties of the evidential sequence, $\{b_0\}Eb_1$ and thus $\{b_0\}R'b'_1$. Since S' does not attack b_0, it cannot contain b'_1 due to admissibility. Consequently, S' does not attack b_1. Let us now analyze b_2. Based on the properties of the evidential sequence, either $\{b_0\}Eb_2$, or $\{b_1\}Eb_2$, or $\{b_0, b_1\}Eb_2$. All of these supporting sets are possible attackers of b'_2, and due to the fact that S' cannot attack any of them, $b'_2 \notin S'$. Since S does not contain any subset that would attack b_2 either, we can finally conclude that S' does not attack b_2. We can continue this line of reasoning until we reach b_{n-1} and the conclusion that E' could not have defended $a = b'$. We reach a contradiction. Therefore, if S' defends a primed argument in SF^{ES}, then this argument is already in S'.

Let us thus assume that a is an argument in A and that it is defended by S' in SF^{ES}, but is not acceptable w.r.t. S in ES. If a is not acceptable w.r.t. S, then either S does not e–support a or there exists a minimal e–supported attack T on a not attacked by S. If it is the first case, then based on the fact that S is self–supporting, this means that there is no $X \subseteq S$ s.t. XEa. Moreover, clearly $a \neq \eta$, as η has to be in S already. Therefore, we can show that S' cannot defend a against a' and we reach a contradiction. If it is the second case, then we can repeat previously performed analysis in order to show that if S does not attack any argument in T, then S' cannot attack any subset $T' \subseteq T$ carrying out the actual conflict. Consequently, a cannot be defended by S' in SF^{ES}, and we reach a contradiction. We can finally conclude that S' is complete in SF^{ES}.

Let $S' \subseteq A'$ be a complete extension of SF^{ES}. From the previous parts of this proof we know that $S = S' \cap A$ is an admissible extension of ES. What remains to be shown is that there is no argument $a \in A \setminus S$ that is acceptable w.r.t. S in ES. Assume it is not the case. We can observe that if a is acceptable w.r.t. S, then S e–supports a. Thus, there is a subset $X \subseteq S$ s.t. XEa, and based on the construction of SF^{ES}, it holds that S' attacks the auxiliary argument a' for a. Consequently, based on the fact that S' does not defend a, there must be an attack on a that is carried out by a set of standard arguments only.

Let us thus assume that there is a set $B \subseteq A$ s.t. $BR'a$ and S' does not attack B. Based on the relation between R and R', it holds that BRa as well and no argument in B is directly attacked by S either. Due to the fact that ES is strongly valid, there will be an e–supported attack T on a in ES s.t. $B \subseteq T$. Now, since S' does not attack B, then it cannot contain any primed argument b' of any $b \in B$. This also means that S' does not defend any of the primed arguments; consequently for every b', there is a set $C \subseteq A$ s.t. $CR'b'$ and S' (and thus also S) does not attack any of the arguments in C (it might even be the case that $C \subseteq S'$). Again, the primed arguments of such a set cannot be in S'. We can continue this analysis and gathering the sets of arguments that are attacking primed arguments which are not attacked by S' until we cannot add any further elements. Since these sets correspond to support sets from E and we are dealing with a strongly valid framework, at some point all of the required arguments will be gathered. Based on the construction of SF^{ES} and Theorem 4.37, it can be shown that the union of all of the gathered sets will form a self–supporting set containing all elements in B. Thus, this set will be a self–supported attack in ES against a and S will not attack any of its elements. This means that a cannot be acceptable w.r.t. S in ES and we reach a contradiction. Hence, S is a complete extension of ES.

Let us now assume that the relation between the complete extensions in ES and SF^{ES} is not one–to–one. Based on the previous proofs this means that there exist two complete extensions $S', S'' \subseteq A'$ of SF^{ES} s.t. $S' \cap A = S'' \cap A$. This means there is a primed argument $a' \in A'$ s.t. either $a' \in S' \setminus S''$ or $a' \in S'' \setminus S'$. By construction, a' is attacked by those subsets of A that support a through E. Thus, there exists such a set $B \subseteq A$ s.t. $BR'a'$ (equivalently, BEa) and S' attacks an argument B, but S'' does not. Due to the fact that we are dealing with strongly valid EAS, we can observe that every such set B is a part of an evidential sequence for a. We can also note that the subset of S' attacking an argument in B cannot consist of standard arguments (i.e. the attack cannot originate from R), otherwise both sets S' and S'' would be capable of it. Hence, there is another primed argument corresponding to the argument in B attacked by E'. We can can find a set B' attacking this primed argument which in turn is attacked by E', but not E''. By using previous observations, we can show that both B and B' are a part of a powerful sequence for a. We can continue this analysis until we reach a set of arguments in which no argument possesses a primed attacker, and as we are dealing with a strongly valid framework, this is bound to happen. Therefore, if S' attacks this argument, it is only by using attacks from standard arguments. However, if this is the case, then S'' attacks this argument as well, and we reach a contradiction. Therefore, $S' \subseteq S''$. In a similar manner we can show that $S'' \subseteq S'$. Hence, there is a one–to–one relation between the complete extensions of ES and SF^{ES}.

By using Theorems 2.24 and 2.112 and the analysis above we can easily prove the relation between preferred and grounded extensions of these frameworks.

Let us focus on the stable semantics. Assume that $S \subseteq A$ is stable in ES. Based on Theorem 2.112 and the previous analysis, $S' = S \cup np(S)$ is complete (and thus conflict–free) in SF^{ES}. Now, since every argument $e \notin S$ has every evidential sequence attacked by S, then it is easy to see that every such e is attacked by S' as well, be it by the use of standard or primed arguments. What remains to be shown is that every auxiliary argument not in S' is also attacked by S'. Let a' be an arbitrary auxiliary argument outside of the extension. If a is in S', then by the fact that S' is complete it has to be the case that S' attacks a'. If a is not in S', then by the correspondence with S it means that every evidential sequence of a is attacked by S. If these attacks occur on non–a members, then a' must be in S'. Thus, we are left with a case were all the sequences are attacked on a only. However, by the completeness of S this means that S must have included the members of these sequences (excluding a). Thus, S' had sufficient means to attack a' again. Consequently, S' is stable in SF^{ES}.

Let us now focus on the other way around and show that if $S' \subseteq A'$ is stable in SF^{ES}, then so is $S = S' \cap A$ in ES. By Theorems 2.23 and 2.24 and the previous analysis, we know that both S' and S are complete. Let us now assume that S is not stable, i.e. there is an argument $a \in A \setminus S$ which has an evidential sequence unattacked by S. We can use previously done analysis to show that no subset of standard arguments of S' can attack a and that a' cannot be in S' (we would not be able to defend it). Thus, we reach a contradiction with the stability of S' and can conclude that S is stable in ES. $\square$

Theorem 11.8. *Let $ES = (A, R, E)$ be a support singular, attack binary and strongly valid EAS and $BF^{ES} = (A, R', S)$ its associated BAF created with Translation 77. Then BF^{ES} is support acyclic and $\eta \in A$ is the only argument s.t. $\nexists a \in A, aS\eta$.*

Proof. Let $E' = \{(a, b) \mid \exists X \subseteq A \text{ s.t. } a \in X \text{ and } (X, b) \in E\}$ be the binary version of E. Due to the fact that ES is support singular, we can that here exists only one support assignment function for ES from Definition 4.35. Since ES is strongly valid, A can be ordered into an evidential sequence w.r.t. this function. We can observe that this ordering will define a topological ordering on A w.r.t. E'. Thus, (A, E') has to be a directed acyclic graph, and therefore the support subgraph of BF^{ES} is directed acyclic. $\square$

Theorem 11.10. *Let $ES = (A, R, E)$ be an attack binary EAS and $FN^{ES} = (A, R', N)$ its associated AFN obtained through Translation 78. If FN^{ES} is strongly consistent, then so is ES. However, it is not the case that if ES is strongly consistent, then so is FN^{ES}. If ES is all–supported and strongly consistent, then FN^{ES} is strongly consistent. FN^{ES} might not be in minimal form, even if ES is. If ES is weakly (strongly) valid, then so is FN^{ES}. If ES is weakly and relation valid, then FN^{ES} is relation valid.*

Proof. Let us focus on strong consistency. Let FN^{ES} be strongly consistent and let $N(a) = \{b \mid \exists B \subseteq A, b \in B \text{ s.t. } BNa\}$ and $R'(a) = \{b \mid bRa\}$ be the sets of arguments supporting and attacking an argument $a \in A$ in FN. In a similar fashion, we can introduce the sets $E(a) = \{b \mid \exists B \subseteq A, b \in B \text{ s.t. } BEa\}$ and $R(a) = \{b \mid \exists B \subseteq A, b \in B \text{ s.t. } BRa\}$ of arguments supporting and attacking $a \in A$ in FN^{ES}. It is easy to see that if $a = \eta$, then $N(a) = E(a) = R(a) = R'(a) = \emptyset$. If $E(a) = \emptyset$ and $a \neq \eta$, then $N(a) = \{a\}$; if $E(a) \neq \emptyset$ and $a \neq \eta$, then $E(a) = N(a)$. Finally, due to the fact that ES is attack binary, $R'(a) = R(a)$. Consequently, in all cases $E(a) \subseteq N(a)$ and $R'(a) = R(a)$. This means that if $N(a) \cap R'(a) = \emptyset$, then $E(a) \cap R(a) = \emptyset$. This shows that if FN^{ES} is strongly consistent, then so is ES.

Let us now assume a simple strongly consistent EAS $(\{\eta, a\}, \{(\{a\}, a)\}, \emptyset)$. We can observe that a is not supported by evidence. Its corresponding AFN is $(\{\eta, a\}, \{(a, a)\}, \{(\{a\}, a)\})$, which is neither consistent nor strongly consistent.

Let us now assume that FN^{ES} is a strongly consistent and an all–supported framework. This means that every non–η argument in the framework is supported by a subset of arguments. Thus, $\{a \mid \exists X \subseteq A \text{ s.t. } a \in X, XEa\} = \{a \mid \exists X \subseteq A \text{ s.t. } a \in X, XNa\}$. Since the collection of arguments connected through the attack relation is also the same in ES as in FN^{ES}, strong consistency of FN^{ES} follows.

Let us assume an evidential framework in which argument a is supported by the set $\{a, b\}$ in E. The necessary support sets created for a will be $\{a\}$, $\{b\}$ and $\{a, b\}$. Thus, the new collection of support sets is comparable, even though the old one was not. Therefore, the produced AFN does not have to be in minimal form, even if the original EAS was.

By using the correspondence between the powerful and evidential sequences (see Theorem 11.11) it is easy to show that if ES is weakly valid, then so is FN^{ES}. If ES is weakly valid, then we can observe that $E(a) = N(a)$ for any argument $a \in A$. Consequently, if for every supporting set of a in ES we can find an evidential sequence for a s.t. the elements of this set precede a, then we can do the same with powerful sequences in FN^{ES}. Hence, if ES is weakly and relation valid, then FN^{ES} is relation valid. Strong validity can be easily shown based on the similarity between the support functions from Definitions 4.30 and 4.35 and the changes the E relation undergoes in order to become N. $\square$

Theorem 11.11. *Let $ES = (A, R, E)$ be an attack binary EAS and $FN^{ES} = (A, R', N)$ its associated AFN obtained through Translation 78. $(a_0, ..., a_n)$ is an evidential sequence for a on $S \subseteq A$ in ES iff $(a_0, ..., a_n)$ is a powerful sequence for a on S in FN^{ES}. S is self–supporting in ES iff it is coherent FN^{ES}.*

Proof. Let $(\eta, a_0, ..., a_n)$, where $a_n = a$, be an evidential sequence for a in ES on $S \subseteq A$. Then it is also a powerful sequence for a on S in FN^{ES}. Since by the definition of an evidential framework, there is no $T \subseteq A$ s.t. $TE\eta$ and Translation 78 adds no additional support relation to η, it holds that there is no $T \subseteq A$ s.t. $TN\eta$ in FN^{ES}. Therefore, the first two requirements of the powerful sequence are satisfied. By the evidential sequence, we know that $\{\eta\}Ea_0$. Hence, by Translation 78, every set $Z \subseteq A$ s.t. ZNa_0 contains η, and thus a_0 satisfies the powerful requirement. Let a_i, where $1 \leq i \leq n$, be an element of the evidential sequence. We know that for every such a_i, there exists a nonempty $T \subseteq \{\eta, a_0, .., a_{i-1}\}$s.t. TEa_i. By Translation 78, for every $Z \subseteq A$ s.t. ZNa_0, $T \cap Z \neq \emptyset$, it holds that for every Z, $Z \cap \{\eta, a_0, .., a_{i-1}\} \neq \emptyset$. Thus the powerful requirements are satisfied.

Let now $(a_0, ..., a_n)$, where $a_n = a$, be a powerful sequence for a on $S \subseteq A$ in FN^{ES}. Then it is also an evidential sequence for a on S in ES. By Translation 78, it is easy to see that η is the only argument that requires no support in FN^{ES} and thus it is the only candidate for a_0. Thus, the first two requirements of an evidential sequence are satisfied. Let a_i be an arbitrary, nonzero argument. This means that for every $Z \subseteq A$ s.t. ZNa_i, $Z \cap \{a_0, ..., a_{i-1}\} \neq \emptyset$. Please note that Translation 78 guarantees the existence of at least one supporting set Z. Let us assume that a_i does not satisfy the evidential requirements, i.e. $\forall T \subseteq A$ s.t. TEa_i, $T \nsubseteq \{a_0, ..., a_{i-1}\}$. This means that for every such T, there is some argument $t \in T$ s.t. $t \notin \{a_0, ..., a_{i-1}\}$. However, by Translation 78, from such $t's$ we can construct a set Z s.t. ZNa_i. For this set it holds that $Z \cap \{a_0, ..., a_{i-1}\} = \emptyset$, which breaks the powerful requirement and we reach a contradiction. Therefore, the evidential conditions are satisfied and $(a_0, ..., a_n)$ is an evidential sequence for a on S in ES.

Based on these results, the correspondence between the self–supporting and coherent sets of ES and FN^{ES} is straightforward. $\square$

Theorem 11.12. *Let $ES = (A, R, E)$ be an attack binary EAS and $FN^{ES} = (A, R', N)$ its associated AFN obtained through Translation 78. Let $S \subseteq A$ be a self–supporting (coherent) set in ES (FN^{ES}). An argument $a \in A$ is acceptable w.r.t. S in ES iff it is defended by S in FN^{ES}.*

Proof. Let a be acceptable w.r.t. a self–supporting set $S \subseteq A$ in ES. We can observe that as S is self–supporting and e–supports a, $S \cup \{a\}$ is also self–supporting in ES. Thus, by Theorem 11.11, $S \cup \{a\}$ is coherent in FN^{ES}. Let us focus on the conflict part of acceptability. Given any set $C \subseteq A$ that carries out a minimal e–supported attack on a, S support attacks a member of C. Since S is self–supporting, any attack carried out by it will be e–supported. Please note that although technically we attack only minimal e–supported attacks on a, it is easy to see that it cannot be the case that there exists an unattacked e–supported attack on a. Every such attack either contains a minimal one, or is one – either case, it still remains attacked. By Lemma 2.108, C is self–supporting, and thus coherent in FN^{ES}. Since C attacks a and ES in an attack binary EAS, $\exists c \in C$ s.t. $\{c\}Ra$. By Translation 78, it follows that $cR'a$ in FN^{ES}. The attack by S against C follows a similar analysis. Since for every conflict in FN^{ES} there is a corresponding one in ES, it holds that for every coherent set C containing an attacker of a, S contains an attacker of an element in C in FN^{ES}. Taking into account the fact that $S \cup \{a\}$ is coherent, we can conclude that if a is acceptable w.r.t. S in ES, then it is defended by S in FN.

Let a be defended by a coherent set $S \subseteq A$ in FN^{ES}. Since $S \cup \{a\}$ is coherent in FN^{ES}, it is also self–supporting in ES. This means that a has an evidential sequence on $S \cup \{a\}$, and by Theorem 2.99 is e–supported by S. Let now $b \in A$ be an argument s.t. $bR'a$. Since S defends a, every coherent set $C \subseteq A$ containing b is attacked by S. By Translation 78, $bR'a$ in FN^{ES} iff $\{b\}Ra$ in ES. Similarly, if S attacks C in FN^{ES}, then S attacks C in ES as well. Since every C is coherent, it is also self–supporting by Theorem 11.11. Consequently, it is an e–supported attack against a in ES and as S is assumed to be self–supporting, the attack it carries out against C is also e–supported. Therefore, S can respond to every (and thus also minimal) e–supported attack on a. Hence, a is acceptable w.r.t. S in ES. □

Theorem 11.13. *Let $ES = (A, R, E)$ be an attack binary EAS and $FN^{ES} = (A, R', N)$ its corresponding AFN obtained through Translation 78. A set $S \subseteq A$ is (strongly) self–supporting in ES iff it is (strongly) coherent in FN^{ES}. S is a σ–extension in ES, where $\sigma \in$ {conflict–free, admissible, preferred, complete, grounded, stable}, iff it is a σ–extension in FN^{ES}.*

Proof. The relation between the coherent and self–supporting sets has already been shown in Theorem 11.11. It it is easy to see by Translation 78 that a given set is conflict–free in ES iff it is conflict–free in FN^{ES}. Consequently, we can also conclude that a given set is strongly self–supporting in ES iff it is strongly coherent in FN^{ES}.

The correspondence between the admissible (preferred, complete) semantics follows straightforwardly from Lemma 2.109, Theorems 11.11 and 11.12. Since complete extensions coincide and the grounded extensions are the least w.r.t. set inclusion complete sets both in ES and FN^{ES} (see Theorems 2.112 and 2.95), the grounded extensions coincide as well.

What remains is the analysis of the stable semantics. Let $S \subseteq A$ be a stable extension of ES. Based on the correspondence between the complete extensions of two frameworks and Theorem 2.112, we can observe that S is complete in FN^{ES}. However, assume that it is not stable. This means there exists an argument $a \in A \setminus S$ that is not in the deactivated set of S. Consequently, a is not attacked by S in FN^{ES} and either requires no support at all or sufficient support is provided by S. If a requires no support, by Translation 78 it has to be the case that $a = \eta$. However, since S is self–supporting/coherent, η has to be in S and we reach a contradiction. If a is sufficiently supported by S, then based on the fact that S is coherent, it is easy to see that we can construct a powerful sequence for a on $S \cup \{a\}$. We can use the relation between the powerful and evidential sequence and Theorem 11.11 in order to show that a is e–supported by S in ES. However, as S is stable in ES, there is $s \in S$ s.t. $\{s\}Ra$ in ES or every set minimally supporting a is attacked. Consequently, either $sR'a$ or every coherent set containing a is attacked in FN. If it is the first case, then a has to be in the deactivated set and we reach a contradiction. If it is the latter, then as $S \cup \{a\}$ is coherent and S does not attack a, it has to be the case that S attacks itself. This contradicts the conflict–freeness of S. Hence, there cannot be any $a \in A \setminus S$ that is not in the deactivated set and S is stable in FN^{ES}.

Let $S \subseteq A$ be stable in FN^{ES}. Based on the previous parts of this proof, it holds that S is strongly self–supporting in ES. Let us assume it is however not stable. This means there exists an argument $a \in A \setminus S$ that is e–supported by A that is not attacked by S and has a set of arguments minimally e–supporting it that is not attacked by S either. Since a is in the deactivated set of S in FN, then either $\exists s \in S$ s.t. $sR'a$ or $\exists X \subseteq A$ s.t. XNa and $S \cap X = \emptyset$. If it is the first case, then obviously $\{s\}Ra$ in ES and we reach a contradiction. Thus, let us focus on the case where a is in the deactivated set due to lack of support. Since a is e–supported by A, it is powerful in A by Theorem 11.11. Lack of support means that for every powerful sequence, part of the sequence is not in S. Without the loss of generality, we can assume this sequence is minimal. Let $1 \leq i \leq n$ be the position of the first argument in the sequence that does not belong to S. Since $a_0 = \eta$, it cannot be the case that it does not belong to S. As all the required support for a_i is in S but $a_i \in S^+$, then it has to be the case that $\exists s \in S$ s.t. $sR'a_i$. Thus, $\{s\}Ra_i$ as well, and we reach a contradiction. This minimal powerful sequence for a in FN^{ES} gives rise to a minimal evidential sequence in ES (see proof of Theorem 11.11), from which by Theorem 2.99 we can obtain a minimal set e–supporting a. Since it is the case that for any sequence S carries out an attack in FN^{ES}, then by Translation 78 it also carries out an attack in ES and as it is a self–supporting set, the attack is e–supported. Consequently, we reach a contradiction and S is stable in ES. □

Theorem 11.14. *Let $ES = (A, R, E)$ be an EAS and $FN^{ES} = (A, R', N)$ its associated AFN obtained through Translation 79. It is not the case that if ES is strongly consistent, then so is FN^{ES}. Moreover, it is not the case that if FN^{ES} is strongly consistent, then so is ES. If ES is all–supported and strongly consistent, then FN^{ES} is strongly consistent.*

FN^{ES} might not be in minimal form, even if ES is. If ES is weakly (strongly) valid, then so is FN^{ES}. If ES is weakly and relation valid, then FN^{ES} is relation valid.

Proof. In order to show that it can happen that ES is strongly consistent but FN^{ES} is not, it suffices to repeat the example from the proof of Theorem 11.10.

We will now show that ES does not have to be strongly consistent even if FN^{ES} is. Consider a simple EAS $(\{a, b, c, \eta\}, \{(\{a, b\}, c)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{a\}, c)\})$; we can see it is not strongly consistent, as a participates both in a support for and an attack against c. The associated AFN is $(\{a, b, c, \eta, \{a, b\}\}, \{(\{a, b\}, c)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{a\}, c), (\{a\}, \{a, b\}), (\{b\}, \{a, b\})\})$ and we can observe it satisfies the consistency requirements.

Let us now assume that FN^{ES} is a strongly consistent and an all–supported framework. This means that every non–η argument in the framework is supported by a subset of arguments. Therefore, we can show that for an argument $a \in A$, $E(a) = N(a)$ (see proof of Theorem 11.10). Moreover, we can observe that $(R'(a) \cap A) \subseteq R(a)$. Additionally, every attack argument is strongly consistent, as there cannot be any argument in A' attacking it. Similarly, no attack argument is a part of any supporting set for another argument. All of this combined means that $N(a) \cap R'(a) = \emptyset$ for any argument $a \in A'$. Therefore, FN^{ES} is strongly consistent.

In order to show that FN^{ES} does not have to be in minimal form, even if ES is, we can repeat the example given in the proof of Theorem 11.10.

Please refer to the proof of Theorem 11.15 in order to see that if an argument has an evidential sequence in ES, then it has an evidential sequence in FN^{ES}. Moreover, since we are dealing with a weakly valid ES, then for every set of arguments $B \subseteq A$ s.t. there exists $a \in A, BRa$, we can find an e–supported attack T s.t. $B \subseteq T$ and T is self–supporting. Hence, every attack argument in $att(A)$ will have a powerful sequence in FN^{ES} as well. Thus, if ES is weakly valid, then so is FN^{ES}.

If ES is weakly valid, then we can observe that $E(a) = N(a)$ for any argument $a \in A$. Consequently, if for every supporting set of a in ES we can find an evidential sequence for a s.t. the elements of this set precede a, then we can do the same with powerful sequences in FN^{ES}. Moreover, for every attacking argument in $att(A)$, every supporting set in N contains one argument only; we can combine the coherent sets of these arguments into a single set not containing a given attack argument and at the same time having all of its supporters. Thus, we can show that if ES is weakly and relation valid, then FN^{ES} is relation valid.

Strong validity can be easily shown based on the similarity between the support functions from Definitions 4.30 and 4.35 and the changes the E relation undergoes in order to become N. We can observe that for every attack argument, there will exist precisely one $suf(a)$ set in the sound of Definitions 4.30. Moreover, we can always put attack arguments on the top of the powerful sequences, as they are only on the "receiving" end of support. Consequently, any EAS support function can be adapted to an AFN support function meeting the strong validity requirements. Hence, if ES is strongly valid, then so is FN^{ES}. $\square$

Theorem 11.15. *Let $ES = (A, R, E)$ be an EAS and $FN^{ES} = (A, R', N)$ its corresponding AFN obtained through Translation 79. If $S \subseteq A$ is conflict–free in ES, then both S and $S' = S \cup att(S)$ are conflict–free in FN^{ES}. If $S \subseteq A$ is (strongly) self–supporting in ES, then S and $S' = S \cup att(S)$ are (strongly) coherent in FN^{ES}. If S is a σ–extension of ES, where $\sigma \in \{$admissible, preferred, complete, grounded, stable$\}$, then $S' = S \cup att(S)$ is a σ–extension of FN^{ES}. If $S' \subseteq A'$ is a conflict–free extension of FN^{ES}, then $S = S' \cap A$ might not be conflict–free in ES. If S' is coherent in FN^{ES}, then $S = S' \cap A$ is self–supporting in ES. If S' is strongly coherent in FN^{ES}, then $S = S' \cap A$ might not be strongly self–supporting in ES. If S' is σ–extension of FN^{ES}, then $S = S' \cap A$ is a σ–extension of ES.*

Proof. Let $S \subseteq A$ be a conflict–free set of ES. Therefore, there are no $B \subseteq S$, $a \in S$ s.t. BRa. In particular, there are also no arguments $c, d \in S$ s.t. $\{c\}Rd$. Hence, there are no $c, d \in S$ s.t. $cR'd$ and S is conflict–free in FN. Additionally, there will be no arguments $B, a \in S \cup att(S)$ s.t. $BR'a$. Consequently, $S' = S \cup att(S)$ is conflict–free in FN^{ES} as well.

Let us consider an EAS $(\{a, b, c, \eta\}, \{(\{a, b\}, c)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, c)\})$ and its associated AFN $(\{a, b, c, \eta, \{a, b\}\}, \{(\{a, b\}, c)\}, \{(\{\eta\}, a), (\{\eta\}, b), (\{\eta\}, c), (\{a\}, \{a, b\}), (\{b\}, \{a, b\})\})$. We can observe that the set $\{a, b, c, \eta\}$ is conflict–free (and strongly coherent) in the target AFN. Only the set $\{a, b, c, \eta, \{a, b\}\}$ is no longer conflict–free. However, in the source EAS, $\{a, b, c, \eta\}$ is neither conflict–free nor strongly self–supporting. Thus, not every conflict–free (strongly coherent) extension of FN^{ES} is conflict–free (strongly self–supporting) in ES.

Let us now analyze the powerful and evidential sequences between the two frameworks. We can use the analysis from the proof of Theorem 11.11 in order to show that if an argument $a \in A$ has an evidential sequence on a set $S \subseteq A$, then the same sequence is a powerful sequence for a on S in FN^{ES}. Similarly, a powerful sequence for an argument $a \in A$ on $S \subseteq A$ in FN^{ES} will be an evidential sequence for this argument on S in ES. This analysis can be easily adapted to show that if $(a_0, ..., a_n)$ is an evidential sequence for an argument $a_n \in A$ on $S \subseteq A$ in ES and there exist $B \subseteq \{a_0, ..., a_n\}, a \in A$ s.t. BRa (i.e. we are dealing with an e–supported attack), then $(a_0, ..., a_n, B)$ will be a powerful sequence for $B \in A'$ on $S \cup \{B\}$ in FN^{ES}. In a similar fashion, a powerful sequence for an argument $a \in A'$ on $S' \subseteq A'$ in FN^{ES}, after removing all arguments not in A, remains an evidential sequence on $S' \cap A$ in ES for the last standard argument in the sequence. In other words, if $a \in A$, then the sequence is for a, and if $a \in att(A)$, then the sequence is for an argument b contained in a.

By using the previous analysis, we can show that if $S \subseteq A$ is self–supporting in ES, then both S and $S' = S \cup att(S)$ are coherent in FN^{ES}. We can refer to the conflict–freeness proof in order to show that the same holds for strongly self–supporting and strongly coherent sets.

From the analysis of the sequences it also follows that if $S' \subseteq A'$ is coherent in FN^{ES}, then $S = S' \cap A$ is self–supporting in ES. Nevertheless, the previously given example shows this is not the case for strongly coherent and strongly self–supporting sets.

Let $S \subseteq A$ be an admissible extension of ES and $S' = S \cup att(S)$ the associated set in FN^{ES}. From the previous parts of this proof it follows that S' is strongly coherent. Let us assume it is however not admissible; due to coherence, it can only be the case that there exist arguments $a \in S'$, $b \in A'$ s.t. $bR'a$ and a coherent set $B \subseteq A'$ s.t. $b \in B$ and no argument in B is attacked by S'. Since arguments in $att(S)$ cannot be directly attacked, we can conclude that $a \in S$. Moreover, if $bR'a$, then either bRa or $\{b\}Ra$ (depends on whether $b \in A$ or $b \in att(A)$). Due to the relation between coherent and self–supporting sets, we can observe that B will be an e–supported attack on a. Hence, if there is no $c \in S'$, $d \in B$ s.t. $cR'd$, then either there is no $c \subseteq S, d \in B$ s.t. cRd (or $c \in S, d \in B$ s.t. $\{c\}Rd$ if $c \in A$). Hence, we reach a contradiction with the admissibility of S in ES, and can conclude that $S' = S \cup att(S)$ is admissible in FN^{ES}.

Let $S' \subseteq A'$ be an admissible extension of FN^{ES} and $S = S' \cap A$ the associated set in ES. From the previous parts of this proof it follows that S is self–supporting. Let us assume it is not conflict–free; this means there exist $B \subseteq S, c \in S$ s.t. BRc. Consequently, there exist $B \in A'$, $c \in S$ s.t. $BR'c$ (or bRc if B is of the form $B = \{b\}$). We can observe that all arguments in B possess powerful sequences on A' in FN^{ES}. Since S' is admissible, it has to be the case that it attacks all powerful sequences for B (or b if $B = \{b\}$). However, due to the fact that $B \subseteq S$ and S' is coherent (i.e. attack arguments cannot appear without the standard arguments carrying them out), it holds that $B \subseteq S'$. Since arguments in $att(A)$ cannot be directly attacked, then in order to defend a against B, S' has to attack another argument in S'. We thus breach the conflict–freeness of S' in FN^{ES} and reach a contradiction. Consequently, S is conflict–free (and therefore, also strongly self–supporting) in ES.

Let us now assume that S is not admissible in ES. Since it is already self–supporting, it can only be the case that there exists a minimal e–supported attack $T \subseteq A$ against an argument $a \in S$ which is in turn not attacked by S. Let $T' \subseteq T$ be a subset of T s.t. $T'Ra$.We can observe that T and $T \cup att(T)$ are coherent sets in FN^{ES} (see Lemma 2.108 and previous parts of this proof). Consequently, T' (or t, if $T' = \{t\}$) will have a powerful sequence on $T \cup att(T)$ and it holds that $T'R'a$ (or $tR'a$ if $T' = \{t\}$). Hence, S' has to defend a against T'. Based on the previous analysis and the fact that $att(A)$ arguments cannot be directly attacked we can observe that if there is no subset of S attacking any argument in T in ES, then there cannot be any argument in S' attacking any argument in $T \cup att(T)$ in FN^{ES}. We reach a contradiction with the admissibility of S'. We can therefore conclude that S is admissible in ES.

Let $S \subseteq A$ be a complete extension of ES. By the previous parts of this proof, $S' = S \cup att(S)$ is an admissible extension of FN^{ES}. Let us assume it is not complete; this means there exists an argument $a \in A' \setminus S'$ that is defended by S'. We can observe that if the defended argument is an attack argument, then it has to be the case that S' contains every standard argument contained in the attack one. However, then this attack argument would have been already included in S' and we reach a contradiction. Let us therefore assume that a is a standard argument. If S' defends a, then $S' \cup \{a\}$ is a coherent set. This means that S' has an element in common with every set of arguments supporting a through

N. Based on the way FN^{ES} is constructed, this means that there exists $X \subseteq S$ s.t. XEa, and as S is self–supporting, it holds that S e–supports a. From the fact that S' defends a it follows that for any argument $b \in A'$ attacking a, S' attacks every coherent set $B \subseteq A'$ containing b. We can use the previous parts of this proof to show that $B \cap A$ carries out an e–supported attack on a and that if S' attacks B, then S attacks $B \cap A$. We can therefore conclude that a is acceptable w.r.t. S, which breaches the completeness of our set in ES. We reach a contradiction. Hence, $S' = S \cup att(S)$ is complete in FN^{ES}.

Let $S' \subseteq A'$ be a complete extension of FN^{ES}. Due to the fact that the defense of attack arguments boils down to coherence, we can observe that if S' contains standard arguments carrying out a group attack in R in ES, then it has to contain the attack argument related to this conflict. Now, let $S = S' \cap A$ be the admissible extension of FN^{ES} associated with S'. If it is not complete, then there exists an argument $a \in A \setminus S$ acceptable w.r.t. S. First of all, this means that S e–supports a, which due to the fact that S is self–supporting boils down to the existence of a set $X \subseteq S$ s.t. XEa. From this follows that S' has an argument in common with every set of arguments supporting a in N. As S' is coherent, it can be therefore shown that $S' \cup \{a\}$ is coherent as well. Second of all, S attacks every minimal e–supported attack $T \subseteq A$ against a. Let $T' \subseteq T$ be a set s.t. $T'Ra$. We can observe that $T \cup att(T)$ is a coherent set in FN^{ES} for T' (or t if $T' = \{t\}$) and that $T'R'a$ (or $tR'a$). Let $X \subseteq S$, $y \in T$ be the subset of S and argument in y s.t. XRy. Based on the completeness of S', it has to be the case that $X \in S'$ (or $x \in S'$ of $X = \{x\}$). This means that if S can attack T, then S' can attack both $T \cup att(T)$ and T (please note that arguments in $att(A)$ cannot be attacked in R'). Based on the relation between R and R' and the coherent and self–supporting sets in both frameworks we can also show that all arguments attacking a in R' that need to be defended from would correspond with attacks from R that a needs to be defended from as well. We can thus conclude that if a is acceptable w.r.t. S in ES, then S' defends a in FN^{ES}. This breaches the completeness of S'. We can thus conclude that S has to be complete in ES.

Let $S \subseteq A$ be a complete extension of ES. Let us assume there exists two different complete extensions $S', S'' \subseteq A'$ of FN^{ES} s.t. $S' \cap A = S'' \cap A = S$. It can only be the case that there exists an attack argument $a \in att(A)$ s.t. either $a \in S' \setminus S''$ or $a \in S'' \setminus S'$. Let us focus on the first case. Since S' is complete, it is also self–supporting. Hence, it has to be the case that the arguments participating in a are in S', i.e. not only $a \in S'$, but $a \subseteq S'$ as well. Consequently, $a \subseteq S$ and $a \subseteq S''$. Due to the fact that no argument attacks a, it has to be the case that $a \in S''$; otherwise we breach the completeness of S''. Therefore, $S' \subseteq S''$. In a similar fashion we can show that $S'' \subseteq S'$. Thus, $S' = S''$ and the relation between the complete extensions of ES and FN^{ES} is one–to–one.

We can use the relation between the complete extensions of two frameworks and Theorems 2.95 and 2.112 in order to show the correspondence between the grounded (preferred) extensions of ES and FN^{ES}.

Let $S \subseteq A$ be a stable extension of ES. This means that $S' = S \cup att(S)$ is a complete (and therefore, conflict–free) extension of FN^{ES}. Let us assume it is not stable; by Lemma 2.94 this means there exists an argument $b \in A'$ and a coherent set $B \subseteq A'$

for b s.t. no argument in B is attacked by S'. If $b \in A$ is a standard argument, then we can show it has an unattacked evidential sequence on A, which breaches the stability of S in ES. If $b \in att(A)$, then due to completeness of S' it cannot be the case that all the arguments represented by b are in S' and therefore, in S. Based on the completeness of S, this means that all the self–supporting sets for at least one argument in b are attacked by S. This would however mean that S' attacks all coherent sets for b as well; we reach a contradiction. Therefore, S' is stable in FN^{ES}.

Let $S' \subseteq A$ be a stable extension of FN^{ES} and $S = S' \cap A$ the associated complete extension of ES. If S is not complete, then there exists an argument $a \in A$ and an evidential sequence not attacked by S. Using the previous analysis we can thus show that there exists a powerful sequence (coherent set) for a in FN^{ES} and that this sequence cannot be attacked by S'. We reach a contradiction with the stability of S'. Therefore, S has to be stable in ES. □

Theorem 11.16. *Let $ES = (A, R, E)$ be a strongly consistent EAS and $D^{ES} = (A, L, C)$ its corresponding ADF obtained through Translation 80. Then D^{ES} is a BADF. If ES is all–supported, then D^{ES} is cleansed. If ES is all–supported and minimal, then D^{ES} is redundancy–free. If ES is weakly valid, then D^{ES} is weakly valid. If it is minimal, all–supported and relation valid, then D^{ES} is relation valid. If ES is strongly valid, then D^{ES} is an AADF$^+$. If it is in addition minimal, then D^{ES} is strongly valid.*

Proof. Throughout the proof we will use the following auxiliary notions. Let $E(a) = \{b \mid \exists B \subseteq A, b \in B \text{ s.t. } BEa\}$ and $R(a) = \{b \mid \exists B \subseteq A, b \in B \text{ s.t. } BRa\}$ denote the sets of arguments connected to an argument a either through support or attack. Since ES is strongly consistent, we can observe that $E(a) \cap R(a) = \emptyset$ for every argument $a \in A$.

Let us assume that D^{ES} is not a BADF. This means there exists a link $(a, b) \in L$ that is neither supporting nor attacking. Consequently, there exist a set $X \subseteq par(b)$ s.t. $C_b(X) = in$ and $C_b(X \cup \{a\}) = out$ and a set $X' \subseteq par(b)$ s.t. $C_b(X') = out$ and $C_b(X' \cup \{a\}) = in$. We can also observe that it cannot be the case that $b = \eta$, as η cannot be attacked or supported by any argument.

Let us now assume that originally, there existed a supporting set for b in E in ES, i.e. $E(b) \neq \emptyset$. Therefore, based on Translation 80, we can observe that if $C_b(X) = in$, then there exists $B \subseteq X$ s.t. BEb and there is no subset $F \subseteq X$ s.t. FRb. Thus, if $C_b(X \cup \{a\}) = out$, then it can only be the case that there now is a set $F' \subseteq X \cup \{a\}$ s.t. $F'Rb$ and $a \in F'$. By strong consistency, this means that $a \notin E(b)$. Hence, it cannot be the case that we have a set X' s.t. $C_b(X') = out$ and $C_b(X' \cup \{a\}) = in$. We reach a contradiction. Consequently, the (a, b) link has to be supporting or attacking.

Let us now assume that there was no support set for b in E in ES, i.e. $E(b) = \emptyset$. This means that a self–supporting link is added for b in D^{ES}. We now have two options; either there exists a set $X \subseteq par(b)$ s.t. $C_b(X) = in$ or not (i.e. b also attacked itself and we have created a falsum condition). If it is the first case, then we can observe that $b \in X$ and there is no subset $F \subseteq X$ s.t. FRb. Hence, if $C_b(X) = in$ and $C_b(X \cup \{a\}) = out$, then $a \neq b$ and $a \in R(b)$. Since $E(b) = \emptyset$, it cannot be the case that there is a set $X' \subseteq par(b)$

s.t. $C_b(X') = out$ and $C_b(X' \cup \{a\}) = in$. Consequently, it cannot be the case that the (a, b) link is neither attacking nor supporting. If there is no set evaluating C_b to in, then all links targeted at b will be considered both supporting and attacking (i.e. redundant). We can therefore conclude that D^{ES} is a BADF.

Let us assume that ES is all–supported; this means that for every argument $a \in A \setminus \{\eta\}$, $E(a) \neq \emptyset$. Since it is also strongly consistent, then for every argument $a \in A$, $E(a) \cap R(a) = \emptyset$. Therefore, by the construction of acceptance conditions for arguments s.t. $E(a) \neq \emptyset$, we can observe that $C_a(E(a)) = in$. Moreover, the condition of η is always satisfied. Hence, D^{ES} is cleansed.

Let us now focus on redundancy–freeness. We can observe that if ES is not all–supported, then it can happen that a condition of a given argument cannot be met. In this case, every link coming to this argument would be considered redundant in D^{ES}, independently of whether ES is itself in minimal form or not. Consequently, we need to assume that ES is both all–supported and minimal in order to ensure redundancy–freeness of D^{ES}. Let us now assume that D^{ES} is not redundancy–free. This means there exists a link $(a, b) \in L$ that is both supporting and attacking. In other words, for any set $X \subseteq par(b)$, $C_b(X) = C_b(X \cup \{a\})$ (or equivalently, $C_b(X) = C_b(X \setminus \{a\})$) for every $X \subseteq par(b)$. We can observe that $b \neq \eta$, since the evidence argument always has a condition $\top$ in D^{ES} and is never the target of an attack or support in ES.

Let now a be an argument s.t. $\exists B \subseteq A$, $a \in B$ and BEb. Due to strong consistency assumption, we can observe that $B \cap R(b) = \emptyset$ and $C_b(B) = in$. If the (a, b) link is redundant in D^{ES}, then it would have to be the case that $C_b(B \setminus \{a\}) = in$. However, this would mean that there exists $B' \subseteq B \setminus \{a\}$ s.t. $B'Eb$, which clearly violates the minimality of ES.

Let a be an argument s.t. $\exists C \subseteq A$, $a \in C$ and CRb. Let $B \subseteq A$ be a set s.t. BEb. Due to the fact that ES is all–supported, such a set is guaranteed to exist. Moreover, from the strong consistency of ES we can observe that $C \cap B = \emptyset$, and based on the construction of D^{ES}, it holds that $C_b(B) = in$ and $C_b(B \cup C) = out$. If the $(a, b) \in A$ link is redundant, then it means that $C_b(B \cup (C \setminus \{a\})) = out$. However, this means there has to exist $C' \subseteq B \cup (C \setminus \{a\})$ s.t. $C'Rb$. Moreover, as ES is strongly consistent, this means that $C' \cap B = \emptyset$ and thus $C' \subseteq C \setminus \{a\}$. This clearly violates the minimality of ES.

Since we are dealing with an all–supported EAS, the above analysis accounts for all links in D^{ES}. We can therefore conclude that if ES is minimal and all–supported, then D^{ES} is redundancy–free.

In order to show that weak if ES is weakly valid, then so is D^{ES}, we can use (the proof of) Theorem 11.17.

Let ES be all–supported, minimal and relation valid. By the previous parts of this proof, we know that D^{ES} is redundancy–free. If it is not relation valid, then it means there is an argument $a \in A$ and a minimal decisively in interpretation v_a for it s.t. no pd–function with which we can produce an acyclic pd–evaluation for a assigns v_a to a. In other words, there is no pd–acyclic subset of $A \setminus \{a\}$ s.t. $v_a^{\mathbf{t}}$ is contained in this set. Clearly, we can exclude η from this analysis, as the minimal decisively in interpretation for it is empty and

η forms its own evaluation $((\eta), \emptyset)$ easily. Based on the proof of Theorem 11.17, we can show that $v_a^{\mathbf{t}}$ corresponds to a subset of A supporting a through E, and as v_a cannot be used in an acyclic pd–evaluation, then there is no self–supporting set $B \subseteq A \setminus \{a\}$ s.t. $v_a^{\mathbf{t}} \subseteq B$. This means that ES cannot be relation valid and we reach a contradiction.

Let now ES be strongly valid. Assume D^{ES} is not an AADF$^+$. This means there exists a pd–function and a standard pd–evaluation (F, B) created with it that we cannot transform into an acyclic one. We can observe that given an argument a and the decisively in interpretation v_a assigned to it by the pd–function, then unless $a = \eta$, $v_a^{\mathbf{t}}$ is nonempty and corresponds to a set supporting a through E. Thus, from our pd–function we can derive a function meeting the construction requirements from Definition 4.35. Since we cannot order a pd–evaluation based on this pd–function into an acyclic one, it can be shown that we cannot create an evidential sequence with the associated function in ES. Thus, we reach a contradiction with strong validity of ES. We can conclude that D^{ES} has to be an AADF$^+$.

Let now ES be minimal and strongly valid. By Lemma 4.73, ES is all–supported. From the previous parts of our proof it follows that our D^{ES} is cleansed, redundancy–free, and an AADF$^+$. Consequently, by Theorem 4.43, D^{ES} is strongly valid. □

Theorem 11.17. *Let $ES = (A, R, E)$ be a strongly consistent EAS and $D^{ES} = (A, L, C)$ its corresponding ADF obtained through Translation 80. Let $S \subseteq A$ be a set of arguments. For a given evidential sequence on S for an argument $s \in S$ we can construct a corresponding acyclic pd–evaluation and vice versa. S is self–supporting in ES iff it is pd–acyclic in D^{ES}.*

Proof. Let $S \subseteq A$ be a set of arguments, $s \in S$ and $(a_0, ..., a_n)$ an evidential sequence for s on S (i.e. $\{a_0, ..., a_n\} \subseteq S$). We will show that this sequence satisfies the pd–sequence requirements.

First of all, the $a_n = s$ condition is satisfied. Since $a_0 = \eta$, it has a $\top$ acceptance condition in D^{ES} and its minimal decisively in interpretation is simply empty. This naturally satisfies the pd–sequence conditions for the starting argument. Let us now focus on the last requirement. In the evidential sequence we have that for every nonzero a_i, there exists a nonempty $B \subseteq \{a_0, ..., a_{i-1}\}$ s.t. BEa_i. Thus, our translation does not introduce additional links for a_i. We can naturally choose such B which is minimal w.r.t set inclusion out of all supporting sets contained in $\{a_0, ..., a_{i-1}\}$. Since a_i is consistent, then no subset of B attacks a_i. Therefore, $C_{a_i}(B) = in$, and we can construct a (minimal) decisively in interpretation v_i for a_i s.t. $v_i^{\mathbf{t}} = B$. Let $X = \{X_1, X_2, ..., X_m\}$ be the collection of all and only sets attacking a_i in R. We can observe that $C_{a_i}(B \cup X_i) = out$ for any of the $X_i \in X$. Moreover, due to strong consistency of ES, $B \cap X_i = \emptyset$. Consequently, in order for v_i to be decisively in for a_i, it needs to map to $\mathbf{f}$ at least one argument in every X_i and none of these elements will come from B [33]. We can easily choose a minimal $\mathbf{f}$ assignment that still satisfies this requirement. We now have a description of a possible minimal decisively

[33] In case of doubt, please consult the proof of SETAF–ADF Translation 31.

in interpretation for a_i. It is easy to see that for every (non–zero) a_i, $v_i^{\mathbf{t}} \subseteq \{a_0, ..., a_{i-1}\}$. Consequently, $((a_0, ..., a_n), \bigcup_0^n v_i^{\mathbf{f}})$ is an acyclic pd–evaluation for s on S corresponding to the evidential sequence $(a_0, ..., a_n)$. Please note that we can in fact produce more than just one evaluation. There can be several minimal sets that we can use for constructing $v_i^{\mathbf{f}}$ and even though the sequence part of an evaluation will correspond to the evidential one, the blocking sets can be different. Consequently, we obtain a number of evaluation that differ only by the blocking set.

Let $S \subseteq A$ be a set of arguments, $s \in S$ and $((a_0, ..., a_n), B)$ an acyclic pd–evaluation for s on S. We will show that the sequence part satisfies the evidential conditions. Again, the $a_n = s$ condition is easily met. Since by Translation 80 the only argument that has a minimal decisively in interpretation with an empty $\mathbf{t}$ part is η, then it has to be the case that $a_0 = \eta$ and thus another condition is met.

Now, we know that for every nonzero a_i and its minimal decisively in interpretation v_i, $v_i^{\mathbf{t}} \subseteq \{a_0, ..., a_{i-1}\}$. Since $a_i \neq \eta$, we can observe from Translation 80 that $v_i^{\mathbf{t}} \neq \emptyset$. By the construction of the arguments we know that $\exists Z \subseteq A$ s.t. ZEa_i (it is easy to see that arguments to which a self–support link was added cannot participate in acyclic evaluations). Moreover, we can find such a Z for which $Z = v_i^{\mathbf{t}}$. Consequently, $Z \subseteq \{a_0, ..., a_{i-1}\}$ and the final evidential requirement is satisfied. Therefore, the pd–sequence of the evaluation produces an evidential sequence. □

Theorem 11.18. *Let $ES = (A, R, E)$ be a strongly consistent EAS and $D^{ES} = (A, L, C)$ its corresponding ADF obtained through Translation 80. A set of arguments $S \subseteq A$ is strongly self–supporting in ES iff it is pd–acyclic conflict–free in D^{ES}.*

Proof. Let us assume that S is self–supporting conflict–free in ES, but not pd–acyclic conflict–free in D^{ES}. By Theorem 11.17 we know that S is at least pd–acyclic. What remains to be shown is that all arguments have satisfied accepted conditions and possess unblocked evaluations on S. We can observe that η always has a satisfied acceptance condition. If another argument $e \in S$ has an unsatisfied acceptance condition (i.e. $C_e(E \cap par(e)) = out$), then by Translation 80 it means that $\exists S' \subseteq S$ s.t. SRe or $\nexists Z \subseteq S$ s.t. ZEe. The first one breaches the conflict–freeness in ES, the other self–support. Let us now show that every argument $e \in S$ has an unblocked acyclic pd–evaluation on S. By Theorem 11.17, for every evidential sequence of e on S we can create at least one corresponding acyclic evaluation. As seen in the proof, we can in fact produce a number of evaluations, differing only by the choice of the blocking set. Since S is conflict–free in ES, it means that for every set X_i attacking e, there is an argument $x \in X_i$ s.t. $x \notin S$. By collecting such $x's$ for all attacking sets of all arguments in the sequence we can construct a blocking set of the evaluation for which it is clearly not the case that any of its elements is in S. In this way we obtain an unblocked acyclic evaluation for e on S. We can thus conclude that S is pd–acyclic conflict–free in D^{ES}.

Let us now assume that S is pd–acyclic conflict–free, but not self–supporting conflict–free. By Theorem 11.17, S is at least self–supporting. If S is not conflict–free in ES, it means that $\exists S' \subseteq S$ s.t. SRy. However, by Translation 80, it would mean that $C_y(S \cap$

$par(y)) = out$. Consequently, S could not have been conflict–free in D^{ES} and we reach a contradiction. □

Lemma 11.19. *Let $ES = (A, R, E)$ be a strongly consistent EAS and $D^{ES} = (A, L, C)$ its corresponding ADF obtained through Translation 80. Let $S \subseteq A$ be self–supporting conflict–free in ES and thus pd–acyclic conflict–free in D^{ES}. S attacks $a \in A$ or every set of arguments e–supporting a in ES iff the acyclic range v_S^a of S blocks every acyclic pd–evaluation of a in D^{ES}.*

Proof. It is easy to see that saying that S attacks a or every set of arguments minimally supporting it is equivalent to saying that every evidential sequence of a on A is attacked by S. By Theorem 11.17 we know that every evidential sequence has (at least one) corresponding acyclic pd–evaluation. From the proof we see that the blocking set B of every evaluation $((a_0, ..., a_n, B)$ is $B = \bigcup_{i=0}^{n} A_i$, where the A_i set for an argument a_i is a minimal set s.t. $\forall S' \subseteq A\, S'Ra_i$, $S' \cap A_i \neq \emptyset$. It is easy to see that if a given evidential sequence for a is attacked by S, then the blocking sets of all of the corresponding acyclic pd–evaluations have at least one element in common with S. Finally, we can observe that if an argument does not possess an evidential sequence in ES, then it does not have an acyclic pd–evaluation in D^{ES} and is automatically contained in the acyclic discarded set of S. Hence, if S attacks a or every set of arguments e–supporting a in ES, then the acyclic range v_S^a of S blocks all of its acyclic pd–evaluations.

Let us now assume that every acyclic pd–evaluation of a is blocked by v_S^a in D^{ES}, but there exists an evidential sequence of a unattacked by S in ES. For this sequence we can construct a number of corresponding acyclic pd–evaluations, and since the sequence is not attacked, then it cannot be the case that these evaluations are blocked via the blocking set. Consequently, it has to be the case the sequence part of these evaluations (which is exactly the evidential sequence) contains an argument falsified by v_S^a. Let $(a_0, ..., a_n)$ be the sequence part of the evaluations and a_i the first element falsified by v_S^a. Since by the construction of D^{ES}, $a_0 = \eta$ and $C_\eta = \top$, it is easy to see that $i \neq 0$. Let us move to $i = 1$ and let v_1 be the decisively in interpretation of a_1 that was used in constructing the sequence. We know that $v_1^{\mathbf{t}} = \{a_0\}$, $v_S^a(a_0) \neq \mathbf{f}$ and that $v_1^{\mathbf{f}} \cap S = \emptyset$. This means that a_1 could not have been decisively out w.r.t. v_S^a, and thus by Proposition 2.150 could not have been falsified by v_S^a. Hence, $i \neq 1$. We can proceed with analyzing $i = 2$ and again see that v_2 could not have been falsified by v_S^a. We can continue in the same manner until we reach a (i.e. $i = n$) and conclude that it could have not had all of its acyclic pd–evaluations blocked. We reach a contradiction. Hence if every acyclic pd–evaluation of a was blocked, then every evidential sequence of a had to be attacked. □

Lemma 11.20. *Let $ES = (A, R, E)$ be a strongly consistent EAS and $D^{ES} = (A, L, C)$ its corresponding ADF obtained through Translation 80. Let $S \subseteq A$ be self–supporting conflict–free in ES and thus pd–acyclic conflict–free in D^{ES}. An argument $a \in A$ is acceptable w.r.t. S in ES iff it is decisively in w.r.t. v_S^a in D^{ES}.*

Proof. Let us assume that a is acceptable w.r.t. S in ES, but is not decisively in w.r.t. v^a_S in D^{ES}. This means there exists at least one completion v' of the range interpretation s.t. $C_a(v'^{\mathbf{t}} \cap par(a)) = out$. Let $S' = v'^{\mathbf{t}}$. According to Translation 80, the condition of a is not satisfied if $\exists X \subseteq S'$ s.t. XRa or $\nexists Z \subseteq S'$ s.t. ZEa. If it is the first case, then no member of X was falsified in v, which means that every one of them possessed at least one unblocked acyclic pd–evaluation. Hence, by Lemma 11.19 S could not have attacked any member of X nor any set e–supporting it. Thus, a could not have been acceptable w.r.t. S in ES. Let us therefore assume that it is the case that $\nexists Z \subseteq S'$ s.t. ZEa. Since there is no such subset of S', then there cannot be any supporting subset of S either. Thus, S could not have supported, let alone e–supported, a. We reach a contradiction with our assumptions again. Hence, if an argument is acceptable w.r.t. S in ES, then it is decisively in w.r.t. v^a_S in D^{ES}.

Let us now assume that a is decisively in w.r.t. v^a_S in D^{ES}, but is not acceptable w.r.t. S in ES. This means that a is either not e–supported by S or that there exists some e–supported attack T against a which is in turn not e–support attacked by S. Since S is self–supporting, every attack carried out by it is e–supported and and S e–supports a iff it supports a through E. This means we can simplify the acceptability conditions. Let us first assume that there exists some e–supported attack T not attacked by S. By the definition of e–supported attack and Lemma 11.19 this means that no member of T has every acyclic pd–evaluation blocked. Hence, no argument in T is falsified by v^a_S. Consequently, there exists a completion v' of v^a_S s.t. $T \subseteq v'^{\mathbf{t}}$, which by Translation 80 means that $C_a(v'^{\mathbf{t}} \cap par(a)) = out$. Therefore, a could not have been decisively in w.r.t. v^a_S and we reach a contradiction. Let us thus assume that a is not supported by S. However, looking at Translation 80 brings us to a conclusion that $C_a(S \cap par(a)) = out$ and again, a could not have been decisively in. We reach a contradiction. Hence, if a is decisively in w.r.t. v^a_S in D^{ES}, then it is acceptable w.r.t. S in ES. □

Theorem 11.21. *Let $ES = (A, R, E)$ be a strongly consistent EAS and $D^{ES} = (A, L, C)$ its corresponding ADF obtained through Translation 80. A set of arguments $S \subseteq A$ is a σ–extension of ES, where $\sigma \in \{$admissible, preferred, complete$\}$ iff it is an aa–σ–extension of D^{ES}. S is stable in ES iff it is stable in D^{ES}. S is grounded in ES iff it is acyclic grounded in D^{ES}.*

Proof. Let S be an admissible extension in ES. By Theorems 11.18 and Lemma 11.20 we know that it is pd–acyclic conflict–free in D^{ES} and that all arguments in S are decisively in w.r.t. v^a_S. By the correspondence between the blocking sets and attackers shown in the proofs of the theorems and of Theorem 11.17, the members of the blocking sets are naturally falsified in the range interpretation. Consequently, all aa–admissible criterions are satisfied. The other way around also follows straightforwardly from these theorems and lemma.

We now know that the admissible extensions of ES and D^{ES} coincide. Thus, the maximal w.r.t. set inclusion admissible sets are the same, and S is preferred in ES iff it is aa–preferred in D^{ES}.

The completeness follows straightforwardly from admissibility and Lemma 11.20. We can use Theorems 2.112 and 2.158 in order to show that E is grounded in ES iff it is acyclic grounded in D^{ES}.

What remains to be shown is the correspondence of stable semantics. Let S be stable in ES. By Theorem 11.18 we know that S is at least pd–acyclic conflict–free in D^{ES}. If an argument $a \notin S$ is not e–supported by A, then it has no evidential sequence and thus no acyclic pd–evaluation by Theorem 11.17. Consequently, it will be falsified in v_S^a. If a is e–supported by A, then by the stability condition in ES and Lemma 11.19, all of its acyclic pd–evaluations are blocked by v_S^a. Hence, a is again falsified in v_S^a. By Proposition 2.150 we know that all elements falsified by v_S^a are decisively out w.r.t. v_S^a, and thus their acceptance conditions w.r.t. S are simply out. Consequently, S satisfies the model criterion and we can conclude that it is stable in D^{ES}.

Let now S be a stable extension of D^{ES}. Since by definition it is pd–acyclic conflict–free in D^{ES}, then by Theorem 11.18 it is self–supporting conflict–free in ES. From the fact that S is a model and from Lemma 2.159 it follows that $S^{a+} = A \setminus S$. Consequently, every argument $a \notin S$ is falsified in the acyclic range, which means all of its acyclic pd–evaluations are blocked. By Lemma 11.19 this means that either a or every set of arguments e–supporting a is attacked by S. Consequently, S is a stable extension of ES. $\square$

15.10 Translating ADFs: Proof Appendix

Theorem 12.1. *Let $D = (A, C)$ be an ADF and F_{AA}^D its corresponding AF obtained from Translation 82. If $S \subseteq A$ is a pd–acyclic conflict–free (aa–admissible, aa–complete, aa–preferred, stable, acyclic grounded) extension of D, then there exists a conflict–free (admissible, complete, preferred, stable, grounded) extension $S' = \{(F_1, B_1), ..., (F_n, B_n)\} \subseteq A'$ of F_{AA}^D s.t. $S = \bigcup_{i=1}^n F_i$. If $S' = \{(F_1, B_1), ..., (F_n, B_n)\} \subseteq A'$ is a conflict–free (admissible, complete, preferred, stable, grounded) extension of F_{AA}^D, then $S = \bigcup_{i=1}^n F_i$ is pd–acyclic conflict–free (aa–admissible, aa–complete, aa–preferred, stable, acyclic grounded) extension of D.*

Proof. We will use V to denote the set of all arguments that are pd–acyclic in A in D. Let (E, B) be an arbitrary argument in A' in F_{AA}^D. We can observe that by the nature of the evaluations, if $E \cap B = \emptyset$, then $\forall e \in E, C_e(par(e) \cap E) = in$ and every argument $e \in E$ has an unblocked acyclic pd–evaluation on E.

Let $S = \{(E_1, B_1), ..., (E_n, B_n)\}$ be conflict–free in F_{AA}^D and $T = \bigcup_{i=1}^n E_i$ its corresponding set of arguments in D. First of all, we can observe the positive part of the interpretation with which an argument $a \in E_i$ entered the evaluation (E_i, B_i) is obviously in T. By conflict–freeness of S, $\bigcup_{i=1}^n E_i \cap \bigcup_{i=1}^n B_i = \emptyset$. Consequently, the false part of the interpretation of a is not in T. From this follows that $\forall e \in T$, $C_e(par(e) \cap T) = in$. Consequently, T is conflict–free in D and every acyclic pd–evaluation represented by a given (E_i, B_i) is unblocked in T. We can conclude that $\bigcup_{i=1}^n E_i$ is pd–acyclic conflict–free in D.

Let now $S' \subseteq A$ be pd–acyclic conflict–free in D. Every argument in S' has an unblocked acyclic pd–evaluation on S', and without the loss of generality we can focus on the minimal ones. Thus, for every argument $s \in S'$ we can choose an argument in A' in F^D_{AA} corresponding to this evaluation, and since they are unblocked it follows that the set of the AF arguments is conflict–free in F^D_{AA}.

Let $S = \{(E_1, B_1), ..., (E_n, B_n)\}$ be conflict–free in F^D_{AA} and $T = \bigcup_{i=1}^n E_i$ the corresponding pd–acyclic conflict–free extension in D. We would like to show the relation between the discarded sets of S and T in their respective frameworks. Let $S^+ \subseteq A'$ be the set of arguments attacked by S in F^D_{AA} and $X_S = \{a \in A \mid \forall (F, V) \in A'$ s.t. $a \in F$, $(F, V) \in S^+\}$ be the set of those arguments in D s.t. all of the acyclic pd–evaluations in which they participate are in the discarded set of S^+[34]. We will show that X_S corresponds to T^{a+}. First of all, if an argument $a \in A$ is in X_S, then all of the AF arguments in A' that contain a in the pd–sequence are attacked by S. This means that for every (F, V) containing a, $\bigcup_{i=1}^n E_i \cap V \neq \emptyset$. From this follows that every acyclic pd–evaluation of a is blocked through the blocking set by the members of T. Therefore, every acyclic pd–evaluation of this argument is blocked by v^a_T and thus the argument is in T^{a+}. The fact that every argument that is in T^{a+} is also in X_S follows easily from Lemma 2.128 and the construction of AF arguments.

Let $S = \{(E_1, B_1), ..., (E_n, B_n)\}$ be admissible in F^D_{AA} and $T = \bigcup_{i=1}^n E_i$ its corresponding set of arguments in D. From the previous part of this proof we know that T is pd–acyclic conflict–free. Let us now assume that even though S is admissible in F^D_{AA}, T is not aa–admissible in D. This means that there is an argument $a \in T$ that has no acyclic pd–evaluation on T for which the blocking would be completely contained in T^{a+}. This also implies that all acyclic pd–evaluations of a have a non–empty blocking set and at least one of the arguments in every blocking set has an acyclic pd–evaluation on A (this comes from the fact that arguments not possessing at least one acyclic pd–evaluation are always in T^{a+}). Consequently, the evaluations of these argument appear as arguments in F^D_{AA} and it holds that for any argument $(F, V) \in S$ s.t. $a \in F$, there exists $c \in V$ and $(F', V') \in A'$ s.t. $c \in F'$. Moreover, by the relation between the discarded sets in both frameworks, it is easy to see that we can always find such (F', V') that is not attacked by S. However, this means that no AF argument in S that brought a to T is properly defended by S. Thus, we reach a contradiction and conclude that if S is admissible in F^D_{AA}, then T is aa–admissible in D.

Let now $S' \subseteq A$ be aa–admissible in D. This means that every argument in S' has at least one acyclic pd–evaluation on S' s.t. its blocking set is falsified by the acyclic range interpretation $v^a_{S'}$. By collecting all such evaluations of all arguments in S' and repeating the approach like in pd–acyclic conflict–free extensions of D, we can easily construct a set $V \subseteq A'$ which is at least conflict–free in F^D_{AA}. Let us assume V is not admissible in F^D_{AA}, i.e. there exists an argument $F \in A'$ s.t. F attacks an argument in V and it is not attacked by any element in V. Clearly, F is not in V^+, and by the discarded set analysis in the previous parts of this proof, none of the arguments in the pd–sequence of F can be

[34]Please observe it means that an ADF argument possessing no acyclic pd–evaluation at all is also in X_S

falsified by $v^a_{S'}$. Since the pd–sequence of F contains a member of the blocking set of an evaluation of some argument $s \in S'$ and this member is not falsified by the range, we reach a contradiction with the construction of the AF extension. Thus, V has to be admissible in F^D_{AA}.

Let $S = \{(E_1, B_1), ..., (E_n, B_n)\}$ be complete in F^D_{AA}. From the previous part of the proof we can observe that $T = \bigcup_{i=1}^n E_i$ is aa–admissible in D. Let us assume it is not aa–complete. This means there exists an argument $a \notin T$ which is decisively in w.r.t. v^a_T. We can observe that T can be represented by a single acyclic pd–evaluation containing all arguments in T in its pd–sequence and with a blocking set in T^{a+}. Thus, by using a decisively in interpretation for a of which v^a_T is a completion, we can extend this evaluation for T in order to create an evaluation for a. It has a blocking set contained in T^{a+} and will appear as an argument in F^D_{AA}. Therefore, from the analysis of the discarded sets we are able to observe that this evaluation has to be defended by S, and as it is not in S (otherwise, a would appear in T), we reach a contradiction with the completeness of S. Hence, T is aa–complete in D.

Let now $S' \subseteq A$ be aa–complete in D and $V \subseteq A'$ its corresponding set in F^D_{AA} constructed in the same manner as in the admissible case. We know that V is admissible in F^D_{AA}. Let us now assume it is not complete, i.e. there exists an evaluation argument $F \in A'$ in F^D_{AA} for some argument $a \in A$ in D that is defended by V, but not contained in V. We will analyze two cases; one where the set corresponding to $V \cup \{F\}$ in D is equal to S' and one where it is not.

Let us focus on the first case, i.e. where all arguments in the pd–sequence of F are contained in S'. By the construction of V this means that this particular evaluation for a did not satisfy the admissibility criterion, i.e. its blocking set was not completely falsified by the range. This means there is some argument $b \in A$ that is not falsified by the range and which is in the blocking set of F. Since it is not falsified, it has an acyclic pd–evaluation which is not blocked by the range interpretation. From this follows that there is an AF argument corresponding to this evaluation that is not attacked by V. Thus, F could not have been defended by V and we reach a contradiction.

Let us thus assume the latter case and that the ADF set corresponding to $V \cup \{F\}$ is not equal to S'. This means that there is at least one argument b in the pd–sequence of F which is not contained in S'. Assume that b is the first such argument in the pd–sequence and let v_b be the decisively in interpretation with which it entered F. By aa–completeness of S' this means that b is not decisively in w.r.t. $v^a_{S'}$. Moreover, due to the construction of the sequence and the fact that b is the first argument in the sequence not contained in S', it has to be the case that $v^{\mathbf{t}}_b \subseteq S'$ but $v^{\mathbf{f}}_b$ contains an argument that is not in the acyclic discarded set of S'. If this member is S' itself, then it is easy to see that V attacked F and thus could not have defended it. If this member is not in S', then we can repeat previous parts of this analysis and show that V could not have defended F again. We reach a contradiction and thus V is complete in F^D_{AA}.

We can observe that there is a one–to–one relation between the complete and aa–complete extensions of D and F^D_{AA}. This comes from the fact that if we had two different complete extension S_1 and S_2 in F^D_{AA} corresponding to the same D one, their discarded sets would have to be the same and thus defending the same arguments. Consequently, as we are dealing with complete extensions, every argument in S_1 has to be in S_2 and the vice versa, thus showing that $S_1 = S_2$.

Additionally, it also easy to see that if two aa–complete extensions T_1 and T_2 of D are s.t. $T_1 \subseteq T_2$, then their corresponding complete extensions S_1 and S_2 of F^D_{AA} are also s.t. $S_1 \subseteq S_2$. Thus, by the one–to–one relation and Theorems 2.10 and 2.158 we can show the correspondence between the preferred and aa–preferred extensions and the grounded and acyclic grounded extensions.

Finally, we can focus on the stable semantics. Let $S = \{(E_1, B_1), ..., (E_n, B_n)\}$ be stable in F^D_{AA}. By the previous parts of this proof, its corresponding set $T = \bigcup_{i=1}^{n} E_i$ is pd–acyclic conflict–free in D. What remains to be shown is that it is also a model. Assume it is not the case, i.e. there exists an argument $a \in A \setminus T$ s.t. $C_a(T \cap par(a)) = in$. Since by Proposition 2.150 every argument in the acyclic range interpretation of T is decisively out, it cannot be the case that a is falsified by the range. From this and the construction of the range it also follows that a has at least one acyclic pd–evaluation that is not blocked by v^a_T. We can thus extract a minimal evaluation for a not blocked by the range. It is easy to see that this evaluation will appear as an argument in A' in F^D_{AA} and that it will not be attacked by S. Consequently, S could not have been stable in the first place and we reach a contradiction. Thus, T is in fact a model and as it is also pd–acyclic conflict–free, it is stable in D.

Let now $S' \subseteq A$ be a stable extension of D and $V \subseteq A'$ its corresponding set in F^D_{AA} created using the admissibility construction. From the previous parts of this proof it holds that V conflict–free. What remains to be shown is that $V^+ = A' \setminus V$. Assume it is not the case and that there exists an evaluation argument $F \in A'$ for an ADF argument $a \in A$ which is not in V^+. Let first consider the case where $a \notin S'$. Since F is not in V^+, a is not in S'^{a+} as well and thus S' could not have been stable in D by Lemma 2.139. Let us now assume that $a \in S'$. By the construction of V, since $F \notin V$, then it must be the case that the blocking set of F is not fully falsified by $v^a_{S'}$. If any of the members of the set was in S', then V would have attacked F and we reach a contradiction. Thus, we have at least one argument that is neither in S' nor in S'^{a+}. However, this again by Lemma 2.139 means that S' could not have been stable in D. We reach a contradiction. Thus, if S' is stable in D, then V is stable in F^D_{AA}. □

Theorem 12.2. *Let $D = (A, C)$ be an ADF and F^D_{AC} its corresponding AF obtained from Translation 83. If $S \subseteq A$ is a pd–acyclic conflict–free (ac–admissible, ac–complete, ac–preferred, grounded) extension of D, then there exists a conflict–free (admissible, complete, preferred, grounded) extension $S' = \{(E_1, B_1), ..., (E_n, B_n)\} \subseteq A'$ of F^D_{AC} s.t. $S = \bigcup_{i=1}^{n} E_i$. If $S' = \{(E_1, B_1), ..., (E_n, B_n)\} \subseteq A'$ is a conflict–free (admissible, complete, preferred, grounded) extension of F^D_{AC}, then $S = \bigcup_{i=1}^{n} E_i$ is pd–acyclic conflict–free (ac–admissible, ac–complete, ac–preferred, grounded) extension of D.*

Proof. Please note that if an argument $(E, B) \in A'$ is not self–attacking, then it is an acyclic pd–evaluation for some argument $a \in A$. The inverse does not hold, as evaluations can be self–blocking.

Let $S = \{(E_1, B_1), ..., (E_n, B_n)\}$ be conflict–free in F_{AC}^D and $T = \bigcup_{i=1}^n E_i$ its corresponding set of arguments in D. By the previous comment we can observe that every pd–evaluation argument in S is acyclic. Moreover, due to conflict–freeness of S, we can observe that for all i, j, $E_i \cap B_j = \emptyset$. Consequently, every argument in T has an unlocked acyclic pd–evaluation on T and thus the set is pd–acyclic conflict–free in D.

Let now $S' \subseteq A$ be pd–acyclic conflict–free in D. Every argument $s \in S'$ has an unblocked acyclic pd–evaluation, and without the loss of generality we can take a minimal one. Thus, for every argument $s \in S'$ we can choose an AF argument corresponding to this evaluation, and since the evaluation is unblocked and acyclic then it follows that the set $V \subseteq A'$ consisting of them is conflict–free.

Let now $S = \{(E_1, B_1), ..., (E_n, B_n)\}$ be conflict–free in F_{AC}^D and let $T = \bigcup_{i=1}^n E_i$ be its corresponding pd–acyclic conflict–free set of D. We would like to show the relation between the discarded sets of S and T. Let S^+ be the set of arguments in A' attacked by S in F_{AC}^D and $X_S = \{a \in A \mid \forall(F, V) \in A', a \in F, (F, V) \in S^+\}$ the set of D arguments for which all evaluation arguments are in S^+. We will show that X_S corresponds to the set T^+. First of all, if an argument $a \in A$ is in X_S, then all of the evaluation arguments in A' that contain it in the pd–sequence/pd–set are attacked by S. This means that for every (F, V) containing a, $\bigcup_{i=1}^n E_i \cap V \neq \emptyset$. From this follows that every evaluation of this argument is blocked through the blocking set by the members of T. Thus, by Lemma 2.125, every such argument is in T^+. Let us now assume that an argument a is in T^+, but not in X_S. This means that even though for every evaluation (F, V) of a, $T \cap V \neq \emptyset$ by Lemma 2.125, there is an evaluation (F', V') for a in A' which is not attacked by S. However, if there is no argument (G, H) in S s.t. $G \cap C' \neq \emptyset$, then obviously it cannot be the case that $T \cap V' \neq \emptyset$. We reach a contradiction. Thus, whatever is in T^+, is in X_S.

Let $S = \{(E_1, B_1), ..., (E_n, B_n)\}$ be admissible in F_{AC}^D and $T = \bigcup_{i=1}^n E_i$ its corresponding set of arguments in D. By the previous parts of this proof we know that T is pd–acyclic conflict–free in D. Let us now assume that even though S is admissible in F_{AC}^D, T is not ac–admissible in D. This means it that there is an argument $a \in T$ that has no evaluation on T for which the blocking set would be contained in T^+. Since a originates from some acyclic pd–evaluation in S, then a possesses at least one acyclic pd–evaluation on T, and as none of the blocking sets of such evaluations are completely in T^+, they cannot be empty to start with. Every such blocking set will contain an argument not in T^+, and thus one possessing at least one evaluation on A, which by construction will appear in F_{AC}^D. Hence, for any argument $(F, V) \in S$ s.t. $a \in F$, there exists $c \in V$ and $(F', V') \in A'$ s.t. $c \in F'$. Since the arguments are not in T^+, they will not be present in X_S and consequently there will always be such evaluation argument (F', V') in A' which is not attacked by S. This obviously breaches the defense and we reach a contradiction. Thus, we can conclude that if S is admissible in F_{AC}^D, then T is ac–admissible in D.

Let now $S' \subseteq A$ be ac–admissible in D. First of all, every argument in S' has at least one acyclic pd–evaluation on S' s.t. its blocking set is contained in the standard discarded set of S'. By collecting all such evaluations of all arguments in S' and repeating the procedure from the conflict–free proof, we can easily construct a corresponding conflict–free set $V \subseteq A'$ in F^D_{AC}. Let us assume V is not admissible in F^D_{AC}. This means there exists an argument F in A' s.t. F attacks an argument in V and it is not attacked by any element of V. Clearly, this argument is not in V^+, and consequently none of the arguments in its pd–sequence/pd–set are in X_S. From this follows they are also not in the standard discarded set of S'. Since the pd–sequence/pd–set of F contains an argument f appearing in the blocking set of an evaluation of an argument $s \in S'$ and f is not in the discarded set, we reach a contradiction with the construction of the AF extension. Thus, V has to be admissible in F^D_{AC}.

Let $S = \{(E_1, B_1), ..., (E_n, B_n)\}$ be complete in F^D_{AC}. By the previous parts of this proof, the set $T = \bigcup_{i=1}^{n} E_i$ is ac–admissible in D. Let us assume it is not ac–complete; this means there exists an argument $a \notin T$ which is decisively in w.r.t. v_T. We can observe that it is possible to construct an acyclic pd–evaluation with the members of T being its pd–sequence and its blocking set contained in T^+. Thus, we can use a decisively in interpretation for a of which v_T is a completion to extend this evaluation into an acyclic pd–evaluation (G, H) for a. Due to decisiveness, H would still be contained in T^+. Thus, by the previous parts of this analysis, (G, H) is an argument in A' and is defended by S. As it cannot be the case that it is contained in S (otherwise a would have appeared in T), we reach a contradiction with th completeness of S. Therefore, it has to be the case that T is an ac–complete extension of D.

Let now $S' \subseteq A$ be ac–complete in D and let $V \subseteq A'$ be its corresponding ac–admissible extension in F^D_{AC}. Let us now assume V is not complete, i.e. there is an argument $F \in A' \setminus V$ defended by V. Since V is conflict–free and defends F, then it cannot be the case that F is not an acyclic pd–evaluation – otherwise F would be a self–attacker and defending it would breach the conflict–freeness of V. Let us assume that all arguments in the pd–sequence of F are contained in S'. Therefore, by the construction of V, this particular evaluation for a did not satisfy the admissibility criterion, i.e. its blocking set was not falsified by the range. This means there is some argument b in the blocking set of F that is not in the standard discarded set of S'. From this follows that there is an AF argument in A' corresponding to this evaluation that attacks F and is not attacked by V. Thus, F could not have been defended by V and we reach a contradiction.

Let us thus assume that at least one argument b in the pd–sequence of F is not contained in S'; let b be the first such element and v_b the decisively in interpretation with which it entered F. By ac–completeness of S' this means that b is not decisively in w.r.t. $v_{S'}$. By the construction of the pd–sequence and the fact it is the first argument not contained in S', it has to be the case that $v_b^{\mathbf{t}} \subseteq S'$. Therefore, there exists an argument in $v_b^{\mathbf{f}}$ that is not in the discarded set of S'. If this member is actually in S', then it is easy to see that V attacked F and thus could not have defended it. If this member was not in S', then we can repeat previous parts of the analysis and conclude that V could not have defended F again. We reach a contradiction and thus V is complete in F^D_{AC}.

We can observe that there is a one–to–one relation between the complete and ac–

complete extensions of D and F^D_{AC}. This comes from the fact that if we had two different complete extension S_1 and S_2 in F^D_{AA} corresponding to the same D one, their discarded sets would have to be the same and thus defending the same arguments. Consequently, as we are dealing with complete extensions, every argument in S_1 has to be in S_2 and vice versa, thus showing that $S_1 = S_2$.

Additionally, it also easy to see that if two ac–complete extensions of T_1 and T_2 of D are s.t. $T_1 \subseteq T_2$, then their corresponding complete extensions S_1 and S_2 of F^D_{AC} are also s.t. $S_1 \subseteq S_2$. Thus, by the one–to–one relation and Theorems 2.10 and 2.158 we can show the correspondence between the preferred and ac–preferred extensions. The relation between the grounded extensions can be shown in the same way. □

Theorem 12.3. *Let $D = (A, C)$ be an ADF and F^D_{CC} its corresponding AF obtained from Translation 85. If $S \subseteq A$ is a conflict–free (cc–admissible, cc–complete, cc–preferred, grounded) extension of D, then there exists a conflict–free (admissible, complete, preferred, grounded) extension $S' = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ of F^D_{CC} s.t. $S = \bigcup_{i=1}^{n} F_i \cup G_i$. If $S' = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ is a conflict–free (admissible, complete, preferred, grounded) extension of F^D_{CC}, then $S = \bigcup_{i=1}^{n} F_i \cup G_i$ is conflict–free (cc–admissible, cc–complete, cc–preferred, grounded) extension of D.*

Proof. Let $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ be a conflict–free set of F^D_{CC} and $T = \bigcup_{i=1}^{n} F_i \cup G_i$ its associated set in D. It follows from the construction of the evaluations that for every argument $a \in T$ there is at least one decisively in interpretation v_a s.t. $v_a^{\mathbf{t}} \subseteq T$ with which it entered any of the evaluations (F_i, G_i, B_i). Since S' is conflict–free in F^D_{CC}, the false part of this interpretation is not in T. Thus, every $t \in T$ will have a satisfied acceptance condition. Conflict–freeness of T in D follows easily.

Let $S \subseteq A$ be a conflict–free set of D. Since it is conflict–free, then we can easily construct at least standard evaluations on S for every $s \in S$ s.t. the blocking set of this evaluation is disjoint from S. These evaluations can be transformed into partially acyclic ones, even if the pd–sequence of the result would be empty. Consequently, we can construct a set $V \subseteq A^{ev}$ in F^D_{CC} corresponding to these evaluations. Since their blocking set are disjoint from S, it is easy to see that no two arguments in V can be in conflict. Thus, the set is conflict–free in F^D_{CC}.

Let $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ be a conflict–free set of F^D_{CC} and $T = \bigcup_{i=1}^{n} F_i \cup G_i$ its corresponding conflict–free extension of D. Let S^+ be the set of arguments in A' attacked by S and $X_S = \{a \in A \mid \forall (F, G, B) \in A^{ev}\, s.t.\, a \in F \cup G,\ (F, G, B) \in S^+\}$ be the set of ADF arguments for which all of their evaluations are in S^+. We will show that X_S corresponds to the standard discarded T^+ of T in D. Let us assume that an argument $a \in A$ is in X_S, but not in T^+. By Lemma 2.125, this means that a has a standard evaluation (F', B') on A s.t. $B' \cap T = \emptyset$. This evaluation can be transformed into a partially acyclic one and will appear as an argument in A^{ev}. Since $B' \cap T = \emptyset$, then for every evaluation argument $(G, H, J) \in S$, $(G \cup H) \cap B' = \emptyset$. Therefore, by the construction of F^D_{CC}, it cannot be the case that $(F', B') \in S^+$. Consequently, a could not have been in X_S and we reach a contradiction. Let us now assume there is an argument

$a \in A$ which is in T^+, but not in X_S. Since $A \notin X_S$, there is a partially acyclic pd–evaluation (F', G', B') s.t. $a \in (F' \cup G')$ which is not in S^+. Based on the fact that there is no $(F'', G'', B'') \in S$ s.t. $(F'' \cup G'') \cap B' \neq \emptyset$ and the way T is constructed, we can conclude that $T \cap B' = \emptyset$. As $(F' \cup G', B')$ is a standard evaluation for a, then by Lemma 2.125 we can conclude that a could not have been in T^+. We reach a contradiction and can conclude that $X_S = T^+$.

Let $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ be admissible in F^D_{CC} and let $T = \bigcup_{i=1}^n F_i \cup G_i$ be its corresponding conflict–free set in D. Let us assume that T is not cc–admissible. This means there is an argument $a \in A$ which is not decisively in w.r.t. the range interpretation v_T, which is equivalent to every partially acyclic evaluation on T for a having a blocking set not entirely contained in T^+. Thus, for every evaluation (F, G, B) for a on T, there exists an argument $b \in B$ with a partially acyclic evaluation (F', G', B') s.t. $B' \cap T = \emptyset$ (see Lemma 2.125). Based on the relation between the discarded sets and the fact that S cannot contain arguments from A^b, we can observe that S could not have been admissible in F^D_{CC}. We reach a contradiction and can conclude that T is cc–admissible in D.

Let $S \subseteq A$ be a cc–admissible extension of D. Therefore, every argument $s \in S$ has at least one standard evaluation on S whose blocking set is in S^+. Every such pd–evaluation can be transformed into a partially acyclic one. We can thus collect arguments in A^{ev} corresponding to these evaluations and construct a set $V \subseteq A^{ev}$, which by the previous analysis, will be at least conflict–free in F^D_{CC}. Let us assume that V is not admissible in F^D_{CC}; this means it contains an evaluation argument (F, G, B) for which there exists an attacker $Z \in A'$ that is not attacked by any element of V. If $Z \in A^b$, then by the Translation 85, Z represents an argument $a \in A$ s.t. $a \in F$ and as $(F, G, B)RZ$ and $(F, G, B) \in V$, V defends the argument against Z. Thus, it has to be the case that $Z \in A^{ev}$. Since $Z = (F', G', B')$ is not in V^+, elements of $F' \cup G'$ are not in X_V and consequently, not in S^+. Therefore, the blocking set of (F, G, B) could not have been completely falsified by v_S and we reach a contradiction with the construction of V. Hence, if S is cc–admissible in D, then so is V in F^D_{CC}.

Let $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ be a complete extension of F^D_{CC} and $T = \bigcup_{i=1}^n F_i \cup G_i$ its corresponding cc–admissible set in D. Let us assume T is not cc–complete. This means there exists an argument $a \notin T$ which is decisively in w.r.t. v_T, even though no partially acyclic pd–evaluation for a is defended by S. By the cc–admissibility of T, we can construct a partially acyclic evaluation on T containing all elements of T in its pd–set and pd–sequence and s.t. its blocking set is contained in T^+. Based on the decisiveness of a w.r.t. v_T, we can extend the pd–sequence of this evaluation with a and its decisively in interpretation that was contained in v_T. Thus, we can create a partially acyclic pd–evaluation (F, G, B) for a on $T \cup \{a\}$ s.t. $B \subseteq T^+$. We can extract a minimal evaluation from it that will appear as an argument in A^{ev}. As it is not in S, it is not defended by S. Therefore, by the previous parts of this proof and the relation between the discarded sets, we can observe it can only be the case that there is an argument $Z \in A^b$ attacking it. Thus, by Translation 85, Z represents an argument c in the pd–set of our evaluation. As S does not attack Z, there is no evaluation argument in S containing c in its pd–sequence or pd–set. However, then it could not have been the case that $c \in T$. Since $a \neq c$ as a can only appear in the pd–sequence, then (F, G, B) could not have been an evaluation for a satisfying our requirements. Therefore, it could not have been the case that a was decisively in w.r.t. v_T. We can thus conclude that if S is complete in F^D_{CC}, then T is cc–complete in D.

Let $S \subseteq A$ be a cc–complete extension of D and $V \subseteq A^{ev}$ its corresponding admissible set in F^D_{CC}. Let us assume that V is not complete in F^D_{CC}; this means there exists an argument $Z \in A' \setminus V$ which is defended by V. If $Z \in A^b$, then it is a self–attacker by Translation 85, and defending it would breach the conflict–freeness of V. If $Z = (F, G, B) \in A^{ev}$, then we can consider two situations: one in which the cc–admissible set in D corresponding to $\{Z\} \cup V$ (see Lemma 2.22) is equal to S and one in which it is not. Let us focus on the first case. By the construction of V this means that the blocking set of the evaluation represented by Z is not completely contained in S^+. Consequently, there exists at least one argument $a \in B$ possessing a standard evaluation with a blocking set disjoint from S. We can transform this pd–evaluation into a partially acyclic pd–evaluation and extract a minimal one out of it, which will appear as an argument in A^{ev}. By the relation between S^+, X_V and V^+ it holds that this argument will not be attacked by V and will have the power to attack Z. Thus, V could not have defended Z and we reach a contradiction. Let us thus assume that the set $S' \subseteq A$ in D corresponding to $\{Z\} \cup V$ is not equal to S. It is easy to see that it has to be the case that $S \subset S'$. This means there is at least one argument $a \in F \cup G$ which is not in S (and thus no evaluation containing a in the pd–set or pd–sequence can already be in V). If $a \in F$, then Z could not have been defended by V from the breaker argument $F \in A^b$ and we reach a contradiction. Let us thus assume that $a \in G$ and without the loss of generality, let a be the first argument in the sequence that is not in S. Let v be the decisively in interpretation for a with which it entered Z. Since elements of F and the ones from G preceding it are in S and b is not decisively in w.r.t. v_S, then there is an argument $c \in v^{\mathbf{f}}$ s.t. $c \notin S^+$. If $c \in S$, then (F, G, B) would be attacked by V, thus making defense impossible. If $c \notin S$, then it possesses a partially acyclic evaluation not blocked by S through the blocking set (see Lemma 2.125) which appears in A^{ev}, attacks (F, G, B) and is not attacked in turn. We reach a contradiction. Thus, if S is cc–complete in D, then V is complete in F^D_{CC}.

We can observe that there is a one–to–one relation between the complete and cc–complete extensions of D and F^D_{CC}. This comes from the fact that if we had two different complete extension S_1 and S_2 in F^D_{CC} corresponding to the same one for D, their discarded sets would have to be the same – even the arguments attacked in A^b would have to be the same due the fact that the pd–sets and pd–sequences of arguments in S_1 and S_2 amount to the same set. Thus, S_1 and S_2 defend the same arguments. Consequently, as we are dealing with complete extensions, every argument in S_1 has to be in S_2 and the vice versa, thus showing that $S_1 = S_2$.

Additionally, it also easy to see that if two cc–complete extensions of T_1 and T_2 of D are s.t. $T_1 \subseteq T_2$, then their corresponding complete extensions S_1 and S_2 of F^D_{AC} are also s.t. $S_1 \subseteq S_2$. Thus, by the one–to–one relation and Theorems 2.10 and 2.158 we can show the correspondence between the preferred and cc–preferred extensions. The relation between the grounded extensions can be shown in the same way. □

Theorem 12.4. *Let $D = (A, C)$ be an ADF and $F^D_{CA_2} = (A', R)$ its corresponding AF obtained from Translation 86. If $E \subseteq A$ is a conflict–free (ca$_2$–admissible, ca$_2$–complete, ca$_2$–preferred, model, grounded) extension of D, then there exists a conflict–free (admissible, complete, preferred, stable, grounded) extension $E' = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n), a^b_1, ..., a^b_m\} \subseteq A'$ of $F^D_{CA_2}$ s.t. $E = \bigcup^n_{i=1} F_i \cup G_i$. If $E' = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n), a^b_1, ..., a^b_m\} \subseteq A'$ is a conflict–free (admissible, stable, grounded) extension of $F^D_{CA_2}$, then $E = \bigcup^n_{i=1} F_i \cup G_i$ is conflict–free (ca$_2$–admissible, model, grounded) extension of D.*

Proof. Most of the proof of this theorem will depend on the relation between extensions and labelings of ADFs (see Section 2.3.7) and Theorem 12.5 that will be discussed in the next section.

Let $E \subseteq A$ be a conflict–free extension of D. Let $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\}$ be the collection of all minimal partially acyclic evaluations on E for every argument $a \in E$ s.t. for every i, j, $(F_i \cup G_i) \cap B_j = \emptyset$. We can observe that due to the fact that E is conflict–free, every argument $a \in E$ will posses at least one such unblocked evaluation on E. Based on the construction of $F^D_{CA_2}$ we can observe that S is conflict–free in $F^D_{CA_2}$.

Let $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n), a^b_1, ..., a^b_k\}$ be a conflict–free extension of $F^D_{CA_2}$ and $E = \bigcup^n_{i=1} F_i \cup G_i$ the associated set of arguments in D. Let $a \in E$ be an argument and $(F, G, B) \in S$ an evaluation s.t. $a \in F \cup G$. Let v be the decisively in interpretation with which a entered (F, G, B). Since S is conflict–free, $B \cap E = \emptyset$. Consequently, $v^{\mathbf{t}} \subseteq E$ and $v^{\mathbf{f}} \cap E = \emptyset$. Thus, $C_a(E \cap par(a)) = in$ and E is conflict–free in D. By using Theorems 2.166 and 12.5, we can show that every admissible extension of D is admissible in $F^D_{CA_2}$. What remains to be shown is that the other way around also holds, which had its issue for the labeling–based semantics. Let $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n), a^b_1, ..., a^b_k\}$ be an admissible extension of $F^D_{CA_2}$ and $E = \bigcup^n_{i=1} F_i \cup G_i$ its associated set of arguments in D. By the previous parts of this proof we can observe that E is conflict–free in D. Let E^{p+} be the partially acyclic discarded set of E and $X = \{b \in A \mid$ every minimal partially acyclic evaluation (F, G, B) s.t. $a \in F \cup G$ is in S^+ in $F^D_{CA_2}\}$ the set of arguments s.t. all of their evaluations are attacked by S.

We will show that $X \subseteq E^{p+}$. Let a be an argument in X; assume that it is not present in E^{p+}. This means there exists a partially acyclic evaluation (F, G, B) for a s.t. $F \subseteq E$ and $B \cap E = \emptyset$. Without the loss of generality, we can assume that this evaluation is minimal. This evaluation will appear as an argument in A^{ev}. We can observe that from the construction of $F^D_{CA_2}$ and the fact that $B \cap E = \emptyset$ and $E = \bigcup^n_{i=1} F_i \cup G_i$, it cannot be the case that there is an evaluation argument in S attacking (F, G, B). If $F = \emptyset$, then there is no breaker argument attacking (F, G, B) in A'. Hence, S cannot attack (F, G, B), and $a \notin X$. We reach a contradiction. If $F \neq \emptyset$, then for every $b \in F$, there is a breaker argument $b^b \in A^b$ and this argument attacks (F, G, B) in R. However, as $F \subseteq E$, for every $b \in F$ we can find an evaluation argument in S that contains b in its pd–set or pd–sequence. Since this argument attacks b^b, it cannot be the case that $b^b \in S$. Hence, S cannot attack (F, G, B), and $a \notin X$. We can finally conclude that $X \subseteq E^{p+}$.

We can now observe that based on the admissibility of S and the construction of $F^D_{CA_2}$, if $(F, G, B) \in S$, then $B \subseteq X$. Consequently, for any argument $a \in E$, we can find a partially acyclic evaluation containing this argument in the pd–set or pd–sequence s.t. $F \cup G \subseteq E$ and $B \subseteq E^{p+}$. Therefore, if v_a is the minimal decisively in interpretation for a with which it entered this evaluation, then $v_a^{\mathbf{t}} \subseteq E$ and $v_a^{\mathbf{f}} \subseteq E^{p+}$. Therefore, the partially acyclic range of E is a completion of v_a, and it must be the case that a is decisively in w.r.t. this range. Hence, E is an admissible extension of D.

The correspondence between the grounded extensions follows from Theorems 2.169 and 12.5.

The (one way) relation between the (ca$_2$) preferred extensions follows from Theorems 2.167 and 12.5.

The (one way) relation between the (ca$_2$) complete extensions follows from Theorems 2.168 and 12.5.

What remains to be shown is the correspondence between the model and stable extensions between D and $F^D_{CA_2}$. Let $E \subseteq A$ be a model of D. Let $S^{ev} = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\}$ be the collection of all minimal partially acyclic evaluations on E for every argument $a \in E$ s.t. for every i, j, $(F_i \cup G_i) \cap B_j = \emptyset$. We can observe that due to the fact that E is conflict–free, every argument $a \in E$ will posses at least one such unblocked evaluation on E. Let $S^b = \{a^b \mid a^b \in A^b, \nexists (F, G, B) \in S^{ev}$ s.t. $a \in F \cup G\}$. Based on the construction of $F^D_{CA_2}$ we can observe that $S = S^{ev} \cup S^b$ is conflict–free in $F^D_{CA_2}$. What remains to be shown is that it is also stable. Assume it is not the case and that there exists an argument $Z \in A' \setminus S$ s.t. there is no $W \in S$ with WRZ. We can observe that if a breaker argument is not in S^b, then it has to be the case that is attacked by an argument in S^{ev}. Consequently, it can only be the case that $Z = (F', G', B') \in A^{ev} \setminus S^{ev}$. As Z is not attacked by any breaker argument in S^b, then it must be the case that $F' \subseteq E$. As it is not attacked by any argument in S^{ev}, $E \cap B' = \emptyset$. This means that if $G' = \emptyset$, then by the construction of S^{ev} it has to be the case that $Z \in S^{ev}$ and we reach a contradiction. Let us assume that $G' \neq \emptyset$ and let $G' = (g_0, ..., g_m)$. With every argument g_i we associate a decisively in interpretation v_i with which it entered Z. Let us now go through the sequence. We can observe that $v_0^{\mathbf{t}} \subseteq F'$ and $v_0^{\mathbf{f}} \subseteq B'$, hence $v_0^{\mathbf{t}} \subseteq E$ and $v_0^{\mathbf{t}} \cap E = \emptyset$. Consequently, we can show that $C_{g_0}(E \cap par(g_0)) = in$. Hence, as E is a model, it has to be the case that $g_0 \in E$. From this and the fact that $v_1^{\mathbf{t}} \subseteq F' \cup \{g_0\}$ it follows that $v_1^{\mathbf{t}} \subseteq E$. Moreover, as again $v_1^{\mathbf{f}} \cap E = \emptyset$, we can show that the condition of g_1 is satisfied by E. Therefore, $g_1 \in E$. We can continue in this manner until we reach the conclusion that $g_m \in E$, thus forcing the presence of Z in S^{ev} by the construction of the set. We reach a contradiction with our assumptions. Hence, S is a stable extension of $F^D_{CA_2}$.

Let $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n), a_1^b, ..., a_k^b\}$ be a stable extension of $F^D_{CA_2}$ and $E = \bigcup_{i=1}^n F_i \cup G_i$ the associated conflict–free set of arguments in D. Let us assume that E

is not a model. This means there exists an argument $a \in A \setminus E$ s.t. $C_a(E \cap par(a)) = in$. Due to the fact that E is conflict–free, it can be represented as a standard evaluation (see Section 2.3.5) and thus as a partially acyclic one. Since $C_a(E \cap par(a)) = in$, we can find a minimal decisively in interpretation v_a for a s.t. $v_a^{\mathbf{t}} \subseteq E$ and $v_a^{\mathbf{f}} \cap E = \emptyset$. We can thus extend the partially acyclic evaluation for E with v_a and extract a minimal evaluation (F', G', B') for a from it. This evaluation will appear in A^{ev}. We can observe that since $B' \cap E = \emptyset$, there cannot be any evaluation argument in S attacking (F', G, B'). Due to the fact that $F' \subseteq E$ and a is an element of the pd–sequence, it cannot be the case there is a breaker argument attacking (F', G', B') either – we either contradict the conflict–freeness of E or the construction of $F^D_{CA_2}$. Thus, by the stability of S, $(F', G', B') \in S$, which is impossible due to the fact that $a \in G'$ and $a \notin E$. Therefore, we reach a contradiction, and can conclude that E is a model of D.

Let $E \subseteq A$ be a model of D and $S_1 = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n), a_1, ..., a_k\}$ an associated stable extension of $F^D_{CA_2}$ constructed as in the previous parts of this proof. Let us assume there exists another stable extension $S_2 = \{(H_1, J_1, K_1), ..., (H_m, J_m, K_m), b_1, ..., b_l\}$ of $F^D_{CA_2}$ s.t. $\bigcup_{i=1}^{l} H_i \cup J_i = E$. Let us assume there is an evaluation argument (F', G', B') present in S_1, but not in S_2. This means there has to be an argument $Z_2 \in S_2$ attacking this evaluation. If Z_2 is itself an evaluation argument, then its pd–set or pd–sequence contains an argument from B'. Due to the fact that both pd–set and pd–sequence of Z_2 are in E, based on the previous explanations we would breach the conflict–freeness of both E and S_2. If Z_2 is a breaker argument, then the ADF argument it represents has to be in F'. Thus, it is also in E, and as such appears in an evaluation argument in S_2. Consequently, we breach the conflict–freeness of S_2. Hence, $S_1 \subseteq S_2$, and we can use the same line of reasoning to show that $S_2 \subseteq S_1$. This brings us to the conclusion that $S_1 = S_2$ and every model extension of D is associated with precisely one stable extension of $F^D_{CA_2}$. □

Theorem 12.5. *Let $D = (A, C)$ be an ADF and $F^D_{lab} = (A', R)$ its corresponding AF obtained through Translation 86. If v is an admissible labeling of D, then there exists an admissible labeling v' of F^D_{lab} s.t. $v^{\mathbf{t}} = \biguplus^{E}_{V} in(v')$ and $EV^p(v^{\mathbf{f}}) \subseteq out(v')$. If v is a complete (preferred, grounded) labeling of D, then there exists a complete (preferred, grounded) labeling v' of F^D_{lab} s.t. $v^{\mathbf{t}} = \biguplus^{E}_{V} in(v')$ and $v^{\mathbf{f}} = ALL(out(v'))$.*

If v' is an complete (preferred, grounded) labeling of F^D_{lab}, then a labeling v of D s.t. $v^{\mathbf{t}} = \biguplus^{E}_{V} in(v')$ and $v^{\mathbf{f}} = ALL(out(v'))$ is complete (preferred, grounded) in D. This does not necessarily hold for admissible semantics.

Proof. In order to show the correspondence between the labelings of D and F^D_{lab}, we will use the following construction. Let v be a three–valued admissible labeling of D and v_2 its maximal two–valued subinterpretation. Let us define the following sets:

- $O^{ev} = \{(F, G, B) \in EV^p(A) \mid v_2 \text{ blocks } (F, G, B)\}$
- $I^{ev} = \{(F, G, B) \in EV^p(v^{\mathbf{t}}) \mid F \cup G \subseteq v^{\mathbf{t}},\ B \subseteq v^{\mathbf{f}}\}$

- $O^b = \{a^b \mid \exists(F, G, B) \in I^{ev}$ s.t. $a \in F \cup G\}$
- $I^b = \{a^b \mid \forall(F'', G'', B'')$, s.t. $a \in F'' \cup G''$, $(F'', G'', B'') \in O^{ev}\}$

We will now show that the respective ev and b sets are disjoint. We can observe that no evaluation contained in I^{ev} can possibly be blocked by v and thus does not meet the O^{ev} requirements. Consequently, $I^{ev} \cap O^{ev} = \emptyset$. The fact that O^b and I^b are disjoint follows from the construction. Therefore, all of the introduced sets are disjoint, and we can create a labeling associated with them s.t. the arguments in I^{ev} and I^b are assigned $\mathbf{t}$ and those from O^{ev} and O^b are assigned $\mathbf{f}$.

Let now v be an admissible labeling of D and v' be a labeling in F^D_{lab} s.t. $in(v') = I^{ev} \cup I^b$ and $out(v') = O^{ev} \cup O^b$. We will first show that v' is related to v in the manner described in the theorem; later, we will prove that v' is admissible in F^D_{lab}. Let $a \in v^{\mathbf{t}}$ be an accepted argument. We can show that for any such a, v is a completion of a decisively in interpretation for a (see Theorem 2.148). Consequently, from v we can extract a minimal decisively in interpretation v_a for a. Therefore, $v^{\mathbf{t}}_a \subseteq v^{\mathbf{t}}$ and $v^{\mathbf{f}}_a \subseteq v^{\mathbf{f}}_a$. By assigning every argument $a \in v^{\mathbf{t}}$ such a minimal decisively in interpretation we can create a sound pd–function pd on $v^{\mathbf{t}}$. Since the $\mathbf{t}$ assignments of every such interpretation are contained in $v^{\mathbf{t}}$, it is easy to see that for any argument accepted in $v^{\mathbf{t}}$ we can create a minimal partially acyclic evaluation on this set and w.r.t. pd. Moreover, based on the relation between the $\mathbf{f}$ assignments of v and the interpretations in pd, the blocking set of any such evaluation will have to be contained in $v^{\mathbf{f}}$. Thus, these evaluations will be contained in I^{ev}, and it is now easy to show that $\biguplus^{E}_{V} in(v') = v^{\mathbf{t}}$. Let $a \in v^{\mathbf{f}}$ be an argument rejected by the admissible labeling in D and assume there exists a partially acyclic evaluation for a not blocked by the maximal two–valued subinterpretation v_2 of v. Without the loss of generality, we can focus on minimal evaluations. Let (F, G, B) be this evaluation; we can observe that no argument in $F \cup G$ is mapped to false by v_2 and no argument in B is be mapped to $\mathbf{t}$ by v_2. Let z be the minimal decisively in interpretation of a used in construction of (F, G, B). We can observe that, based on the definition of a decisively in interpretation, it holds that $C_a(z^{\mathbf{t}}) = in$. Moreover, since $B \cap v^{\mathbf{t}} = \emptyset$, it can be shown that $C_a(z^{\mathbf{t}} \cup (v^{\mathbf{t}} \cap par(a))) = in$. However, this means that a cannot be decisively out w.r.t. v_2, which by Theorem 2.148 means that v cannot be admissible. Therefore, all evaluations for a are blocked by v_2 and it holds that $EV^p(v^{\mathbf{f}}) \subseteq O^{ev} \subseteq out(v')$. We can conclude that v' and v are related in the desired manner.

Let us now show that v' is an admissible labeling of F^D_{lab}; however, we first need to prove that it is conflict–free. Assume it is not the case; thus, there exist arguments $a, b \in A'$ s.t. aRb and $v'(a) = v'(b) = in$ or there is an argument $c \in A'$ s.t. $v'(c) = out$, but for no $d \in A'$ s.t. dRc, $v'(d) = in$.

We will now focus on the first case; assume that $a = a^b \in A^b$ is a breaker argument. Thus, by the construction of v', $a^b \in I^b$. By Translation 86, we can observe it can only be the case that b is an evaluation argument. Let $b = (F, G, B)$; by the construction of F^D_{lab}, $a \in F$. However, by the construction of v', $(F, G, B) \in I^{ev}$ and therefore $(F, G, B) \notin O^{ev}$. This means that a^b does not qualify for I^b and we reach a contradiction. Let us now assume that $a = (F', G', B') \in A^{ev}$ is an evaluation argument. Consequently, b is either an evaluation argument (F'', G'', B'') s.t. $(F' \cup G') \cap B'' \neq \emptyset$, or a breaker argument b^b s.t. $b \in F' \cup G'$. If it is the first case and b is an evaluation, then by the construction of v', both $a, b \in I^{ev}$. This means that $B'' \subseteq v^{\mathbf{f}}$ and as a result, $B'' \cap v^{\mathbf{t}} = \emptyset$. Since $F' \cup G' \subseteq v^{\mathbf{t}}$, it has to be the case that $(F' \cup G') \cap B'' = \emptyset$, and we reach a contradiction. If it is the second case and $b = b^b$ is a breaker, then by the construction of v', $a \in I^{ev}$ (and therefore, $a \notin O^{ev}$) and $b^b \in I^b$. However, by the construction of I^b and the fact that $b \in F' \cup G'$, it has to be the case that $a \in O^{ev}$. We reach a contradiction again. We can therefore conclude that there are no arguments $a, b \in in(v')$ s.t. aRb.

Let us now check whether for every argument $c \in A'$ s.t. $v'(c) = out$ we can find an argument $d \in A'$ s.t. dRc and $v'(d) = in$. By Translation 86 and the construction of O^b, it is easy to see that for every argument in O^b we can find an attacker in I^{ev}. What remains to be shown is that for every argument in O^{ev} we can find an attacker in $I^{ev} \cup I^b$. Let (F, G, B) be an arbitrary partially acyclic evaluation in O^{ev}. We know it is blocked by v_2; the blocking can now occur either by accepting a member of the blocking set or by falsifying an argument in the pd–set or the pd–sequence of the evaluation.

We can start by considering the case where blocking occurs through the blocking set. By Translation 86, previous analysis and the construction of I^{ev}, this means there is an evaluation (F', G', B') in I^{ev} s.t. $(F' \cup G') \cap B \neq \emptyset$ and that (F', G', B') attacks (F, G, B) in F^D_{lab}. Consequently, we can find a suitable attacker for c.

Let us now assume that blocking occurs only by falsifying an argument in the pd–set or pd–sequence. We will first consider the case in which $F = \emptyset$; in other words, we are in fact dealing with an acyclic pd–evaluation, and this evaluation will not be attacked by any breaker argument from I^b. Let $G = (g_0, ..., g_n)$ be the pd–sequence of our evaluation and $v_{g_0}, ..., v_{g_n}$ the minimal decisively in interpretations used in the construction of (F, G, B). Since $v^{\mathbf{t}}_{g_0} = \emptyset$ and $v^{\mathbf{f}}_{g_0} \subseteq B$, it follows that $v^{\mathbf{t}}_{g_0} \cap v^{\mathbf{f}} = \emptyset$ and $v^{\mathbf{f}}_{g_0} \cap v^{\mathbf{t}} = \emptyset$. Thus, we can show that g_0 cannot be decisively out w.r.t. v_2 and by Theorem 2.148, we can conclude that $v(g_0) \neq \mathbf{f}$. Consequently, the evaluation cannot be blocked by falsifying g_0. Let us consider g_1 now; since $v^{\mathbf{f}}_{g_1} \subseteq \{g_0\}$, $v^{\mathbf{f}}_{g_1} \subseteq B$ and $g_0 \notin v^{\mathbf{f}}$, it follows that $v^{\mathbf{t}}_{g_1} \cap v^{\mathbf{f}} = \emptyset$ and $v^{\mathbf{f}}_{g_1} \cap v^{\mathbf{t}} = \emptyset$. Again, g_1 cannot be decisively out w.r.t. v_2 and cannot be mapped to $\mathbf{f}$ by v. We can repeat this analysis till we reach g_n and conclude that v could not have blocked the evaluation. Therefore, it could not have possibly been in O^{ev} in the first place and we reach a contradiction.

Let us now consider the case where $F \neq \emptyset$. By repeating the reasoning above, we can show that no argument in G can be mapped to $\mathbf{f}$ by v without some argument $a \in F$ being mapped to $\mathbf{f}$ as well. However, since $a \in F$, then by the construction of partially acyclic evaluations and F^D_{lab}, we can observe that there will exist a breaker argument $a^b \in A^b$ for a. Moreover, since v is admissible and $v(a) = \mathbf{f}$, it means that every partially acyclic evaluation containing a will be blocked by v and thus contained in O^{ev}. Therefore, $a^b \in I^b$ and (F, G, B) is attacked by I^b. Thus, we can again find an accepted attacker for a rejected argument. We can finally conclude that our labeling v' is conflict–free in F^D_{lab}.

Let us now show that v' is an admissible labeling in F^D_{lab}. By the conflict–freeness of

v', every argument in A' that is mapped to out by v' has at least one attacker that is mapped to in by v'. Thus, our out assignments are legal. What remains to be shown that every argument $a \in in(v')$ is legally in.

We will first assume that $a = (F, G, B) \in I^{ev}$. Let argument $b \in A'$ be an attacker of a; we need to show that $b \in out(v')$. If $b \in A^b$ is a breaker argument, then by the construction of F^D_{lab} it means that the D argument represented by b is in the pd–set of a. Consequently, even though bRa, it is the case that aRb as well. Therefore, $b \in O^b$ and $b \in out(v')$. If $b = (F', G', B') \in A^{ev}$, then it means that $B \cap (F' \cup G') \neq \emptyset$. However, by the construction of I^{ev}, $B \subseteq v^{\mathbf{f}}$, and as $EV^p(v^{\mathbf{f}}) \subseteq O^{ev}$, it follows that $b \in O^{ev}$ and $v'(b) = out$. We can therefore conclude that all arguments from I^{ev} are legally in w.r.t. v'. The fact that all arguments from I^b are legally in follows easily from the construction of I^b. Therefore, v' is an admissible labeling of F^D_{lab}.

In order to see that not every admissible labeling v' of F^D_{lab} corresponds to an admissible labeling in D, please consult Example 143.

Let v be a complete labeling of D and v' its associated labeling of F^D_{lab} created using the previous construction. By the previous parts of this proof we know that v' is admissible in F^D_{lab}. We now need to show it is complete. Let us assume it is not; this means there is an argument $a \in A'$ s.t. $v'(a) = undec$ and the assignment is not legal, i.e. a is either legally in or legally out w.r.t. v'.

Let us assume that a is in A^b and is of the form $a = c^b$ for an argument $c \in A$. We will first consider the case where it is legally out. By the construction of F^D_{lab} and v', this means that I^{ev} contains an evaluation argument (F, G, B) s.t. $c \in F \cup G$. However, this means that a qualifies for O^b and therefore $v'(a) = out$. We reach a contradiction. Let us now consider the case in which a is still a breaker argument, but is legally in. By the construction of F^D_{lab}, a is attacked by evaluation arguments (F, G, B) s.t. $c \in F \cup G$ and due to the fact that a was created, at least one such attacker must exist. Since a is legally in w.r.t. v', all such evaluation arguments are mapped to out by v'. Therefore, they must be in O^{ev}, and a qualifies for I^b. Hence, $v'(a) = in$, and we reach a contradiction.

Let us now assume that $a = (F, G, B) \in A^{ev}$ is an evaluation argument. We will first consider the case in which it is legally out. This means there exists an argument in $in(v')$ (and thus in $I^{ev} \cup I^b$) attacking it. If the attacker comes from I^{ev}, then it means that $v_2^{\mathbf{t}} \cap B \neq \emptyset$. Thus, v_2 blocks the evaluation represented by a and $a \in O^{ev}$. Consequently, it can only be the case that the attacker is in I^b. However, based on the construction of F^D_{lab}, if a breaker argument attacks an evaluation, then this evaluation attacks the breaker argument in return. Due to the fact that v' is admissible and the breaker is accepted, it has to be the case that a is in O^{ev} already. Thus, $v'(a) = out$ and we reach a contradiction.

Let now $a = (F, G, B)$ be legally in. This means that if $F \neq \emptyset$, then all breaker arguments for F are out. Hence, by the nature of O^b, $F \subseteq v^{\mathbf{t}}$. Moreover, all of the arguments in A^{ev} attacking a also have to be out. We can therefore show that $B \subseteq v^{\mathbf{f}}$; if it were not the case, then by the construction of O^{ev} it means that v_2 blocks all the evaluations of any argument $b \in B$, but does not map it to $\mathbf{f}$. Since in such a case b would be decisively out w.r.t. v_2, not mapping it to $\mathbf{f}$ would breach the completeness of v by Theorem 2.149. In order to prove that $a \in I^{ev}$, we now only need to show that $G \subseteq v^{\mathbf{t}}$. Let $G = (g_0, ..., g_n)$ be the pd–sequence of the evaluation represented by a. Since $F \subseteq v^{\mathbf{t}}$ and $B \subseteq v^{\mathbf{f}}$, we can show that v is a completion of the decisively in interpretation for g_0 with which it entered the evaluation. Hence, g_0 is decisively in w.r.t. v_2 and by Theorem 2.149, $v(g_0) = \mathbf{t}$. Now, as $F \cup \{g_0\} \subseteq v^{\mathbf{t}}$ and $B \subseteq v^{\mathbf{f}}$, then v is a completion of the decisively in interpretation for g_1 with which it entered the evaluation as well. Consequently, g_1 is also decisively in w.r.t. v_2 and has to be mapped to $\mathbf{t}$ by v. We can follow this line of reasoning to show that $F \cup G \subseteq v^{\mathbf{t}}$ and $B \subseteq v^{\mathbf{f}}$. Therefore, a has to be in I^{ev} already. We can thus conclude that v' is a complete labeling of F^D_{lab}.

Let now v' be a complete labeling of F^D_{lab} and v an interpretation on A in D s.t. $v^{\mathbf{t}} = \biguplus^{E}_{V} in(v')$ and $v^{\mathbf{f}} = ALL(out(v'))$. Let v_2 be the maximal two–valued subinterpretation of v. We will first show that v is an admissible labeling of D. By Theorem 2.148 this means that for every argument $a \in A$, if $v(a) = \mathbf{t}$ then a is decisively in w.r.t. v_2, and if $v(a) = \mathbf{f}$ then a is decisively out w.r.t. v_2.

Let us focus on the decisively in arguments first. Every argument in $v^{\mathbf{t}}$ comes from some evaluation in $in(v')$. An evaluation argument (F, G, B) is assigned in by v' in F^D_{lab} iff all of its attackers are mapped to out. Consequently, every evaluation argument containing any element from B in its pd–set or pd–sequence is mapped to out. Therefore, $B \subseteq ALL(out(v'))$ and as a result, $B \subseteq v^{\mathbf{f}}$. Since $F \cup G \subseteq v^{\mathbf{t}}$ as well, it follows that v_2 is a completion of a decisively in interpretation of any argument in $F \cup G$, including a. We can thus show that all arguments in $v^{\mathbf{t}}$ are decisively in w.r.t. v_2.

Let us now show that every argument mapped to $\mathbf{f}$ is decisively out w.r.t. v_2. Assume it is not the case and that there exists an argument $a \in A$ s.t. $v_2(a) = \mathbf{f}$ and a is not decisively out w.r.t. v_2. This means there exists a completion v_c of v_2 to A s.t. $C_a(v_c^{\mathbf{t}} \cap par(a)) = in$. Let $E = v_c^{\mathbf{t}}$. By the construction of v, $E \cap ALL(out(v')) = \emptyset$. Consequently, every $e \in E$ possesses an evaluation argument (F, G, B) not mapped to *out* by v'. Due to completeness of v', it means that none of these evaluations are attacked by $in(v')$. We can observe that it cannot be the case that $a \in F \cup G$ as all such evaluations are mapped to *out* by v' due to the fact that $a \in ALL(out(v'))$. By collecting these evaluations for the arguments in E we can create another evaluation (F', G', B').We can then extract a minimal decisively in interpretation for a from v_c and extend (F', G', B') with it in order to obtain an evaluation for a. We can then extract a minimal evaluation (F'', G'', B'') for a from the modified (F', G', B') that will appear as an argument in A'. We can observe that B'' is a subset of the union of the blocking sets of the initial evaluations and the false part of v_c. Since the initial evaluations were not attacked by $in(v')$ and v_c is disjoint from $v_2^{\mathbf{t}}$ (and thus from the pd–set and pd–sequence of any evaluation argument accepted in v'), then (F'', G'', B'') is not attacked by any evaluation argument in $in(v')$ either. As F'' is a subset of the union of the pd–sets of the initial evaluations, which are not attacked by any breaker argument in v', (F'', G'', B'') will also not be attacked by a breaker accepted in v'. Thus, (F'', G'', B'') is not attacked altogether. Hence, we obtain an evaluation argument for a not attacked by $in(v')$ in F^D_{lab}. Therefore, it cannot be the case that $a \in ALL(out(v'))$ and we reach a contradiction. Thus, a has to be decisively out w.r.t. v_2, and we can finally conclude that v

is an admissible labeling of D.

In order to show that v is complete in D, we can use Theorem 2.149 and prove that there is no argument in $v^{\mathbf{u}}$ that would be decisively in or decisively out w.r.t. v_2.

Let us assume it is not the case and there is some argument $a \in v^{\mathbf{u}}$ that is decisively in w.r.t. v_2. This means it possesses a minimal decisively in interpretation v_a s.t. $v_a^{\mathbf{t}} \subseteq v_2^{\mathbf{t}}$ and $v_a^{\mathbf{f}} \subseteq v_2^{\mathbf{f}}$. We can observe that as v is an admissible labeling, we can create a partially acyclic evaluation (F, G, B) s.t. $F \cup G = v^{\mathbf{t}}$ and $B \subseteq v^{\mathbf{f}}$. Let $G = (g_0, ..., g_n)$ be the pd–sequence of (F, G, B). We can extend this evaluation with v_a and since $v_a^{\mathbf{t}} \subseteq F \cup G$ and no decisively in interpretation used in the construction of (F, G, B) maps a to $\mathbf{t}$, the evaluation $(F, (g_0, ..., g_n, a), B \cup v_a^{\mathbf{f}})$ will be a partially acyclic evaluation for a. From it, we can extract a minimal evaluation (F', G', B') for a. It is important to notice that $F' \subseteq F$ and $B' \subseteq B \cup v_a^{\mathbf{f}} \subseteq v^{\mathbf{f}}$. Since $v^{\mathbf{f}} = ALL(out(v'))$, then all arguments in A^{ev} attacking (F', G', B') are out in v'. As $F' \subseteq F$ and all breaker arguments for F are out in v', it has to be the case that that (F', G', B') is legally in w.r.t. v' and therefore $a \in v^{\mathbf{t}}$. We reach a contradiction. Therefore, an argument that is decisively in w.r.t. v_2 is assigned $\mathbf{t}$ by v in D.

Let us now assume that there is some argument $a \in v^{\mathbf{u}}$ that is decisively out w.r.t. v_2. We can observe that if a possesses no decisively in interpretation at all (i.e. there is no set of arguments evaluating its condition to in), then $a \in ALL(out(v'))$ automatically and it cannot be the case that $a \in v^{\mathbf{u}}$. Let us therefore assume at least one such interpretation exists. Since a is decisively out w.r.t. v_2, in every minimal decisively in interpretation v_a for a there is an argument b s.t. $v_a(b) = \mathbf{t}$ and $v_2(b) = \mathbf{f}$ or $v_a(b) = \mathbf{f}$ and $v_2(b) = \mathbf{t}$. We can therefore observe that $b \neq a$ and that all partially acyclic evaluations for a have to be blocked by v_2. Let (F, G, B) be an arbitrary evaluation for a. If the blocking occurs through the blocking set, i.e. $v_a(b) = \mathbf{f}$ and $v_2(b) = \mathbf{t}$, then by the construction of v there is an evaluation argument in $in(v')$ attacking (F, G, B). If the blocking occurs through falsifying a required argument, i.e. $v_a(b) = \mathbf{t}$ and $v_2(b) = \mathbf{f}$, then it means that $b \in ALL(out(v'))$. We can observe that due to the fact that $v_a(b) = \mathbf{t}$, $(F, G, B) \in EV^p(b)$. Since $b \in ALL(out(v'))$, $in(v')$ has to attack (F, G, B). Thus, for every evaluation for a we can find an attacker in $in(v')$. Since v' is complete in F_{lab}^D, all those evaluations have to be mapped to out and therefore $a \in ALL(out(v'))$. Consequently, $v(a) = \mathbf{f}$ and we reach a contradiction with the initial assumption that $v(a) = \mathbf{u}$. This means that every argument decisively out w.r.t. v_2 is assigned $\mathbf{f}$ by v. We can now conclude that v is a complete labeling of D.

Let v be a complete labeling of D. Let us assume there are two complete labelings v' and v'' of D_{lab}^D s.t. $v' \neq v''$, $\bigcup_{V}^{E} in(v') = \bigcup_{V}^{E} in(v'')$ and $ALL(out(v')) = ALL(out(v''))$. Since $v' \neq v''$, then there exists $a \in A'$ s.t. $v'(a) \neq v''(a)$.

Let us assume that $a = b^b \in A^b$. We will first consider the case where $v'(a) = in$. Therefore, all evaluation arguments $(F, G, B) \in A^{ev}$ s.t. $b \in F \cup G$ are out in v'. In other words, all evaluation arguments containing b need to be out. Due to the fact that $ALL(out(v')) = ALL(out(v''))$, all evaluations arguments containing b need to be out in v'' as well. Thus, by the completeness of v'', it has to be the case that $v''(b^b) = in$, and we reach a contradiction.

We can now consider the case that $a = b^b$ is still a breaker argument, but $v'(a) = out$. This means there exists an evaluation argument $(F, G, B) \in A^{ev}$ s.t. $b \in F \cup G$ and $v'((F, G, B)) = in$. However, since $\overset{E}{\underset{V}{\bigcup}} in(v') = \overset{E}{\underset{V}{\bigcup}} in(v'')$, there has to exist an evaluation argument in $in(v'')$ containing b in the pd–set or pd–sequence as well. Therefore, $v''(b^b) = out$, and we reach a contradiction.

Let us now assume that $a = (F, G, B) \in A^{ev}$ is an evaluation argument and that $v'(a) = out$. Therefore, there is either an argument $b \in F$ s.t. $v'(b^b) = in$ or an evaluation argument $(F', G', B') \in in(v')$ s.t. $(F' \cup G') \cap B \neq \emptyset$. Based on the previous analysis, it suffices to focus on the latter case. Due to the fact that $\overset{E}{\underset{V}{\bigcup}} in(v') = \overset{E}{\underset{V}{\bigcup}} in(v'')$, for every argument $c \in F' \cup G'$ we can find an evaluation argument accepted in v'' that has this argument in the pd–sequence or pd–set. Consequently, we will be able to find a suitable attacker for (F, G, B) in $in(v'')$, and $v''(a) = out$. We reach a contradiction.

Let us now assume that $a = (F, G, B) \in A^{ev}$ is an evaluation argument and that $v'(a) = in$. Therefore, all evaluation arguments $(F', G', B') \in A^{ev}$ s.t. $(F' \cup G') \cap B \neq \emptyset$ need to be out in v' and all breaker arguments for elements in F (if they exist) need to be out w.r.t. v' as well. However, we can use previous analysis to show that v'' satisfies these requirements. Consequently, $v''(a) = in$, and we reach a contradiction. We can finally conclude that $v' = v''$ and that the relation between the complete labelings of D and the complete labelings of D^D_{lab} is one–to–one.

Let v a preferred labeling of D and v' the associated complete labeling in F^D_{lab} created using the previously described construction. If v' is not preferred in F^D_{lab}, it means there exists another complete labeling v'' s.t. $in(v') \subset in(v'')$ (see Definition 2.14). Let $a \in in(v'') \setminus in(v')$. Since we are working with AFs, we can observe that the set of arguments attacked by $in(v'')$ contains all the arguments attacked by $in(v')$. Therefore, due to the fact that both v' and v'' are complete, it has to be the case that $out(v') \subseteq out(v'')$. It is also easy to see that $v'(a) \neq out$.

Let us now assume that $a = b^b \in A^b$ is in fact a breaker argument for an argument $b \in A$. Since it is accepted in v'', it means that all of its attackers are mapped to out by v''. As the attackers are all evaluation arguments containing b in the pd–set or pd–sequence, it follows that $b \in ALL(out(v''))$. We can also observe that there has to be an evaluation argument containing b that is not attacked by $in(v')$; otherwise, v' would be capable of defending a and thus would have to accept a due to completeness. Therefore, it cannot be the case that $b \in ALL(out(v'))$. This, along with the fact that $out(v') \subseteq out(v'')$, means that $ALL(out(v')) \subset ALL(out(v''))$. Since $in(v') \subseteq in(v'')$, then $\overset{E}{\underset{V}{\bigcup}} in(v') \subseteq \overset{E}{\underset{V}{\bigcup}} in(v'')$ as well. Therefore, the complete labeling of D associated with v'' contains more information than v. Thus, v could not have been preferred in the first place and we reach a contradiction.

Let us now assume that $a = (F, G, B) \in A^{ev}$ is an evaluation argument. Due to the fact that $in(v') \subset in(v'')$, we can consider two cases; one where $\overset{E}{\underset{V}{\bigcup}} in(v') = \overset{E}{\underset{V}{\bigcup}} in(v'')$ and one where $\overset{E}{\underset{V}{\bigcup}} in(v') \subset \overset{E}{\underset{V}{\bigcup}} in(v'')$. If $\overset{E}{\underset{V}{\bigcup}} in(v') = \overset{E}{\underset{V}{\bigcup}} in(v'')$, then it means that for every standard argument $b \in F \cup G$, we can find an evaluation argument accepted in v'

containing b in its pd–set or pd–sequence. Therefore, every breaker argument for b (if it exists) is out w.r.t. both v' and v''. Hence, if v'' accepts a but v' does not, then there must exist another evaluation argument (F', G', B') attacking a (i.e. $(F' \cup G') \cap B \neq \emptyset$) which is out w.r.t. v'', but not w.r.t. v'. Using the previous analysis, we can therefore show that $B \subseteq ALL(out(v''))$ and $\exists c \in B$ s.t. $c \notin ALL(out(v'))$. Therefore, we can again show that the complete labeling associated with v'' contains more information than v – in this particular case, more $\mathbf{f}$ assignments. Hence, v cannot be preferred, and we reach a contradiction.

We are now left with the case where $\biguplus_{V}^{E} in(v') \subset \biguplus_{V}^{E} in(v'')$. Since $out(v') \subseteq out(v'')$ as well, then $ALL(out(v')) \subseteq ALL(out(v''))$. Consequently, the complete labeling associated with v'' is again more informative than v and we reach a contradiction. We can finally conclude that if v is a preferred labeling, then the labeling v' we constructed for it is preferred in F^D_{lab}.

Let us now assume we have an arbitrary preferred labeling v' of F^D_{lab}. Let v be its associated complete labeling of D. Due to the one–to–one relation between the complete labelings of D and F^D_{lab}, we can observe that v' follows the construction described at the beginning of this proof. We will now show that v is a preferred labeling of D. Assume it is not the case. This means there exists another complete labeling z of D that contains more information, i.e. $v \leq_i z$. Hence, there exists at least one argument $a \in A$ s.t. $v(a) = \mathbf{u}$ and $z(a) \neq \mathbf{u}$. We will denote the complete labeling of F^D_{lab} associated with z by z'.

Let us consider the case where $v(a) = \mathbf{u}$ and $z(a) = \mathbf{t}$. Since $v \leq_i z$, then we can show that the I^{ev} set associated with v' is a strict subset of the I^{ev} set associated with z'. Moreover, as $v^{\mathbf{f}} \subseteq z^{\mathbf{f}}$, then the O^{ev} set associated with v' is (not necessarily a strict) subset of the O^{ev} set associated with z'. Consequently, the same holds for the I^b sets. Thus, in total, $in(v') \subset in(z')$ and v' could not have been preferred in F^D_{lab}. We reach a contradiction.

Let us now assume that $v(a) = \mathbf{u}$ and $z(a) = \mathbf{f}$. Using the previous analysis, we can show that $in(v') \subseteq in(z')$ and that $out(v') \subseteq out(z')$. However, since $v^{\mathbf{f}} = ALL(out(v'))$ and $z^{\mathbf{f}} = ALL(out(z'))$ and a is contained in the latter, but not the former, then it has to be the case that the subset relation between $out(v')$ and $out(z')$ is in fact strict. Hence, by Theorem 2.17, v' could not have been a preferred labeling of F^D_{lab}. We reach a contradiction. Thus, we can finally conclude that if v' is a preferred labeling of F^D_{lab}, then v is a preferred labeling of D.

Let v be the grounded labeling of D and v' its associated complete labeling in F^D_{lab}. Let us assume v' is not grounded; this means there exists another complete labeling v'' of F^D_{lab} s.t. $in(v'') \subset in(v')$. Let $a \in in(v') \setminus in(v'')$. Since we are working with AFs, we can observe that the set of arguments attacked by $in(v')$ contains all the arguments attacked by $in(v'')$. Therefore, due to the fact that both v' and v'' are complete, it has to be the case that $out(v'') \subseteq out(v')$. It is also easy to see that $v''(a) \neq out$.

Let $a = b^b$ be in fact a breaker argument. If it is neither accepted nor rejected in v'', then due to completeness of v'' it means that no attacker of a in A' is in w.r.t. v'' and at least one is $undec$ w.r.t. v''. However, at the same time all such arguments need to be out w.r.t. v'. We can use the previous analysis to show that it has to be the case that $b \in ALL(out(v'))$, but $b \notin ALL(out(v''))$. As $out(v'') \subseteq out(v')$, it also holds that $ALL(out(v'')) \subset ALL(out(v'))$. Since $in(v'') \subset in(v')$, it is also the case that $\bigcup^{E}_{V} in(v'') \subseteq \bigcup^{E}_{V} in(v')$. Therefore, the complete labeling associated with v'' in D contains less information than v. Consequently, v cannot be a grounded extension of D, and we reach a contradiction.

Let now $a = (F, G, B) \in A^{ev}$ be an evaluation argument. Due to the fact that $in(v'') \subset in(v')$, we can consider two cases; one where $\bigcup^{E}_{V} in(v'') = \bigcup^{E}_{V} in(v')$ and one where $\bigcup^{E}_{V} in(v'') \subset \bigcup^{E}_{V} in(v')$.

If $\bigcup^{E}_{V} in(v'') = \bigcup^{E}_{V} in(v')$, then it means that for every standard argument $b \in F \cup G$, we can find an evaluation argument accepted in v'' containing b in its pd–set or pd–sequence. Therefore, every breaker argument for b (if it exists) is out both w.r.t. v'' and v'. Hence, if v' accepts a but v'' does not, then there must exist another evaluation argument (F', G', B') attacking a (i.e. $(F' \cup G') \cap B \neq \emptyset$) which is out w.r.t. v', but not w.r.t. v''. Using the previous analysis, we can therefore show that $B \subseteq ALL(out(v'))$ and $\exists c \in B$ s.t. $c \notin ALL(out(v''))$. Therefore, we can show that the complete labeling associated with v'' contains less information than v – in this particular case, less $\mathbf{f}$ assignments. Hence, v cannot be a grounded labeling of D, and we reach a contradiction.

We are now left with the case where $\bigcup^{E}_{V} in(v'') \subset \bigcup^{E}_{V} in(v')$. Since $out(v'') \subseteq out(v')$ as well, then $ALL(out(v'')) \subseteq ALL(out(v'))$. Consequently, the complete labeling associated with v'' is less informative than v and we reach a contradiction with v being grounded. We can finally conclude that if v is a grounded labeling, then the labeling v' we constructed for it is grounded in F^{D}_{lab}.

Let v' be the grounded labeling of F^{D}_{lab} and v its associated complete labeling of D. Due to the one–to–one relation between the complete labelings of D and F^{D}_{lab}, we can observe that v' follows the construction described at the beginning of this proof. Let us now assume that v is not grounded in D. This means there exists another complete labeling z of D that contains less information, i.e. $z \leq_i v$. Hence, there is at least one argument $a \in A$ s.t. $z(a) = \mathbf{u}$ and $v(a) \neq \mathbf{u}$. We will denote the complete labeling of F^{D}_{lab} associated with z by z'.

Let us consider the case where $z(a) = \mathbf{u}$ and $v(a) = \mathbf{t}$. Since $v \leq_i z$, then we can show that the I^{ev} set associated with z' is a strict subset of the I^{ev} set associated with v'. Moreover, as $z^{\mathbf{f}} \subseteq v^{\mathbf{f}}$, then the O^{ev} set associated with z' is (not necessarily a strict) subset of the O^{ev} set associated with v'. Consequently, the same holds for the I^{b} sets. Thus, in total, $in(z') \subset in(v')$ and v' could not have been the grounded labeling of F^{D}_{lab}. We reach a contradiction.

Let us now assume that $z(a) = \mathbf{u}$ and $v(a) = \mathbf{f}$. Using the previous analysis, we can show that $in(z') \subseteq in(v')$ and that $out(z') \subseteq out(v')$. However, since $v^{\mathbf{f}} = ALL(out(v'))$ and $z^{\mathbf{f}} = ALL(out(z'))$ and a is contained in the former, but not the latter, then it has to be the case that the subset relation between $out(z')$ and $out(v')$ is in fact strict. Hence, by Theorem 2.16, v' could not have been the grounded labeling of F^{D}_{lab}. We reach a contradiction. Thus, we can finally conclude that if v' is the grounded labeling of F^{D}_{lab}, then v is the grounded labeling of D. $\square$

Theorem 12.8. *Let $D = (A, C)$ be an ADF and $SF^D_{CC} = (A', R)$ its corresponding SETAF obtained from Translation 87. If $S \subseteq A$ is a conflict–free (cc–admissible, cc–complete, cc–preferred, grounded) extension of D, then there exists a conflict–free (admissible, complete, preferred, grounded) extension $S' = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ of SF^D_{CC} s.t. $S = \bigcup_{i=1}^n F_i \cup G_i$. If $S' = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ is a conflict–free (admissible, complete, preferred, grounded) extension of SF^D_{CC}, then $S = \bigcup_{i=1}^n F_i \cup G_i$ is conflict–free (cc–admissible, cc–complete, cc–preferred, grounded) extension of D.*

Proof. Let $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ be a conflict–free set of SF^D_{CC} and $T = \bigcup_{i=1}^n F_i \cup G_i$ its associated set in D. From the construction of the evaluations it follows that for every argument $a \in T$, there is at least one decisively in interpretation v_a s.t. $v_a^{\mathbf{t}} \subseteq T$ with which it entered any of the evaluations (F_i, G_i, B_i). Since S' is conflict–free in SF^D_{CC}, we can observe that $\bigcup_{i=1}^n F_i \cup G_i \cap \bigcup_{i=1}^n B_i = \emptyset$. Therefore, $T \cap v_a^{\mathbf{f}} = \emptyset$ as well. Therefore, $C_a(T \cap par(a)) = in$, and T is a conflict–free extension of D.

Let $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ be a conflict–free set of SF^D_{CC} and $T = \bigcup_{i=1}^n F_i \cup G_i$ its corresponding conflict–free extension in D. Let S^+ be the set of arguments in A' attacked by S and $X_S = \{a \in A \mid \forall (F, G, B) \in A^{ev}\, s.t.\, a \in F \cup G,\, (F, G, B) \in S^+\}$ be the collection of arguments in A s.t. all of the evaluations containing them are attacked by S. We will show that X_S corresponds to the standard discarded T^+ of T in D.

Let us assume that an argument $a \in A$ is in X_S, but not in T^+. By Lemma 2.125, this means that a has a standard evaluation (F', B') on A s.t. $B' \cap T = \emptyset$. Without the loss of generality, we can focus on minimal evaluations. (F', B') can be transformed into a partially acyclic evaluation that will appear as an argument in A^{ev}. Since $B' \cap T = \emptyset$, then there is no argument in S that would contain an element of B' in its pd–set or pd–sequence. Thus, it cannot be the case that S attacks this evaluation. Consequently, it is not contained in S^+, and as a result a cannot be in X_S. We reach a contradiction and therefore $X_S \subseteq T^+$.

Let us now assume there is an argument $a \in A$ which is in T^+, but not in X_S. Therefore, there is a partially acyclic pd–evaluation $(F', G', B') \in A^{ev}$ s.t. $a \in (F' \cup G')$ and $(F', G', B') \notin S^+$. From Translation 87 we can observe that arguments in A^{ev} are attacked by sets of size 1. Hence, we can simplify our analysis to attacks by single evaluation arguments. Based on the way T is constructed and the fact that there is no $(F'', G'', B'') \in S$ s.t. $(F'' \cup G'') \cap B' \neq \emptyset$, we can conclude that $T \cap B' = \emptyset$. As $(F' \cup G', B')$ is a standard evaluation for a, a could not have been in T^+ by Lemma 2.125. We reach a contradiction. Hence, $T^+ \subseteq X_S$, and when joined with our previous result, it holds that $X_S = T^+$.

Let $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ be an admissible extension of SF^D_{CC} and $T = \bigcup_{i=1}^n F_i \cup G_i$ its corresponding set conflict–free set in D. Since S is admissible in SF^D_{CC}, any evaluation containing an argument of any B_i has to be in S^+. Therefore, we can show that $\bigcup_{i=1}^n B_i \subseteq X_S$, and based on the previous analysis, $\bigcup_{i=1}^n B_i \subseteq T^+$.

Let $a \in T$. From the construction of T it holds that there is at least one evaluation $(F, G, B) \in S$ s.t. $a \in F \cup G$. Let v_a be the decisively in interpretation with which a entered (F, G, B). Based on the construction of the evaluation, we can observe that $v_a^{\mathbf{t}} \subseteq F \cup G$ and $v_a^{\mathbf{f}} \subseteq B$. Therefore, $v_a^{\mathbf{t}} \subseteq T$ and $v_a^{\mathbf{f}} \subseteq T^+$. This means that the range interpretation v_T is a completion of v_a. Consequently, v_T is a decisively in interpretation for a, and we can conclude that T has to be cc–admissible in D.

Let $S \subseteq A$ be a conflict–free set of D. This means that for every argument $a \in S$, $C_a(S \cap par(a)) = in$. We can therefore construct a trivial decisively in interpretation for a that assigns $\mathbf{t}$ to S and $\mathbf{f}$ to everything else. From this interpretation, we can extract a minimal one v_a. Hence, we can create a sound pd–function on S, and as for every v_a, $v_a^{\mathbf{t}} \subseteq S$, then for every argument we can construct a minimal partially acyclic evaluation on S. Due to the fact that $v_a^{\mathbf{f}} \cap S = \emptyset$, the blocking sets of our evaluations will be disjoint from S. By collecting them we obtain a set $E \subseteq A^{ev}$ in SF_{CC}^D which can be easily shown to be conflict–free in SF_{CC}^D. As all of the evaluations are on S and every argument in S possesses one, we can observe that the union of pd–sets and pd–sequences of elements in E will be equal to S. Therefore, E satisfies our requirements.

Let $S \subseteq A$ be a cc–admissible extension of D. Every argument $s \in S$ has at least one (minimal) standard evaluation on S whose blocking set is falsified by v_S. These pd–evaluations can be transformed into a partially acyclic ones and will appear as arguments in A^{ev}. Let $E = \{(F, G, B) \mid (F, G, B)$ is a minimal partially acyclic evaluation for $a \in S$ on S s.t. $B \subseteq S^+\}$ be the collection of such evaluations. By using the analysis above, we can show that E is conflict–free in SF_{CC}^D and that the union of the pd–sets and pd–sequences of the evaluations in E equals S. Moreover, we can repeat the previous parts of this proof in order to show that $S^+ = X_E$. Let us now assume that E is not admissible in SF_{CC}^D. This means it contains an evaluation argument (F, G, B) for which there exists an attacker $Z \in A'$ that is not attacked by (any subset of) E. If $Z \in A^b$, then by the Translation 87, $Z = F^b$. Since $\{(F, G, B)\}$ attacks Z and $(F, G, B) \in E$, E defends (F, G, B) from Z. Thus, it has to be the case that $Z \in A^{ev}$. If $Z = (F', G', B')$ is not in E^+, then the elements of $F' \cup G'$ are not in X_E. Therefore, they are not in S^+ either. Consequently, the blocking set of (F, G, B) could not have been contained in S^+ and we reach a contradiction with the construction of E. Thus, if S is cc–admissible in D, then so is E in SF_{CC}^D.

Let $S = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ be a complete extension of SF_{CC}^D and $T = \bigcup_{i=1}^n F_i \cup G_i$ its corresponding cc–admissible extension of D. Let us assume T is not cc–complete in D. This means there exists an argument $a \notin T$ which is decisively in w.r.t. v_T. It is possible to represent T as a single standard (and thus, also partially acyclic) evaluation covering all elements in T and with a blocking set in T^+. Let (F, G, B) s.t. $F \cup G = T$ and $B \subseteq T^+$ be this evaluation. Since a is decisively in w.r.t. v_T, we can extract from it a minimal decisively in interpretation v_a for a. We can observe that $v_a^{\mathbf{t}} \subseteq T$ and $v_a^{\mathbf{f}} \subseteq T^+$. We can now extend G with a and B with $v_a^{\mathbf{f}}$ in order to obtain a partially acyclic evaluation for a. Again, from this evaluation we can extract a minimal one (F', G', B') that will appear as an argument in A^{ev}. Due to the fact that a can be added to the pd–sequence, $F' \subseteq F$. Since all arguments in F' appear in some evaluations in S, it is easy to see that the breaker argument for F' will be attacked by a subset of S. Additionally, $B' \subseteq T^+$, and therefore $B' \subseteq X_S$. This means that every partially acyclic evaluation of every argument in B' is attacked by S. We can therefore conclude that S defends (F', G', B'), and as this evaluation could not have been included in S (otherwise, a would have appeared in T), we reach a contradiction with the completeness of S. We can thus conclude that if S is complete in SF^D_{CC}, then T is cc–complete in D.

Let $S \subseteq A$ be a cc–complete extension of D and $E \subseteq A^{ev}$ its corresponding admissible extension of SF^D_{CC}. Let us assume that E is not complete in SF^D_{CC}. This means there exists an argument $Z \in A' \setminus E$ that is defended by E. We can observe that all arguments from A^b are self–attackers. Defending them would breach the conflict–freeness of E. Consequently, it has to be the case that $Z = (F, G, B) \in A^{ev}$. Based on the SETAF Fundamental Lemma (Lemma 2.22) we can observe that $E' = E \cup \{Z\}$ is an admissible extension of SF. Let S' be the cc–admissible extension of D corresponding to E'. We can now consider two situations: one in which $S = S'$ and one in which $S \neq S'$.

Let us focus on the first case. By the construction of E, this means that there is an argument $a \in B$ not contained in S^+. Consequently, by Lemma 2.125 there exists a (minimal) standard evaluation for a with a blocking set disjoint from S. We can transform this pd–evaluation into a partially acyclic one (F', G', B') that will appear as an argument in A^{ev}. By using the previous analysis and the relation between S^+, X_E and E^+, we can show that $\{(F', G', B')\}$ attacks Z but is not attacked by E in return. Thus, E could not have defended Z and we reach a contradiction.

Let us now focus on the second case. Since $E \subset E'$ and $S \neq S'$, we can show that $S \subset S'$. Hence, there is least one argument $a \in F \cup G$ which is not in S and does not have an evaluation in E. If $a \in F$, then we can observe that no subset of E could have attacked the breaker argument $F^b \in A^b$ for Z. This contradicts our assumption that E defends Z. Hence, a has to be contained in the pd–sequence and $F \subseteq S$. Let $G = (g_0, ..., g_n)$ and let v_i be the decisively in interpretation with which g_i entered in Z. Assume that $g_0 \notin S$. We can observe that $v_0^{\mathbf{t}} \subseteq F \subseteq S$ and $v_0^{\mathbf{f}} \subseteq B$. However, since g_0 is not decisively in w.r.t. v_S, there is an argument $b \in v_0^{\mathbf{f}}$ s.t. $b \notin S^+$. Therefore, by using the previous analysis we can show that there exists a partially acyclic evaluation containing b that is not in E^+. This means that E cannot defend Z and we reach a contradiction. Let us therefore assume that $g_0 \in S$ and it is g_1 that is missing from our extension. By the definition of partially acyclic evaluation, $v_1^{\mathbf{t}} \subseteq F \cup \{g_0\}$ and $v_1^{\mathbf{f}} \subseteq B$. Hence, $v_1^{\mathbf{t}} \subseteq S$, and the lack of decisiveness has to be due to insufficient coverage of B by S^+. We can therefore again construct an appropriate attacker for Z in A' and show that E could not have defended it. We can continue in this manner till we reach g_n and the conclusion that E did not defend Z, which contradicts our assumption. Thus, if S is cc–complete in D, then E is complete in SF^D_{CC}.

Let now $S \subseteq A$ be a cc–complete extension of D. We will now show that that there is exactly one complete extension in SF^D_{CC} corresponding to it. Assume it is not the case and let $E = \{(F_1, G_1, B_1), ..., (F_n, G_n, B_n)\} \subseteq A^{ev}$ and $E' =$

$\{(F'_1, G'_1, B'_1), ..., (F'_n, G'_n, B'_n)\} \subseteq A^{ev}$ be two complete extensions of SF^D_{CC} s.t. $S = \bigcup_{i=1}^{n} F_i \cup G_i = \bigcup_{i=1}^{n} F'_i \cup G'_i$ and $E \neq E'$. This means there exists an argument $Z \in A^{ev}$. s.t. $Z \in E \setminus E'$ or $Z \in E' \setminus E$.

Let us focus on the first case. If $Z = (F, G, B)$ is in E, but not in E', then by the completeness of both of the extensions, it means that E defends Z and E' does not. Since both of the extensions correspond to the same set in D, for any argument $a \in F \cup G$ we can find an evaluation argument in E' containing it. Thus, we can show that E' is capable for attacking the breaker argument F^b for Z (assuming it exists). Hence, if E' does not defend Z, then the attacker is an evaluation argument. We can repeat the analysis of the discarded sets in order to show that this again cannot be the case. We thus reach a contradiction and can conclude that $E \subseteq E'$. We can do the same for the other direction and show that $E = E'$, which contradicts our assumption that these two extensions are different. Hence, the relation between the complete extensions of D and SF^D_{CC} is one to one.

Let S, S' be two cc–complete extensions of D and E, E' the associated complete extensions in SF^D_{CC}. We will show that $S \subset S'$ iff $E \subset E'$.

Let us focus on showing that if $S \subset S'$, then $E \subset E'$. By using Lemma 2.124, we can show that every argument in the standard discarded set of S is in the discarded set of S'. For this reason we can also observe that no argument in $S' \setminus S$ can be in S^+. Therefore, the standard range of S' is a completion of S. Consequently, for any minimal partially acyclic evaluation (F, G, B) on S for an argument in $a \in S$ s.t. $F \cup G \subseteq S$ and $B \subseteq S^+$, it holds that $F \cup G \subseteq S'$ and $B \subseteq S'^+$. Based on the construction of E and E' and the one to one relation between the complete extensions of D and SF^D_{CC}, this means that $E \subseteq E'$. Since S' contains an argument not included in S, then E' must contain an evaluation for this argument which is not in E either. Thus, $E \subset E'$.

The other direction is quite straightforward. If $E \subset E'$, then clearly $S \subseteq S'$. However, due to the one–to–one relation between the complete extensions of D and SF^D_{CC}, it cannot be the case that $E \subset E'$ and $S = S'$. Hence $S \subset S$, and this concludes our proof.

Due to the fact that $S \subset S'$ iff $E \subset E'$ and that the relation between the complete extensions of both frameworks is one–to–one, we can therefore use Theorems 2.10 and 2.158 in order to prove the relations between the preferred and grounded extensions of D and SF^D_{CC} stated in the theorem. This concludes our proof. □

Theorem 12.10. *Let $D = (A, C)$ be a weakly valid ADF and $SF^D_{AA} = (A, R')$ its associated SETAF created through Translation 88. If $E \subseteq A$ is pd–acyclic conflict–free (aa–admissible) in D, then it is conflict–free (admissible) in SF^D_{AA}. Not every conflict–free (admissible) extension of SF^D_{AA} is pd–acyclic conflict–free (aa–admissible) in D. $E \subseteq A$ is an aa–complete (aa–preferred, acyclic grounded, stable) extension of D iff it is a complete (preferred, grounded, stable) extension of SF^D_{AA}.*

Proof. Let $E \subseteq A$ be a pd–acyclic conflict–free extension of D. Consequently, every argument $a \in A$ possesses an acyclic pd–evaluation (F, B) s.t. $B \cap E = \emptyset$. This means that for no subset $E' \subseteq E$, E' has the power to block all evaluations of a through the blocking set. Therefore, for no such subset E' and argument a, $E'Ra$. Consequently, E is conflict–free in SF_{AA}^D.

Let $E \subseteq A$ be aa–admissible in D and let E^{a+} be its acyclic discarded set. Based on the previous paragraph we know that E is conflict–free in SF_{AA}^D. From Lemma 2.128 and the construction of R it is easy to see that $E^{a+} = E^+$, where E^+ is the discarded set of E in SF_{AA}^D (see Definition 2.25). Every $a \in E$ has to have an acyclic pd–evaluation (F, B) s.t. $B \subseteq E^{a+}$. We can observe that every set of arguments attacking a in SF_{AA}^D will have at least one element in common with B. From the discarded set analysis it follows that every set of arguments attacking a in SF_{AA}^D will have at least one element attacked by E in R. Consequently, every $a \in E$ is defended by E and E is admissible in SF_{AA}^D.

Let us now consider a trivial framework $(\{a, b\}, \{C_a = \top, C_b = a\})$. The associated SETAF is $(\{a, b\}, \emptyset)$ and it produces $\{b\}$ as a conflict–free and admissible extension. Unfortunately, $\{b\}$ is neither pd–acyclic conflict–free nor aa–admissible in our ADF.

Let $E \subseteq A$ be an aa–complete extension of D. Based on the previous parts of this proof we can observe that E is admissible in SF_{AA}^D and that the discarded sets of E are the same in both frameworks. Let us assume that E is not complete in SF_{AA}^D. This means there exists an argument $a \in A \setminus E$ defended by E. Consequently, there exists an acyclic pd–evaluation (F, B) for a s.t. $B \subseteq E^{a+}$; if it were not the case, we could always construct an attacking set in R not covered by E^+. Since $a \notin E$ and $B \subseteq E^{a+}$, then it means that there is an argument in the pd–acyclic sequence (different than a) that is not in E. Let $F = (a_0, ..., a_n)$ be our sequence and let us focus on a_0. Due to the construction of pd–acyclic sequences, the decisively in interpretation for a_0 can only consist of $\mathbf{f}$ mappings. As these mappings are in the acyclic discarded set, then a_0 has to be decisively in w.r.t. the acyclic range of E in D. Therefore, it has to be the case that $a_0 \in E$. Let us analyze a_1. The $\mathbf{t}$ part of the decisively in interpretation for a_1 that was used in the construction of (F, B) is a subset of $\{a_0\}$ and thus a subset of E as well, while the $\mathbf{f}$ part is contained in the acyclic discarded set. Consequently, a_1 is also decisively in w.r.t. the acyclic range of E in D. Thus, it has to be the case that $a_1 \in E$. We can continue reasoning in this manner until we reach the conclusion that $a = a_n \in E$, which contradicts our assumptions. Therefore, if E is aa–complete in D, then it is complete in SF_{AA}^D.

Let $E \subseteq A$ be a complete extension of SF_{AA}^D. By the construction of SF_{AA}^D, we can observe that if an argument $a \in A$ is defended by E, then it has an acyclic pd–evaluation (F, B) s.t. $B \subseteq E^+$. If it were not the case and for every such evaluation we could find an argument in the blocking set not included in the discarded one, then the collection of such arguments would contain an attacking set on a that E does not defend a from. However, since from (F, B) we can extract an acyclic pd–evaluation for any of the members in the sequence and any such evaluation will have a blocking set in E^+, we can show that if E defends a, then it defends every other argument F. Therefore, by the completeness of E, if $b \in E$, then it has an acyclic pd–evaluation (F', B') s.t. $F' \subseteq E$ and $B' \subseteq E^+$. By the conflict–freeness of E it also means that $F' \cap B' = \emptyset$. Consequently, for any argument $b \in E$, there exists an acyclic pd–evaluation for b on E that is not blocked by E. Thus, E is pd–acyclic conflict–free. We can now exploit the relation between the discarded sets to show that E is also aa–admissible. Finally, let us assume that E is not aa–complete in SF^D_{AA}. In other words, there is an argument $c \in A \setminus E$ which is decisively in w.r.t. the acyclic range. Thus, it possesses an acyclic pd–evaluation (F'', B'') with $F'' = (a_0, ..., a_n)$ s.t. $\{a_0, ..., a_{n-1}\} \subseteq E$ and $B'' \subseteq E^+$. However, by using the previous parts of this analysis we can clearly show that c is defended by E in SF^D_{AA} and thus reach a contradiction with our assumptions. Therefore, E is aa–complete in D.

Based on the correspondence between the complete extensions of both structures, we can use the discarded set analysis and Theorems 2.23, 2.24, 2.138 and 2.158 in order to prove the relation between the preferred (grounded, stable) extensions of both frameworks. □

Proposition 12.13. *Let $SF = (A, R)$ be a SETAF and $E, E' \subseteq A$ two admissible extensions of SF. If there are no $b \in E, B' \subseteq E'$ s.t. $B'Rb$ and no $b' \in E', B \subseteq E$ s.t. BRb', then $E \cup E'$ is admissible in SF.*

Proof. Let us assume that $X = E \cup E'$ is not conflict–free. This means there exists $x \in X$, $X' \subseteq X$ s.t. $X'Rx$. If $X' \subseteq E$ and $x \in E$ (resp. $X' \subseteq E'$ and $x \in E'$), we violate the conflict–freeness of E (resp. E'). If $X' \subseteq E'$ and $x \in E$ (or $X' \subseteq E$ and $x \in E'$) we violate our assumptions. Thus, we are left with the case that one "part" of X is in E and the other in E'. Without the loss of generality, let us assume that the argument attacked by X is in E. Due to the admissibility of E, it has to be the case that there is $Z \subseteq E$ and $z \in X$ s.t. ZRz. Moreover, due to conflict–freeness of E, $z \notin E$. Therefore, $z \in E'$, and we reach a contradiction with our assumptions. We can thus conclude that X is conflict–free. Showing that it defends all of its members follows easily from the admissibility of E and E'. Hence, X is admissible in SF. □

Theorem 12.17. *Let $D = (A, C)$ be a cleansed form ADF and $FN^D_{AA} = (A', R', N')$ its corresponding AFN obtained through Translation 89. Then, FN^D_{AA} is in minimal and (strongly) consistent normal forms. FN^D_{AA} might not be weakly valid if D is weakly valid. If D is relation valid, then FN^D_{AA} is weakly and relation valid. If D is strongly valid, then so is FN^D_{AA}.*

Proof. Let $(a, v_a) \in A'$ be an argument in FN^D_{AA} created for an argument $a \in A$ in D. Based on the construction of FN^D_{AA}, we can observe that every set of arguments supporting (a, v_a) in N' consists of arguments created for precisely one argument from $v^{\mathbf{t}}_a$. Consequently, all of the sets supporting (a, v_a) in N' are completely disjoint and incomparable. Hence, FN^D_{AA} is trivially in minimal normal form. It is also easy to see that $v^{\mathbf{t}}_a \cap v^{\mathbf{f}}_a = \emptyset$. Therefore, an argument (c, v_c) created for an argument $c \in v^{\mathbf{f}}_a$ cannot appear in any supporting set of (a, v_a). Thus, FN^D_{AA} satisfies strong consistency requirements.

This part of the proof concerning the validity normal forms is strongly related to the proof of Theorem 12.18. Let us now assume that we are dealing with a weakly valid D. This means that every argument $a \in A$ possesses an acyclic pd–evaluation on A. Given such an evaluation $((a_0, ..., a_n), B)$ and the pd–function it was created with, we can create a sequence $((a_0, pd_E^D(a_0)), ..., (a_n, pd_E^D(a_n)))$ of arguments in A'. This sequence will satisfy the powerful requirements (see proof of Theorem 12.18) in FN_{AA}^D. Thus, we can show that every argument $(a, v_a) \in A'$ s.t. v_a is a decisively in interpretation used in the construction of an acyclic pd–evaluation for a on A in D, will possess a powerful sequence on A' in FN_{AA}^D. The question therefore is, what happens to the arguments paired with interpretations that are not used. Since D is weakly valid, A can be represented as an acyclic pd–evaluation containing all arguments in A in its pd–sequence. This evaluation induces a corresponding powerful sequence built from arguments $E \subseteq A'$. Let $(a, v_a) \in A' \setminus E$ be an AFN argument not in this sequence. We can observe that $v_a^{\mathbf{t}} \subseteq A$ and therefore for any argument $b \in v_a^{\mathbf{t}}$, there exists an argument $(b, v_b) \in E$ for some v_b. Consequently, we can extend the powerful sequence representing E by (a, v_a) and still obtain a powerful sequence (see proof of Theorem 12.18). This procedure can be repeated for all arguments from $A' \setminus E$. Hence, for every argument in A' we can create a powerful sequence on A'. Therefore, FN_{AA}^D is weakly valid.

Let us move on to relation validity. With $coh(X)$ we will denote the collection of all coherent sets on $X \subseteq A'$; by $Arg(a)$ we will denote all arguments in A' representing an argument $a \in A$, i.e. $Arg(a) = \{(a, v_a) \mid v_a \in min\_dec(in, a)\}$. Let D be relation valid. Since it is also cleansed, then by Lemma 4.77, D is weakly valid. By the previous parts of this proof, this means that FN_{AA}^D is weakly valid as well.

However, let us assume that FN_{AA}^D is not relation valid. Therefore, there exists an argument $(a, v_a) \in A'$, a set of arguments $E \subseteq A$ s.t. $EN'(a, v_a)$ and an argument $(b, v_b) \in E$ s.t. $(b, v_b) \notin \bigcup coh(A' \setminus \{(a, v_a)\})$, where $coh(X)$ denotes the set of all coherent subsets on $X \subseteq A'$. Due to the fact that FN_{AA}^D is weakly valid, $(b, v_b) \in \bigcup coh(A'\})$. Since $(b, v_b) \notin \bigcup coh(A' \setminus \{(a, v_a)\})$, it has to be the case that every powerful sequence for (b, v_b) requires the presence of (a, v_a).

Let us assume that $Arg(a) = \{(a, v_a)\}$, i.e. (a, v_a) is the only representation of a. Therefore, v_a is the only minimal decisively in interpretation for a. Moreover, since $(b, v_b) \in E$, then $v_a(b) = \mathbf{t}$. Due to the fact that D is relation valid, b possesses an acyclic pd–evaluation s.t. v_b was used in its construction. Based on the previous analysis, we can show that every acyclic pd–evaluation for b that uses v_b contains a in its pd–sequence. Since v_a depends on b, it is not possible that we can create an acyclic pd–evaluation for b that uses v_b. We reach a contradiction with the relation validity of D.

Let us assume that there exists at least one alternative representation (a, v'_a) for a. Since D is relation valid, v'_a can be used in constructing an acyclic pd–evaluation for a. From this it also follows that there is a powerful sequence for (a, v'_a) s.t. the arguments preceding it are contained in $A' \setminus Arg(a)$, i.e. the sequence does not depend on other representations for a. Consequently, this powerful sequence will not use (a, v_a). Since every powerful sequence for (b, v_b) depends on (a, v_a), this means that (a, v_a) cannot be replaced by (a, v'_a). Hence, we can show that every powerful sequence for (a, v'_a) depends on (b, v_b) – if it were not the case, we would have been able to recombine the sequences for (b, v_b) and (a, v'_a). We can now distinguish two cases; one where (b, v_b) is the only representation for b and one in which there are more representations.

If (b, v_b) is the only argument in FN_{AA}^D for b, then it means that every representation of a depends on b, while every evaluation for b using v_b depends on a. This means that we cannot construct an acyclic pd–evaluation for b using this interpretation, which breaches the relation validity of D. We can therefore consider the case in which b has more than representation; let (b, v'_b) be one of them. Since D is relation valid, we can show that there exists a powerful sequence for this argument s.t. the arguments preceding (b, v'_b) are contained in $A' \setminus Arg(b)$. If we cannot replace the presence of (b, v_b) in any of the powerful sequences for (a, v'_a) with (b, v'_b), then this means that (b, v'_b) has to depend on (a, v'_a). We can continue this analysis and reach the conclusions that every representation of b comes back to a and every representation of a comes back to b. This not only breaches relation validity, but also weak validity of D. We reach a contradiction and can therefore conclude that if D is relation valid, then so is FN_{AA}^D.

Let D be strongly valid. By Theorem 4.44, we can create a sequence $(a_0, ..., a_n)$ consisting of all arguments in A s.t. independently of the chosen pd–function, we can create an acyclic pd–evaluation with $(a_0, ..., a_n)$ as its pd–sequence. For each argument a_i, let $seq_i = ((a_i, v_{a_i}^1), ..., (a_i, v_{a_i}^{n_i}))$ be an arbitrary sequence consisting of all of its associated arguments in A'. Since D is in cleansed form, this sequence will not be empty. Let seq be the sequence of all elements in A' obtained by merging the sequences from a_0 to a_n (in this order). We will now show that independently of the chosen support function f in the terms of Definition 4.30, this sequence will be a powerful sequence w.r.t. f.

Due to the fact that $(a_0, ..., a_n)$ is a pd–sequence independently of the chosen pd–function, every minimal decisively in interpretation for a_0 has an empty $\mathbf{t}$ part (see proof of Theorem 4.44). Therefore, no argument in A' corresponding to a_0 will require support through N' based on the construction of FN_{AA}^D. Hence, every support function f will assign to such an A' argument $\emptyset$, and all arguments in seq_0 will meet the requirements of a starting argument of a powerful sequence. Thus, seq_0 is a powerful sequence independently of the chosen f.

Let us now focus on a_1 and seq_1. Every minimal decisively in function for a_1 will have a $\mathbf{t}$ part contained in $\{a_0\}$ (see proof of Theorem 4.44). Consequently, based on the construction of FN_{AA}^D, every set $S \in suf((a_1, v_{a_1}^j))$ of an argument $(a_1, v_{a_1}^j)$ in seq_1 is a subset of $\{(a_0, v_{a_0}^1), ..., (a_0, v_{a_0}^{n_0})\}$. Consequently, independently of the chosen f, the merged sequence consisting of seq_0 and seq_1 (in this order), will be a powerful sequence.

We can continue on in this way till we reach seq_n and the conclusion that the produced sequence is a powerful sequence independently of the chosen support function f. Since this sequence covers all arguments in A', then by Theorem 4.33, FN_{AA}^D is strongly valid. $\square$

Theorem 12.18. *Let $D = (A, C)$ be a cleansed form ADF and $FN_{AA}^D = (A', R', N')$ its corresponding AFN obtained through Translation 89. If $S = \{(a_1, v_{a_1}), ..., (a_n, v_{a_n})\} \subseteq A'$ is a coherent (strongly coherent, admissible, preferred, complete, grounded, stable) extension of FN_{AA}^D, then $S' = \bigcup_{i=1}^n \{a_i\}$ is a pd–acyclic (pd–acyclic conflict–free, aa–admissible, aa–preferred, aa–complete, acyclic grounded, stable) extension of D.*

If $S' \subseteq A$ is a pd–acyclic (pd–acyclic conflict–free, aa–admissible, aa–preferred, aa–complete, acyclic grounded, stable) extension of D, then there exists a coherent (strongly coherent, admissible, preferred, complete, grounded, stable) extension $S = \{(a_1, v_{a_1}), ..., (a_n, v_{a_n})\} \subseteq A'$ of FN^D_{AA} s.t. $S' = \bigcup_{i=1}^{n}\{a_i\}$.

Proof. We will start the proof by analyzing the correspondence between the acyclic pd–evaluations in D and powerful sequences in FN^D_{AA}.

Let $((a_0, ..., a_n), B)$ be an acyclic pd–evaluation on a set $E \subseteq A$ for an argument $a \in A$ and pd^D_E the pd–function it was created with. We can observe that for every a_i in the sequence, $(a_i, pd^D_E(a_i))$ will appear as an argument in A'. Let $((a_0, pd^D_E(a_0)), ..., (a_n, pd^D_E(a_n)))$ be a sequence of arguments in FN^D_{AA} associated with our evaluation. We will show that it is powerful on $\bigcup_{i=0}^{n}(a_i, pd^D_E(a_i))$ in FN^D_{AA}. By the pd–sequence requirements, $pd^D_E(a_0)^{\mathbf{t}} = \emptyset$. Consequently, by Translation 89, there will be no set of arguments $V \subseteq A'$ s.t. $VN(a_0, pd^D_E(a_0))$. Thus, $(a_0, pd^D_E(a_0))$ satisfies the requirements of a starting argument in a powerful sequence. Let now $(a_i, pd^D_E(a_i))$, where $1 \leq 1 \leq n$, be an arbitrary element in our sequence. By Translation 89, every $V \subseteq A'$ s.t. $VN(a_i, pd^D_E(a_i))$ is generated for some argument $c \in pd^D_E(a_i)$. By this and the pd–sequence requirement that $pd^D_E(a_i)^{\mathbf{t}} \subseteq \{a_0, ..., a_{i-1}\}$, it is easy to see that for every $V \subseteq A'$ s.t. $VN(a_i, pd^D_E(a_i))$, $V \cap \{(a_0, pd^D_E(a_0)), ..., (a_{i-1}, pd^D_E(a_{i-1}))\} \neq \emptyset$. Thus, the powerful conditions are satisfied and we can conclude that the presented sequence is powerful in FN^D_{AA}.

Let now $seq = ((b_0, v_{b_0}), ..., (b_m, v_{b_m}))$ be a powerful sequence on a set $S \subseteq A'$ for an argument $(b_m, v_{b_m}) \in S$. Due to the fact that a given argument in A can have multiple representations, it can happen that $b_i = b_j$ even though $i \neq j$. The difference lies in the choice of the accompanying decisively in interpretation. This duplication can also lead to the use of decisively in interpretations in a powerful sequence that would never be used in an acyclic pd–evaluation, as seen in Example 148. However, we will show that we can still extract an acyclic pd–evaluation from $((b_0, v_{b_0}), ..., (b_m, v_{b_m}))$. We will start by "purging" seq.

Let seq' be a subsequence of seq built in the following way. First, add (b_0, v_{b_0}) to seq'. It is easy to see that seq' is a powerful sequence. Let us now consider (b_1, v_{b_1}). Let now (b_i, v_{b_i}) be the first argument in seq s.t. $b_i \neq b_0$. Based on the construction of FN^D_{AA} and the properties of a powerful sequence, we can observe that $v^{\mathbf{t}}_{b_i} \subseteq \{b_0, ..., b_{i-1}\}$. However, since $b_0 = ... = b_{i-1}$, then $v^{\mathbf{t}}_{b_i} \subseteq \{b_0\}$. Therefore, $\{(b_0, v_{b_0})\}$ will still have an element in common with every set supporting (b_i, v_{b_i}) in N', assuming they even exist. Consequently, we can add (b_i, v_{b_i}) to seq' and seq' will still be a powerful sequence. We can now find the first argument in seq that represents an argument different from b_0 and b_i. Let us assume that it is (b_j, v_{b_j}). It is easy to see that it appears later in the sequence than (b_i, v_{b_i}). We can repeat the previous line of reasoning in order to show that seq' with (b_j, v_{b_j}) added is still a powerful sequence. We can continue in this manner until there are no further arguments to add to seq'. We thus obtain a powerful subsequence of seq s.t. for every $i \neq j$, $b_i \neq b_j$, but for every (b_k, v_{b_k}) in seq we can find $(b_p, (v_{b_p}))$ in seq' s.t. $b_p = b_k$, even though v_{b_k} and v_{b_p} do not have to be the same. Moreover, we can observe that the last argument in seq

and seq' will correspond to the same argument in D. To the subsequence of a powerful sequence of FN^D_{AA} obtained in this manner we will refer to as the purged subsequence.

Let now $seq = ((b_0, v_{b_0}), ..., (b_m, v_{b_m}))$ be a powerful sequence on a set $S \subseteq A'$ for an argument $(b_m, v_{b_m}) \in S$. Let $seq' = ((a_0, v_{a_0}), ..., (a_k, v_{a_k}))$ be the purged subsequence of seq. Based on the parts of the proof showing that seq' is still a powerful sequence, we can show that $((a_0, ..., a_k), \bigcup_{i=0}^{k} v^{\mathbf{f}}_{a_i})$ is an acyclic pd–evaluation for a_k on the set $\{a \mid \exists(c, v_c) \in S \text{ s.t. } c = a\}$ in D.

Based on the relations between the powerful sequences and evaluations, we can show that if $S \subseteq A$ is a pd–acyclic set of D, then we can find a set $S' = \{(a_0, v_{a_0}), ..., (a_n, v_{a_n}))\} \subseteq A'$ s.t. $S = \bigcup_{i=0}^{n}\{a_n\}$. Furthermore, we can also show that if $S' = \{(a_0, v_{a_0}), ..., (a_n, v_{a_n}))\} \subseteq A'$ is a coherent set of FN^D_{AA}, then $S = \bigcup_{i=0}^{n}\{a_n\}$ is pd–acyclic in D.

Let $S \subseteq A$ be a pd–acyclic conflict–free set of D. Every argument $a \in S$ possesses at least one acyclic pd–evaluation (F, B) on S s.t. $B \cap S = \emptyset$. Let pd^D_S be the pd–function with which (F, B) was created. For every $b \in F$ we can create a pair (b, pd^D_S) that will appear as an argument in A' in FN^D_{AA}. By collecting all such pairs induced by all the evaluations meeting our requirements we can create a set $S' \subseteq A$' which, based on the previous parts of this proof, will be coherent in FN^D_{AA}. In every evaluation we had used, the blocking set was disjoint from S. Moreover, the $\mathbf{f}$ part of any interpretation that appeared in the induced argument–interpretation pairs was contained in a blocking set of an evaluation that we picked. Consequently, we can show that there are no elements $c, d \in S'$ s.t. $cR'd$. Therefore, S' is strongly coherent in FN^D_{AA}.

Let $S' = \{(a_0, v_{a_0}), ..., (a_n, v_{a_n})\} \subseteq A'$ be a strongly coherent set of FN^D_{AA} and $S = \bigcup_{i=0}^{n}\{a_i\}$ the associated set in D. By using previous parts of this proof, we can show that S is coherent. Since S' is conflict–free in FN^D_{AA}, this means that for no arguments $(a_i, v_{a_i}), (a_j, v_{a_j}) \in S'$ it is the case that $v_{a_j}(a_i) = \mathbf{f}$. We can therefore use previous analysis and show that for every argument in S we can find an acyclic pd–evaluation on S s.t. its blocking set is disjoint from S. Hence, S is a pd–acyclic conflict–free extension of D.

Let $S \subseteq A$ be an aa–admissible extension of D. We can observe that for every argument $a \in S$ there exists an acyclic pd–evaluation (F, B) on S s.t. $B \subseteq S^{a+}$. Using the previously described approach, we can extract argument–interpretation assignments from these evaluations in order to construct a strongly coherent set $S' \subseteq A'$ in FN^D_{AA}. Let us assume that this set is not admissible in FN^D_{AA}; this means there exists an argument $(b, v_b) \in S'$, $(c, v_c) \in A'$ s.t. $(c, v_c)R'(b, v_b)$ and a coherent set $V \subseteq A'$ containing (c, v_c) that is not attacked by S'. We can observe that $v_b(c) = \mathbf{f}$ based on the construction of FN^D_{AA}. From V we can extract a powerful sequence for (c, v_c) and then purge it in order to obtain a sequence for (c, v'_c). The pair (c, v'_c) will still be attacking (b, v_b) in R'. From the purged sequence we can create an acyclic pd–evaluation for c with a blocking set disjoint from S. Therefore, $c \notin S^{a+}$. However, since $v^{\mathbf{f}}_b \subseteq S^{a+}$ and $v_b(c) = \mathbf{f}$, we reach a contradiction with our construction. Therefore, we can conclude that S' is an admissible extension of FN^D_{AA}.

Let $S' = \{(a_0, v_{a_0}), ..., (a_n, v_{a_n}))\} \subseteq A'$ be an admissible set of FN^D_{AA}. By the previous parts of this proof we know that the associated set $S = \bigcup_{i=0}^{n}\{a_i\}$ is pd–acyclic conflict–free in D. Let us assume it is not aa–admissible. This means there exists an argument $b \in S$ that is not decisively in w.r.t. the acyclic range of S in D. Hence, for every decisively in interpretation v_b for b s.t. $v_b^{\mathbf{t}} \subseteq S$, there exists an argument $c \in v_b^{\mathbf{f}}$ that is not falsified in the acyclic range. This includes all interpretations for b that appeared with b as a paired argument in S'. If c is not falsified in the acyclic range, then by Lemma 2.128 it means it possess an acyclic pd–evaluation (F, B) on A s.t. $B \cap S = \emptyset$. By using the previous analysis we can observe that this evaluation will induce a corresponding powerful sequence in A'. We can also see that none of the arguments $a_0, ..., a_n$ are in B and B corresponds to the $\mathbf{f}$ parts of the interpretations of arguments in F. Therefore, based on the construction of FN^D_{AA}, it cannot be the case that S' attacks the powerful sequence created from (F, B). Since we can construct such an attacker related to any decisively in interpretation for b that could have been used in S', we can conclude that S' does not defend one of its arguments. We reach a contradiction with the admissibility of S' in FN^D_{AA}. Hence, it has to be the case that S is aa–admissible in D.

Let $S \subseteq A$ be an aa–complete extension of D. Let $S' = \{(a, v_a) \mid a \in S$ and there exists a pd–function pd on S and an acyclic pd–evaluation (F, B) for a on S created with it s.t. $B \subseteq S^{a+}$ and $pd(a) = v_a\}$ be a set of arguments in FN^D_{AA} created in the same fashion as in the admissible case. Although S' is admissible in FN^D_{AA}, it does not have to be complete (see Example 148). We will therefore use an extended construction.

Let $E = \{(a, v_a) \mid a \in S, v_a^{\mathbf{t}} \subseteq S, v_a^{\mathbf{f}} \subseteq S^{a+}\}$ be the set argument–interpretation pairs s.t. the acyclic range interpretation is a completion of every interpretation. We can observe that $S' \subseteq E$. Since $S \cap S^{a+} = \emptyset$ by Lemma 2.128, then for no two $(a, v_a), (b, v_b) \in E$ it can be the case that $v_b(a) = \mathbf{f}$. Therefore, we can show that E is conflict–free in FN^D_{AA}. Let us now show it is coherent. By using the previous parts of the proof, we only need to show that the arguments in $E \setminus S'$ possess a powerful sequence on E. We can observe that S' can be represented as a single powerful sequence. Let now (a, v_a) be an argument in $E \setminus S'$. Due to the fact that both S' and E correspond to S, we can observe that for any argument $b \in v_a^{\mathbf{t}}$, we can find an argument $(b, v_b) \in S'$. Consequently, E has an element in common with every set of arguments supporting (a, v_a) in N'. We can therefore append (a, v_a) to the powerful sequence representing S' in order to obtain a powerful sequence for (a, v_a) on E. Hence, we can show that E is coherent in FN^D_{AA}. We can repeat the previous parts of this proof in order to show that E is also admissible in FN^D_{AA}.

Let us now assume that E is not complete in FN^D_{AA}. This means there exists an argument $(a, v_a) \in A'$ that is defended by E. Based on the coherence requirements of defense and the construction of FN^D_{AA}, we can therefore observe that for every argument $b \in v_a^{\mathbf{t}}$, there exists an argument $(b, v_b) \in E$. We can also observe that for any argument $(c, v_c) \in A'$ s.t. $v_a(c) = \mathbf{f}$, every powerful sequence for (c, v_c) is attacked by E. We can therefore show that every acyclic pd–evaluation for c is blocked through the blocking set by S in D. Consequently, c is assigned $\mathbf{f}$ in the acyclic range interpretation (see Lemma 2.128). Therefore, $v_a^{\mathbf{t}} \subseteq S$ and $v_a^{\mathbf{f}} \subseteq S^{a+}$. If $a \in S$, we reach a contradiction with the construction of E. If $a \notin E$, we reach a contradiction with aa–completeness of S in D. Hence, we can conclude that if S is aa–complete in D, then E is complete in FN^D_{AA}.

Let now $S' \subseteq A'$ be a complete extension of FN^D_{AA} and S its associated aa–admissible set in D. Assume that S is not aa–complete; this means there exists an argument $a \in A \setminus S$ that is decisively in w.r.t. the acyclic range interpretation v^a_S of S. We can therefore observe that v^a_S is a completion of a minimal decisively in interpretation v_a for a. Due to the fact that $v^{\mathbf{t}}_a \subseteq S$, then for any argument $b \in v^{\mathbf{t}}_a$, we can find an argument $(b, v_b) \in S'$. Therefore, we can show that $S' \cup (a, v_a)$ is coherent in FN^D_{AA}. Now, due to the fact that $v^{\mathbf{f}}_a \subseteq S^{a+}$, then every acyclic pd–evaluation for every argument $c \in v^{\mathbf{f}}_a$ is blocked through the blocking set by S. Thus, we can reuse the previous parts of the proof to show that every powerful sequence for argument $(c, v_c) \in A'$ is attacked in S'. Hence, S' defends (a, v_a), and as $(a, v_a) \notin S'$ (otherwise a would have been in S), we reach a contradiction with the completeness of this set in FN^D_{AA}. Hence, if S' is complete in FN^D_{AA}, then S is aa–complete in D.

Let $S \subseteq A$ be an aa–complete extension of D. Let us assume there exist two different complete extensions $E, E' \subseteq A'$ of FN^D_{AA} s.t. $\{a \mid (a, v_a) \in E\} = \{b \mid (b, v_b) \in E'\}$ (i.e. they both correspond to S). Therefore, there exists an argument $(a, v_a) \in A'$ s.t. $(a, v_a) \in E \setminus E'$ or $(a, v_a) \in E' \setminus E$. Let us focus on the first case. Due to the completeness of E', it must be the case that it does not defend (a, v_a). Based on the construction of FN^D_{AA} and the fact that both sets correspond to the same set of arguments in D, we can show that if $E \cup (a, v_a)$ is coherent, then so is $E' \cup (a, v_a)$. Consequently, if E' does not defend (a, v_a), there must exist an argument (c, v_c) with a powerful sequence on A' unattacked by E' s.t. $v_a(c) = \mathbf{f}$. In other words, there is a powerful sequence $((c_0, v_{c_0}), ..., (c_n, v_{c_n}))$ for (c, v_c) s.t. for no argument $d \in \bigcup_{i=0}^{n} v^{\mathbf{f}}_{c_i}$, there is an element $(d, v_d) \in E'$. However, since $\{a \mid (a, v_a) \in E\} = \{b \mid (b, v_b) \in E'\}$, then it cannot be the case that E attacks this sequence for (c, v_c). Consequently E cannot defend (a, v_a) and we reach a contradiction. We can conclude that every argument in E is contained in E'. We can adapt this proof to show that every argument in E' is contained in E as well. This means that $E = E'$ and we reach a contradiction with our assumptions. Hence, the relation between the aa–complete extensions of D and the complete extension of FN^D_{AA} is one–to–one.

Let now $S, S' \subseteq A$ be two aa–complete extensions of D and $E, E' \subseteq A'$ their associated complete extensions of FN^D_{AA}. We will show that $S \subset S''$ iff $E \subset E'$.

Let S be a subset of S'. By using Lemma 2.128, we can show that the acyclic range of S' is a completion of the acyclic range of S. Consequently, if the acyclic range of S is a completion of a decisively in interpretation for an argument $a \in S$, then so is the range of S'. Hence, $E \subseteq E'$. We can observe that S' contains at least one argument b not present in S. Therefore, there will be an argument $(b, v_b) \in E'$ that is not in E, and we can conclude that $E \subset E'$.

Let E be a subset of E'. Therefore, $\{a \mid (a, v_a) \in E\} \subset \{b \mid (b, v_b) \in E'\}$. From this follows that $S \subseteq S'$. However, based on the proved one–to–one relation between the complete extensions of D and FN^D_{AA}, we can see it cannot be the case that $S = S'$. Thus, it holds that $S \subset S'$.

By using the one–to–one and monotonicity relations between the complete extensions of D and FN^D_{AA} and Theorems 2.95 and 2.158, we can show that the preferred and grounded extensions of D and FN^D_{AA} are related in the way stated in our theorem. $\square$