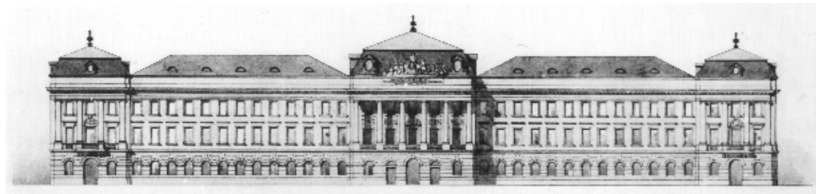
dbai

TECHNICAL REPORT



INSTITUT FÜR INFORMATIONSSYSTEME

ABTEILUNG DATENBANKEN UND ARTIFICIAL INTELLIGENCE

Preliminary Report on Complexity Analysis of Extension–Based Semantics of Abstract Dialectical Frameworks

DBAI-TR-2017-103

Sylwia Polberg and Johannes P. Wallner

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

DBAI TECHNICAL REPORT
2017

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

DBAI TECHNICAL REPORT

DBAI TECHNICAL REPORT DBAI-TR-2017-103, 2017

Preliminary Report on Complexity Analysis of Extension–Based Semantics of Abstract Dialectical Frameworks

Sylwia Polberg [1] and Johannes P. Wallner [2]

Abstract. The point of this report is to provide a preliminary analysis of the computational complexity of the extension–based semantics for abstract dialectical frameworks [Pol15]. We focus on four types of problems – verification, existence, skeptical and credulous reasoning. We also provide complexity results for the basic notions used in the construction of our semantics, such as the decisive interpretations, various types of discarded sets and evaluations.

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

[2]Institute for Information Systems 184/2, Technische Universität Wien, Austria.

Acknowledgements: We would like to thank Thomas Linsbichler and Stefan Woltran for their help, valuable comments and proofreading of this work. This research was funded by projects FWF I2843 and FWF I1102. Sylwia Polberg is a member of the Vienna PhD School of Informatics and is currently supported by EPSRC Project EP/N008294/1 "Framework for Computational Persuasion". Johannes P. Wallner is currently supported by project FWF I2854.

Contents

1 Introduction

The point of this report is to provide a preliminary analysis of the computational complexity of the extension–based semantics for abstract dialectical frameworks [Pol15]. We focus on four types of problems – verification, existence, skeptical and credulous reasoning. We also provide complexity results for the basic notions used in the construction of our semantics, such as the decisive interpretations, various types of discarded sets and evaluations.

2 Abstract Dialectical Frameworks

Abstract dialectical frameworks have been defined in [BW10] and till today various results as to their semantics, instantiation and complexity have already been published in [BES$^+$13, PWW13, Str13a, Str13b, SW15]. The main goal of ADFs is to be able to express arbitrary relations between arguments 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.

Definition 1. 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.

One can also represent the acceptance conditions by propositional formulas over arguments instead of "boolean" functions [Ell12]. 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.

2.1 Preliminaries

2.1.1 Interpretations and decisiveness

Interpretations will be equally important both in labeling and extension–based semantics. In the first case the semantics produce interpretations instead of extensions (sets of arguments). In the latter, accepting given arguments can cause rejecting others and interpretations are used to store this data.

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. In the labeling–based approach, the values are compared w.r.t. precision (information) ordering: $\mathbf{u} \leq_i \mathbf{t}$ and $\mathbf{u} \leq_i \mathbf{f}$. 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 than v, denoted $v \leq_i v'$, iff $\forall_{s \in A}\, v(s) \leq_i v'(s)$. We can define the meet of interpretations in a similar manner. In case $v \leq_i v'$ and v is three and v' two–valued, we say that v' extends v. Extending an interpretation can also be seen as replacing the $\mathbf{u}$ assignments with $\mathbf{t}$ and $\mathbf{f}$. The set of all two–valued interpretations extending v is denoted $[v]_2$. We will use v^x to denote a set of arguments mapped to x by v, where x a truth–value. Given an acceptance condition C_s for some 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_s and an interpretation v defined over all of the atoms of the formula, $v(\varphi_s)$ will just stand for the value of the formula under v.

The notion of decisiveness is a key concept in the extension–based semantics for abstract dialectical frameworks. Basically speaking, a (partial) interpretation is decisive for an argument if no new information will cause the outcome of the acceptance condition to change. For example, given a condition $\varphi_s = a \wedge \neg b$ for some 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 some 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. Let $D = (A, C)$ be an ADF, $E \subseteq A$ a set of arguments and v a two–valued interpretation on E. A **completion** of v to a set Z where $E \subseteq Z$, is an interpretation v' defined on Z in a way that $\forall a \in E\ v(a) = v'(a)$. v' is an x–completion of v, where $x \in \{\mathbf{t}, \mathbf{f}\}$, iff all arguments in $Z \setminus E$ are mapped to x.

Remark 1. By the abuse of notation we will also talk about **u**–completions when comparing extension and labeling–based approaches. It should be understood as a three–valued interpretation that assigns **u** to the "missing" mappings of a given two–valued interpretation.

Remark 2. 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, we will use the completion notion in order not to confuse extending interpretations with extensions, i.e. sets of arguments.

Definition 3. 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.

By $dec(x, a)$, where $x \in \{in, out\}$, we will denote the set of all decisively x interpretations for an argument $a \in A$.

Remark 3. In the definition we can actually use any set of completions, not just the one to $E \cup par(s)$ – using ones to A would be equally fine. The only important thing is that the set of completions contains every completion defined for at least $par(s)$.

Example 1. 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 an ADF. 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 disjunction, and thus the acceptance condition, are satisfied. Formally speaking, v_1 is decisively in for b as both of its completions $\{c : \mathbf{t}, a : \mathbf{f}\}$ and $\{c : \mathbf{t}, a : \mathbf{t}\}$ satisfy the condition.

2.1.2 Evaluations and acyclicity

Let $D = (A, C)$ be an ADF and $a \in A$ an argument. The acceptance condition of a can tell us how and on what other arguments a depends. We can see if they need to be accepted or rejected for the condition to be in our out and can derive a range of decisively in interpretations for a based on this information. We can then move our attention from a to the arguments in the condition and investigate them in a similar manner. Repeating this process until no new arguments show up provides us with a full picture telling us when, how, and if at all, the argument a can be accepted or rejected, if it can be derived from initial arguments, is it based on cyclic dependencies and so on. In order to track this procedure we have introduced the notions of positive dependency functions and evaluations [Pol14, Pol15] [1].

In majority of the argumentation frameworks, the nature of a relation between the arguments is stated openly in the structure of the framework, e.g. R is the attack, N is the support and so on. This is not the case in ADFs and in order to say what is the nature of the link between two arguments, we need to analyze the acceptance condition. In our approach, 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. We will focus only 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 $x \in \{in, out\}$. Explanation for this choice will be given at the end of this section.

Let us now 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 4. 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 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 is maximally sound on E iff it is a sound function on $E' \subseteq E$ and there is no sound positive dependency function pd' on E'', where $E' \subset E'' \subseteq E$, s.t. $\forall a \in E', pd(a) = pd'(a)$.

[1] 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.

We will now trace the arguments that a given argument requires for its acceptance by the use of dependency evaluations. Standard evaluations just hold the required and undesired arguments in sets. While they are quite useful, we will also be interested in the more specialized types that deal with positive dependency cycles. The informal understanding of a cycle is simply whether acceptance of an argument depends on this argument. Partially acyclic evaluations can be seen as refinement of the standard ones, where the set of required arguments is further split into two parts: 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. The last type of evaluations, the acyclic ones, are a subclass of partially acyclic. We simply require that the "cycle container" is empty.

Definition 5. Let $D = (A, C)$ be an ADF, $X \subseteq A$ a set of arguments 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. $e \in F$, $\forall a \in F\, pd_E^D(a)^{\mathbf{t}} \subseteq F$, 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.

Definition 6. Let $D = (A, C)$ be an ADF, $X \subseteq A$ a set of arguments 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 $F \subseteq E$ and $B \subseteq A$ are sets of arguments and $(a_0, ..., a_n)$ is a sequence of distinct elements of E satisfying the requirements:

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

Definition 7. 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 $F = \emptyset$.

We will use the shortened notation $((a_0, ..., a_n), B)$ in order to denote the acyclic evaluations. 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. 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.

There are two ways we can "attack" an evaluation. Either we accept an argument that the evaluation rejects or we discard an argument that the evaluation needs. This leads to the following formulation: [2]

Definition 8. 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** $(F, (a_0, ..., a_n), B)$ if $\exists b \in B$ s.t. $v(b) = \mathbf{t}$ or $\exists x \in \{a_0, ..., a_n\} \cup F$ s.t. $v(x) = \mathbf{f}$. A set of arguments $X \subseteq A$ blocks $(F, (a_0, ..., a_n), B)$ if $X \cap B \neq \emptyset$.

Remark 4. 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 2. 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 1. 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 $(\{e\}, \emptyset)$. Let us now consider partially acyclic evaluations. 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 standard evaluation for d looks structurally similar to the one of e, 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.

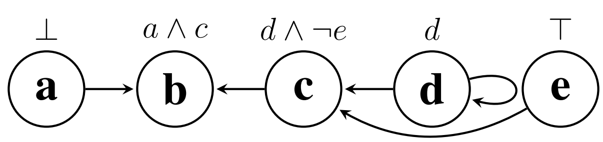


Figure 1: Sample ADF

We will close this section with a discussion on why we require minimal interpretations in our construction. Allowing every type of interpretation would not affect our semantics, as we are mostly interested in the existence of an unblocked evaluation of a given type or in blocking all evaluations. Existence of an unblocked evaluation built with an arbitrary interpretation implies

[2] 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.

existence of one built with minimal ones – we can always "remove" unnecessary elements from the blocking set in order to trim it to minimal interpretations. And 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. there is no evaluation that cannot be made acyclic (see Definition 47), minimality makes a difference.

Example 3. 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.

2.2 Labeling–Based Semantics

There are two main families of labeling–based (or three–valued) semantics for ADFs: the ultimate [BES$^+$13] and approximate ones [Str13a]. In this section we will focus only on the first, which are based on a characteristic operator:

Definition 9. 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_A. The **three–valued characteristic operator** of D is a function $\Gamma_D : V_A \to V_A$ s.t. $\Gamma_D(v) = v'$ with $v'(a) = \bigsqcap_{w \in [v]_2} w(\varphi_a)$.

The labeling–based semantics are now as follows:

Definition 10. 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. We say that 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; and
- a **grounded** labeling of D iff it is the least fixpoint of Γ_D.

The stable semantics will be described in Section 2.3.3.

Example 4. 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 2. Let us check if $v_1 = \{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{u}, d : \mathbf{u}\}$ is an admissible labeling. It has in total 4 extending interpretations: $[v_1]_2 = \{\{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{t}, d : \mathbf{t}\}, \{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{t}, d : \mathbf{f}\}, \{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{f}, d : \mathbf{t}\}, \{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{f}, d : \mathbf{f}\}\}$. All of them satisfy the condition of a and none of them satisfies the condition of b. Consequently, the interpretation computed by the operator is at least as informative as v_1 and v_1 is admissible. We can however observe that under these assignments, the condition of c is also never satisfied, while the outcome of d depends on the extending interpretation. Therefore, v_1 is not complete, but $v_2 = \{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{f}, d : \mathbf{u}\}$ is – again, the condition of d changes and thus the argument can only be assigned $\mathbf{u}$. Nevertheless, v_2 is not preferred – the interpretations $v_3 = \{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{f}, d : \mathbf{f}\}$ and $v_4 = \{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{f}, d : \mathbf{t}\}$ are both admissible and contain more information than v_2. As acceptance of all of the arguments depends on other arguments, it is not surprising that our grounded labeling consists only of $\mathbf{u}$ assignments: $\{a : \mathbf{u}, b : \mathbf{u}, c : \mathbf{u}, d : \mathbf{u}\}$.

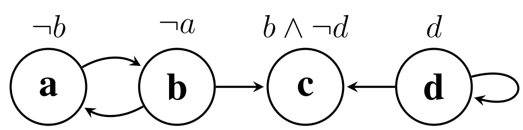


Figure 2: Sample ADF

2.3 Extension–Based Semantics

In [Pol14, Pol15] 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. We will now recall them briefly and refer the reader to the original work for proofs and further explanations.

2.3.1 Conflict–free Semantics

The conflict–free extensions of ADFs represent "arguments that can stand together" [BG09]. In other words, conflict–freeness stands for satisfying the acceptance conditions.

Definition 11. 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 $C_s(E \cap par(s)) = in$.

The pd–acyclic version of conflict–freeness needs to take into account also the attacks on the evaluation level, as seen in the following example:

Example 5. Let us now look at the ADF $(\{a, b\}, \{C_a : \top, C_b : \neg a \vee b\})$ depicted in Figure 3. The conflict–free extensions are $\emptyset, \{a\}, \{b\}$ and $\{a, b\}$. We can observe that the last one cannot intuitively be pd–acyclic conflict–free, as the presence of a forced a self–support cycle on b. Both arguments possess an acyclic pd–evaluation: $((a), \emptyset)$ for a and $((b), \{a\})$ for b. Please note that the decisively in interpretation $\{b : \mathbf{t}\}$ cannot be used to construct an acyclic pd–evaluation for b. The evaluation for b is blocked by the set $\{a, b\}$, even though the acceptance condition of b still is satisfied.

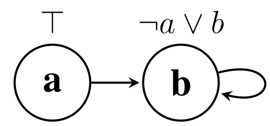


Figure 3: Sample ADF

Definition 12. Let $D = (A, C)$ be an ADF. A conflict–free extension $E \subseteq A$ of D is a **pd–acyclic conflict–free extension** of D iff for every argument $a \in E$, there exists an acyclic pd–evaluation $((a_0, ..., a_n), B)$ on E s.t. $B \cap E = \emptyset$.

2.3.2 Ranges and Discarded Sets

Admissibility in ADFs is based on the concept of range, which in turn in its original version required conflict–freeness. We will recall it here and later show that with the use of evaluations, we can drop the conflict–freeness assumption. The basic concept of range is based on decisive outing. We start with 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 13. 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. for every $a \in E$ set $v_E(a) = \mathbf{t}$;
2. for every argument $b \in A \setminus v_E^{\mathbf{t}}$ that is decisively out w.r.t. v_E, set $v_E(b) = \mathbf{f}$
3. now repeat the previous step until there are no new mappings are added to v_E.

The resulting v_E is the **range (interpretation)** of E. The **discarded set** of E is defined as $E^+ = v_E^{\mathbf{f}}$.

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 discarded by our set without the conflict–freeness assumption.

Lemma 14. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a set of arguments and $X = \{a \in A \mid E$ blocks every standard pd–evaluation for a in $D\}$. If E is conflict–free, then $X = E^+$.*

The standard notions of the discarded set and 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, as depending on the approach we might not treat cyclic arguments as valid[3]:

Definition 15. Let $D = (A, C)$ be an ADF, $E \subseteq A$ a conflict–free extension of D and v^a_E a partial two–valued interpretation built as follows:

1. for every $a \in E$ set $v^a_E(a) = \mathbf{t}$.
2. for every argument $b \in A \setminus (v^a_E)^{\mathbf{f}}$ s.t. every acyclic pd–evaluation of b in A is blocked by v^a_E, set $v^a_E(b) = \mathbf{f}$.

The resulting v^a_E is the **acyclic range (interpretation)** of E. The **acyclic discarded set** of E is defines as $E^{a+} = (v^a_E)^{\mathbf{f}}$.

We can rephrase this definition in the following way, similar to Lemma 14:

Lemma 16. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a set of arguments and $X = \{a \in A \mid E$ blocks every acyclic pd–evaluation for a in $D\}$. If E is conflict–free, then $X \setminus E = E^+$. If E is pd–acyclic conflict–free, then $X \cap E = \emptyset$.*

The last type of range – the partially acyclic one – can be seen as a certain middle ground between the standard and acyclic range. We discard an argument if we block all of its acyclic pd–evaluations unless it is based on a "cycle" that we are ready to accept.

Definition 17. Let $D = (A, C)$ be an ADF, $E \subseteq A$ a set of arguments and $X = \{a \in A \mid$ there is no partially acyclic evaluation (F, G, B) for a in D s.t. $B \cap E = \emptyset$ and $F \subseteq E\}$. If E is conflict–free, then the **partially acyclic discarded set** of E is $E^{p+} = X$ and the **partially acyclic range (interpretation)** is v^p_E s.t. $(v^p_E)^{\mathbf{t}} = E$ and $(v^p_E)^{\mathbf{f}} = X$.

The three versions of the discarded set can be ordered w.r.t. $\subseteq$:

Lemma 18. *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 $E^{p+} = E^{a+}$.*

Finally, the following properties of ranges and the discarded sets will be useful to us in the context of this work:

Lemma 19. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a conflict–free set of arguments and v a two–valued interpretation s.t. $v^{\mathbf{t}} = E$. If every argument in $v^{\mathbf{f}}$ is decisively out w.r.t. v, then the partially acyclic range v^a_E of E is a completion of v. For any conflict–free set E there is an interpretation v with $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} \subseteq E^{p+}$ s.t. all arguments in $v^{\mathbf{f}}$ are decisively out w.r.t. v.*

[3]This is an updated definition and different from [Pol14]. Please see [Pol15] for explanation.

Lemma 20. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a conflict–free set of arguments and $X \subseteq A \setminus E$ a set of arguments. Let v be a two–valued interpretation s.t. $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} = X$. Then v is the partially acyclic range interpretation of E iff all arguments in X are decisively out w.r.t. v and every $x \in A \setminus (E \cup X)$ has a partially acyclic evaluation (F^x, G^x, B^x) s.t. $B^x \cap E = \emptyset$ and $F^x \subseteq E$.*

Lemma 21. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a conflict–free set of arguments and v a two–valued interpretation s.t. $v^{\mathbf{t}} = E$. If every argument in $v^{\mathbf{f}}$ is decisively out w.r.t. v, then the acyclic range v_E^a of E is a completion of v. If E is pd–acyclic conflict–free, then there is an interpretation v with $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} \subseteq E^{a+}$ s.t. all arguments in $v^{\mathbf{f}}$ are decisively out w.r.t. v.*

Lemma 22. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a pd–acyclic conflict–free set of arguments and $X \subseteq A \setminus E$ a set of arguments. Let v be a two–valued interpretation s.t. $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} = X$. Then v is the acyclic range interpretation of E iff all arguments in X are decisively out w.r.t. v and every $x \in A \setminus (E \cup X)$ has an acyclic evaluation (F^x, B^x) s.t. $B^x \cap E = \emptyset$.*

Example 6. Let us consider the framework $(\{a, b, c, d, e\}, \{C_a : a \wedge \neg b, C_b : a, C_c : \neg b, C_d : \neg a, C_e : d, C_f : f\})$ depicted in Figure 4 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 partial 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.

2.3.3 Model and Stable Semantics

The concept of a model basically follows the intuition that if something can be accepted, it should be accepted. It was meant as a basis for the stable semantics [BW10]. Due to the problems of the initial definition of stability, a different reduct–based method using grounded labelings was proposed [BES$^+$13]. In [Pol14, Pol15] alternative methods were introduced.

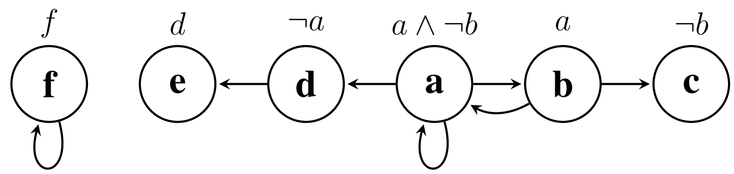


Figure 4: Sample ADF

Definition 23. 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 $s \in E$.

Definition 24. Let $D = (A, C = \{\varphi_a\}_{a \in A})$ be an ADF and $E \subseteq A$ a set of arguments. The reduct of D w.r.t. E is a framework $D^E = (E, C^E)$, where for $e \in E$ we set $C_e^E = \varphi_e[b/\mathbf{f} : b \notin E]$.

Definition 25. Let $D = (A, C)$ be an ADF, $M \subseteq A$ a model of D and $D^M = (M, C^M)$ the reduct of D w.r.t. M. Let gv be the grounded model of D^M. Model M is **stable** in D iff $M = gv^{\mathbf{t}}$.

Theorem 26. *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.*

Lemma 27. *Let $D = (A, C)$ be an ADF. A set $E \subseteq A$ is a stable extension of D iff it is a pd–acyclic conflict–free extension of D s.t. $E^{a+} = A \setminus E$.*

2.3.4 Grounded Semantics

The extension–based version of grounded semantics has already been introduced in [BW10]. It's equivalent formulation is as follows (see [Pol15] for more details).

Proposition 28. *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 29. 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 $v^{\mathbf{t}}$.

2.3.5 Admissible, Preferred and Complete Semantics

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 as in the usual argumentation semantics– admissibility represents a defensible stand, preferred extensions are maximal admissible, and completeness accepts whatever is defended. By replacing defense with decisiveness w.r.t. range, we basically obtain the ADF semantics. The difference between the approaches lies in the type of range we use and if acyclicity of the extension is also desired.

Definition 30. Let $D = (A, C)$ be an ADF and $E \subseteq A$ a set of arguments. E is:

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

Just like in pd–acyclic conflict–free semantics we had to check that there are no conflicts arising on the level of evaluations, in the case of aa and ac–admissible semantics we add the "defense" of evaluations. It cannot be handled by decisiveness alone, the same way acyclicity could not have been ensured by the outcome of the condition only. Please note that in the original definition of these semantics decisiveness was also required [Pol14]. However, it is subsumed by defending the evaluation: see [Pol15] for discussion.

Definition 31. Let $D = (A, C)$ be an ADF. A set $E \subseteq A$ is an **x–preferred** extension of D, where $x \in \{cc, aa, ac, ca_1, ca_2\}$, iff it is a maximal (w.r.t. set inclusion) x–admissible extension of D.

Definition 32. Let $D = (A, C)$ be an ADF, $E \subseteq A$ an x–admissible extension of D, where $x \in \{cc, aa, ac, ca_2\}$, and v_E^x its appropriate range. E is an x–complete extension of D if every $a \in A$ that is decisively in w.r.t. v_E^x is in E. A ca$_1$–admissible extension E is ca$_1$–complete if every $a \in A \setminus E^{a+}$ that is decisively in w.r.t. v_E^a is in E.

2.3.6 Properties of Extension–Based Semantics

In this section we briefly recall the properties of the ADF semantics from [Pol15] that will be useful to us:

Proposition 33. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a standard conflict–free 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:*

1. *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.*
2. *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.*
3. *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.*
4. *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 34. *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 35. ***CC/AC/AA/CA$_2$ Fundamental Lemma**: Let $D = (A, C)$ be an ADF, $E \subseteq A$ an x–admissible extension of D where $x \in \{cc, aa, ac, ca_2\}$, v_E^x its appropriate range interpretation and $a, b \in A$ two arguments decisively in w.r.t. v_E^x. Then $E' = E \cup \{a\}$ is x–admissible in D and b is decisively in w.r.t. $v_{E'}^x$.*

Lemma 36. ***Weak CA$_1$ Fundamental Lemma**: Let $D = (A, C)$ be an ADF, $E \subseteq A$ a ca$_1$–admissible extension, v_E^a 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 $A \setminus E'^{a+}$.*

Theorem 37. *Let $D = (A, C)$ be an ADF and $x \in \{cc, ac, aa\}$. The following holds:*

1. *The sets of all x–admissible extensions of D forms a complete partial order w.r.t. set inclusion.* [4]
2. *D possesses at least one x–preferred extension.*
3. *For each x–admissible set E of D, there exists an x–preferred extension E' of D s.t. $E \subseteq E'$.*

Theorem 38. *Let $D = (A, C)$ be an ADF. The following holds:*

[4] There are many definitions of complete partial orders. We will assume that a partial order $(A, \leq)$ is a complete iff it has a least element and each of its directed subsets has a lub.

1. *Every xy–preferred extension of D is an xy–complete extension of D for $x, y \in \{a, c\}$, but not vice versa.*

2. *The grounded extension of D is the least w.r.t. set inclusion ac and cc–complete extension of D.*

3. *The acyclic grounded extension of D is the least w.r.t. set inclusion aa–complete extension of D and a minimal ca$_1$ and ca$_2$–complete extension of D.*

4. *The cc, ac and aa–complete extensions of D form complete meet–semilattices w.r.t. set inclusion.*

Theorem 39. *Let $D = (A, C)$ be an ADF s.t. A is finite. Given that $x \in \{ca_1, ca_2\}$, D possesses at least one x–preferred extension and for every x–admissible set E of D, there exists an x–preferred extension E' of D s.t. $E \subseteq E'$.*

Proposition 40. *There exists an ADF $D = (A, C)$ s.t. :*

1. *The grounded extension of D is neither an aa, ca$_1$ nor a ca$_2$–complete extension of D.*

2. *The ca$_1$ and ca$_2$–complete extensions of D do not form complete meet–semilattices w.r.t. set inclusion.*

2.4 Comparison of Extension–Based and Labeling–Based Semantics

Definition 41. 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 $v^{\mathbf{t}} = E$.

Please note that the correspondence relation between the extensions and labelings, if it exists, is in general not bijective.

Theorem 42. *Let $D = (A, C)$ be an ADF and $x \in \{cc, aa, ac, ca_2\}$. The following holds:*

1. *Let $E \subseteq A$ be an x–admissible extension of D and v_E^x its appropriate range. The $\mathbf{u}$–completion of v_E^x to A is an admissible labeling of D.*

2. *Not for every ca$_1$–admissible extension of D there exists a corresponding admissible labeling of D.*

3. *If v is an admissible labeling of D, then $v^{\mathbf{t}}$ is a ca$_1$ and ca$_2$–admissible extension of D.*

Theorem 43. *Let $D = (A, C)$ be an ADF. The following holds:*

1. *Let $E \subseteq A$ be an aa, ac, cc or ca$_1$–preferred extension. There might not exist a corresponding preferred labeling of D.*

2. *Let $E \subseteq A$ be an ca$_2$–preferred extension. The* $\mathbf{u}$*–completion of* v^p_E *to* A *is a preferred labeling of* D*.*

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

Theorem 44. *Let* $D = (A, C)$ *be an ADF and* $x \in \{cc, aa, ac, ca_2\}$*. The following holds:*

1. *Let* $E \subseteq A$ *be an x–complete extension of* D *and* v^x_E *its appropriate range. The* $\mathbf{u}$*–completion of* v^x_E *is a complete labeling of* D*.*

2. *Not every complete labeling has a corresponding complete extension of any type.*

3. *For every complete labeling* v *of* D*,* $v^{\mathbf{t}}$ *is contained in some ca$_2$–complete extension.*

4. *Not every ca$_1$–complete extension of* D *has a corresponding complete labeling.*

As the grounded semantics has a very clear meaning, it is no wonder that both available approaches coincide, as already noted in [BES$^+$13].

Theorem 45. *Let* $D = (A, C)$ *be an ADF. The two–valued grounded extension of* D *and the grounded labeling of* D *correspond.*

However, in the acyclic grounded case, the best we can get is that it has a complete labeling. It will of course not be the least one, since that corresponds to the standard grounded semantics.

Theorem 46. *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*.*

Last, but not least, we will describe a subclass of ADFs for which our classification system collapses. By this we understand that all xy–subsemantics of a given semantics coincide, e.g. every aa–admissible extension is cc–admissible and so on. 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.

Definition 47. Let $D = (A, C)$ be an ADF. D is an AADF$^+$ iff every standard evaluation in the framework can be made acyclic.

Theorem 48. *Let* $D = (A, C)$ *be an AADF$^+$. The following holds:*

1. *Every conflict–free extension of* D *is pd–acyclic conflict–free*

2. *Every model of* D *is stable*

3. *Given a conflict–free set of arguments* $E \subseteq A$ *of* D*,* $E^+ = E^{p+} = E^{a+}$

4. *The aa/cc/ac/ca$_1$/ca$_2$–admissible extensions of D coincide*
5. *The aa/cc/ac/ca$_1$/ca$_2$–complete extensions of D coincide*
6. *The aa/cc/ac/ca$_1$/ca$_2$–preferred extensions of Dcoincide*
7. *The grounded and acyclic grounded extensions coincide*

Theorem 49. *Let $D = (A, C)$ be an AADF$^+$. The following holds:*

1. *Every admissible labeling of D has a corresponding aa/ac/cc/ca$_1$/ca$_2$–admissible extension and vice versa.*
2. *Every complete labeling of D has a corresponding aa/ac/cc/ca$_1$/ca$_2$–complete extension and vice versa.*
3. *Every preferred labeling of D has a corresponding aa/ac/cc/ca$_1$/ca$_2$–preferred extension 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 7. 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$^+$.

3 Computational Complexity

Complexity theory concerns itself with analyzing how difficult it is to solve a given problem. Such a problem is described in terms of input and a question to be answered. Depending on how the question is formulated, we distinguish various types of problems. If the answer is supposed to be yes or no, we deal with the decision type; this is also what we will focus on in this work. In this report we assume that the reader is familiar with the basics of complexity theory and classes such as P, NP and coNP [Pap94]. However, we will also go beyond these classes and primarily deal with those that belong to the polynomial hierarchy:

Definition 50. Let $\Sigma_0^{\mathsf{P}} = \Pi_0^{\mathsf{P}} = \Delta_0^{\mathsf{P}} = P$, where P is the set of decision problems solvable in polynomial time. For $i \geq 1$, we define:

- $\Delta_i^{\mathsf{P}} = P^{\Sigma_{i-1}^{\mathsf{P}}}$,
- $\Sigma_i^{\mathsf{P}} = NP^{\Sigma_{i-1}^{\mathsf{P}}}$,
- $\Pi_i^{\mathsf{P}} = coNP^{\Sigma_{i-1}^{\mathsf{P}}}$,

where P^A is the set of decision problems solvable in polynomial time with an access to an A–oracle. Classes NP^A and $coNP^A$ are defined analogously.

In particular, we can observe that $\Sigma_1^{\mathsf{P}} = NP$ and $\Pi_1^{\mathsf{P}} = coNP$. The relations between the classes belonging to polynomial hierarchy are visible in Figure 5.

In addition to the aforementioned classes, we will also deal with D^{P} (difference polynomial time) problems [SW15]. We say that a problem L is in D^{P} iff it can be characterized as $L = L_1 \cap L_2$ (the intersection of "yes" instances) and $L_1 \in NP$ and $L_2 \in coNP$. For example, SAT–UNSAT is a known D^{P}–complete problem. The polynomial hierarchy can be naturally extended to account for higher levels of the D^{P} class, i.e. for $i \geq 1$, a problem L is in $\mathrm{D}_i^{\mathsf{P}}$ iff it can be characterized as $L = L_1 \cap L_2$ and $L_1 \in \Sigma_i^{\mathsf{P}}$ and $L_2 \in \Pi_i^{\mathsf{P}}$ [5].

$$
\begin{tikzcd}
 & NP \arrow[dr, phantom, "\subsetneq"] & & & \Sigma_2^{\mathsf{P}} \arrow[dr, phantom, "\subsetneq"] & & & \Sigma_3^{\mathsf{P}} & \\
P \arrow[ur, phantom, "\subsetneq"] \arrow[dr, phantom, "\subsetneq"] & & \mathrm{D}^{\mathsf{P}} \arrow[r, phantom, "\subseteq"] & \Delta_2^{\mathsf{P}} \arrow[ur, phantom, "\subsetneq"] \arrow[dr, phantom, "\subsetneq"] & & \mathrm{D}_2^{\mathsf{P}} \arrow[r, phantom, "\subseteq"] & \Delta_3^{\mathsf{P}} \arrow[ur, phantom, "\subsetneq"] \arrow[dr, phantom, "\subsetneq"] & & \cdots \\
 & coNP \arrow[ur, phantom, "\subsetneq"] & & & \Pi_2^{\mathsf{P}} \arrow[ur, phantom, "\subsetneq"] & & & \Pi_3^{\mathsf{P}} &
\end{tikzcd}
$$

Figure 5: Relation between complexity classes

4 Complexity Analysis of ADF Semantics

In this section we would like to focus on the computational complexity of the presented extension–based semantics for the abstract dialectical frameworks. We will assume that the frameworks we are working with are finite, i.e. the set of arguments is finite. We will focus on the four traditional problems – verification, existence, credulous and skeptical acceptability:

[5] In this case, $\mathrm{D}_1^{\mathsf{P}}$ is simply D^{P}

Verification of (Ver_σ)	
Instance:	An ADF $D = (A, C)$, a set of arguments $X \subseteq A$ or a labeling $Lab : A \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ and a semantics σ
Problem:	Is X (or Lab) a σ–extension (σ–labeling) of D?

Existence of a σ–extension or labeling ($Exists_\sigma$)	
Instance:	An ADF $D = (A, C)$ and a semantics σ
Problem:	Does there exist a σ–extension (σ–labeling) for D?

Credulous acceptance ($Cred_\sigma$)	
Instance:	An ADF $D = (A, C)$, an argument $a \in A$ and a semantics σ
Problem:	Is a contained in any σ–extension (σ–labeling) of D?

Skeptical acceptance ($Skept_\sigma$)	
Instance:	An ADF $D = (A, C)$, an argument $a \in A$ and a semantics σ
Problem:	Is a contained in every σ–extension (σ–labeling) of D?

The existence problem will receive the least attention, as for the majority of our semantics this problem is trivial. This is a result of Theorems 37 and 39 and the fact that $\emptyset$ is an admissible extension of any type. However, we will introduce a number of smaller problems focusing on the computational complexity of the building blocks of the extension–based semantics, that is: decisiveness, evaluations and ranges. Please note that in our analysis we assume that we are dealing with propositional ADFs, i.e. the acceptance conditions are represented as propositional formulas. Our study can be seen as an extension of the work in [Wal14] to other families of the ADF semantics. The existing results for the labeling–based and extension–based semantics [BW10, Wal14, BES$^+$13, SW15] can be seen in Table 1.

Table 1: Existing computational complexity results for ADF semantics.

σ		Ver_σ	$Exists_\sigma$	$Cred_\sigma$	$Skept_\sigma$
LAB	ADM	coNP–c	trivial	Σ_2^{P}–c	trivial
	COMP	D^{P}–c	trivial	Σ_2^{P}–c	coNP–c
	PREF	Π_2^{P}–c	trivial	Σ_2^{P}–c	Π_3^{P}–c
	GRD	D^{P}–c	trivial	coNP–c	coNP–c
EXT	MOD	in P	NP–c	NP–c	NP–c
	STB	coNP–c	Σ_2^{P}–c	Σ_2^{P}–c	Π_2^{P}–c
	GRD	coNP–h	trivial		

4.1 Basic Components

Before we analyze our semantics, we would like to focus on the complexity analysis of simpler notions. In particular, we will address the issues of the verification and existence of particular types of interpretations, evaluations and ranges. In order to simplify our task, we would like to introduce somewhat more relaxed versions of our evaluations. In Section 2.1.2, we have observed that the use of arbitrary decisively in interpretations, not just the minimal ones, would not affect our semantics. Minimality plays a role in deciding whether a given ADF is an AADF$^+$, however, we will not be dealing with this problem here. Consequently, we propose the following new notions:

Definition 51. Let $D = (A, C)$ be an ADF and $E \subseteq A$ a set of arguments. A **weak positive dependency function** on E is a function wpd assigning every argument $a \in E$ an interpretation $v \in dec(in, a)$ or $\mathcal{N}$ for null iff no such interpretation can be found. A **weak** X **pd–evaluation** for $X \in \{$standard, partially acyclic, acyclic$\}$ is an X pd–evaluation constructed using wpd.

Given these notions, we can now define the problem we would like to tackle in this section:

Verification of argument's decisiveness (Ver^{x}_{dec})	
Instance:	An ADF $D = (A, C)$, an argument $a \in A$, a partial interpretation v on A and $x \in \{in, out\}$.
Problem:	Is a decisively x for v in D?
Verification of a weak standard pd–evaluation ($Ver^{w}_{\textit{st-pd}}$)	
Instance:	An ADF $D = (A, C)$, $a_n \in A$ and a pair (F, B) with $F, B \subseteq A$ and $a \in F$.
Problem:	Is (F, B) a weak standard pd–evaluation on A for a?
Verification of a weak acyclic pd–evaluation ($Ver^{w}_{\textit{acy-pd}}$)	
Instance:	An ADF $D = (A, C)$, $a_n \in A$ and a pair (F, B) with $F = (a_0, \dots, a_n)$ being a sequence of elements on $E \subseteq A$ and $B \subseteq A$.
Problem:	Is (F, B) a weak acyclic pd–evaluation on A for a_n?
Verification of a weak partially acyclic pd–evaluation ($Ver^{w}_{\textit{par-pd}}$)	
Instance:	An ADF $D = (A, C)$, $a_n \in A$ and a pair (F, G, B) with $F, B \subseteq A$ and $G = (a_0, \dots, a_n)$ being a sequence of elements on $E \subseteq A$.
Problem:	Is (F, G, B) a weak partially acyclic pd–evaluation on A for a_n?
Existence of a weak X pd–evaluation ($Exists^{w}_{\textit{X-pd}}$)	
Instance:	An ADF $D = (A, C)$ and an argument $a \in A$.
Problem:	Is there an X pd–evaluation on A for a?
Existence of a weak non self–blocking X pd–evaluation ($nsb\text{-}Exists^{w}_{\textit{X-pd}}$)	
Instance:	An ADF $D = (A, C)$ and an argument $a \in A$.
Problem:	Is there an X pd–evaluation on A for a s.t. its blocking set is disjoint from its pd–set and/or pd–sequence?

Verification of standard discarded set (Ver_{st}^{dis})	
Instance:	An ADF $D = (A, C)$, a conflict–free set of arguments $E \subseteq A$ and a set $X \subseteq A \setminus E$.
Problem:	Is X the standard discarded set of E in D?

Verification of a subset of standard discarded set (sub-Ver_{st}^{dis})	
Instance:	An ADF $D = (A, C)$, a conflict–free set of arguments $E \subseteq A$ and a set $X \subseteq A \setminus E$.
Problem:	Is X a subset of the standard discarded set of E in D?

Verification of partially acyclic discarded set (Ver_{par}^{dis})	
Instance:	An ADF $D = (A, C)$, a conflict–free set of arguments $E \subseteq A$ and a set $X \subseteq A \setminus E$.
Problem:	Is X the partially acyclic discarded set of E in D?

Verification of a subset of partially acyclic discarded set (sub-Ver_{par}^{dis})	
Instance:	An ADF $D = (A, C)$, a conflict–free set of arguments $E \subseteq A$ and a set $X \subseteq A \setminus E$.
Problem:	Is X a subset of the partially acyclic discarded set of E in D?

Verification of acyclic discarded set (Ver_{acy}^{dis})	
Instance:	An ADF $D = (A, C)$, a conflict–free set of arguments $E \subseteq A$ and a set $X \subseteq A \setminus E$.
Problem:	Is X the acyclic discarded set of E in D?

Verification of a subset of acyclic discarded set (sub-Ver_{acy}^{dis})	
Instance:	An ADF $D = (A, C)$, a conflict–free set of arguments $E \subseteq A$ and a set $X \subseteq A \setminus E$.
Problem:	Is X a subset of the acyclic discarded set of E in D?

Verification of acyclic discarded set (acy-Ver_{acy}^{dis})	
Instance:	An ADF $D = (A, C)$, a pd–acyclic conflict–free set of arguments $E \subseteq A$ and a set $X \subseteq A \setminus E$.
Problem:	Is X the acyclic discarded set of E in D?

4.1.1 Decisiveness and Evaluations

First we deal with the problem of verifying the decisiveness of an argument w.r.t. a given interpretation v. In reality, it is little more than deciding whether the acceptance condition turns into a tautology or becomes unsatisfiable if the occurrences of arguments in the condition are replaced by the values assigned to them by v (if applicable).

Proposition 52. Ver_{dec}^{in} *is* coNP*-complete.*

Proposition 53. Ver^{out}_{dec} *is* coNP*-complete.*

With this at hand, we can focus on the verification problems for our evaluations; we focus only on the weak approach. The first step is checking whether a given interpretation assignment can serve as a basis for a pd–function. Consequently, we make sure that all of the interpretations are decisively in for their arguments at the same time:

Proposition 54. *Let $D = (A, C)$ be an ADF, $X \subseteq A$ a set of arguments and $V_X : X \to v_x$ a function assigning partial two–valued interpretations on subsets of A to arguments in X. Verifying every interpretation is decisively in for its argument is coNP–complete.*

The same results hold for checking whether all interpretations are decisively out, which will be more useful in the analysis of admissibility.

Proposition 55. *Let $D = (A, C)$ be an ADF, $X \subseteq A$ a set of arguments and $V_X : X \to v_x$ a function assigning partial two–valued interpretations on subsets of A to arguments in X. Verifying that every interpretation is decisively out for its argument is coNP–complete.*

We will also consider the problem of verifying that there is no argument outside a given interpretation that is decisively in. This will prove useful in the analysis of the complete semantics as it will allow us to check that there is no argument outside the accepted set of arguments that should be included in it w.r.t. the desired range:

Proposition 56. *Let $D = (A, C)$ be an ADF and v an interpretation defined on $E \subseteq A$. Verifying that no argument in $A \setminus E$ is decisively in w.r.t. v is NP–complete.*

From Proposition 54 it also follows that verifying if an assignment is a pd–function is also in coNP. We can now move on to the problem of verifying whether a given pair (triple) is an evaluation of the desired type. It practically boils down to guessing an interpretation assignment, verifying its decisiveness and checking if the conditions of an evaluation are satisfied, which can be done in Σ^{P}_2:

Proposition 57. $Ver^{w}_{st\text{-}pd}$ *is* Σ^{P}_2*-complete.*

Proposition 58. $Ver^{w}_{acy\text{-}pd}$ *is in* Σ^{P}_2*.*

Proposition 59. $Ver^{w}_{par\text{-}pd}$ *is in* Σ^{P}_2*.*

Next comes the problem concerning the existence of evaluations of a given type. In practice, it basically boils down to guessing a weak pd–function and verifying whether we can construct a suitable evaluation. Since we are not restricted to non self–blocking evaluations, we can use full two–valued interpretations in our analysis. This is not unlike considering $\mathbf{f}$–completions of the minimal decisively in interpretations we would use in the construction of the original pd–functions and evaluations – in practice, only the blocking set is affected, which is not relevant for the existence of an evaluation that does not have to meet any special constraints.

Proposition 60. *$Exists^{w}_{st\text{-}pd}$ is in NP.*

Similar analysis follows for the acyclic pd–evaluations:

Proposition 61. *$Exists^{w}_{acy\text{-}pd}$ is in NP.*

These results also imply that the problem of existence of normal standard (acyclic) evaluations is also in NP – the blocking set can be trimmed down to fit the original approach. Since partially acyclic evaluations are just a refinement of the standard ones, again the NP membership for existence follows. We can thus sum it up in the following way:

Proposition 62. *$Exists^{y}_{x\text{-}pd}$ for $x \in \{st, par, acy\}$ and $y \in \{w, n\}$, where w stands for weak and n for normal, is in NP.*

Apart from the simple existence of an evaluation of any kind, we are also interested in the ones that are not self–blocking, since they are the ones that arguments in an extension actually form. For the standard evaluations, we can still create an approach that uses full interpretations and therefore the problem remains in NP.

Proposition 63. *$nsb\text{-}Exists^{w}_{st\text{-}pd}$ is in NP.*

Just like in the case of unconstrained existence, the NP result carries over to normal standard evaluations and to weak (normal) partially acyclic ones. However, the problem becomes slightly more difficult in the acyclic case. This is due to the fact that we cannot exploit the full interpretation assignment anymore: given an assignment of partial interpretations with which we can form an acyclic pd–evaluation, if we "fill the missing mappings" with $\mathbf{f}$, the evaluation can become self–blocking. If we fill it with $\mathbf{t}$, we can lose acyclicity.

Proposition 64. *$nsb\text{-}Exists^{w}_{acy\text{-}pd}$ is in Σ^{P}_{2}.*

We can thus sum it up in the following way and continue with the analysis of the ranges and discarded sets.

Proposition 65. *$nsb\text{-}Exists^{y}_{x\text{-}pd}$ for $x \in \{st, par\}$ and $y \in \{w, n\}$, where w stands for weak and n for normal, is in NP. $nsb\text{-}Exists^{y}_{acy\text{-}pd}$ is in Σ^{P}_{2}.*

4.1.2 Discarded Sets

In this section we will focus on the verification problems concerning the various types of discarded sets and ranges. We will consider both exact discarded sets and just their subsets, which can be useful in the case of e.g. admissible semantics.

We start with the standard discarded set; in this case, the membership results we have managed to obtain are the same both for computing the exact set and only a part of it:

Proposition 66. *Ver^{dis}_{st} is in Δ^{P}_{2}.*

Proposition 67. *sub-Ver^{dis}_{st} is in Δ^{P}_2.*

There appears to be little gain in considering only subsets of the partially acyclic range as well:

Proposition 68. *Ver^{dis}_{par} is in Σ^{P}_2.*

Proposition 69. *sub-Ver^{dis}_{par} is in Π^{P}_2.*

Fortunately, the results for the acyclic range are somewhat more encouraging. The general $\mathrm{D}^{\mathsf{P}}_2$ result for the discarded sets paired with conflict–free extensions can be improved either by searching for a subset of the discarded set (we go down to Π^{P}_2) or by taking the pd–acyclic conflict–free extensions as our basis (we obtain the Σ^{P}_2 class).

Proposition 70. *Ver^{dis}_{acy} is in $\mathrm{D}^{\mathsf{P}}_2$.*

Proposition 71. *sub-Ver^{dis}_{acy} is in Π^{P}_2.*

Proposition 72. *acy-Ver^{dis}_{acy} is in Σ^{P}_2.*

These results conclude our section on the "building blocks" of the extension–based ADF semantics. We can now move on to analyzing the actual extensions.

4.2 Conflict–free Semantics

The verification problem for the conflict–free semantics is quite straightforward; checking that the conditions evaluate to in under a given set of arguments can be done in polynomial time:

Proposition 73. *Ver_{cf} is in P.*

In order to verify that a given set is pd–acyclic conflict–free, we will use the following theorem that relates these sets to grounded extensions of ADF reducts [Pol15] and the complexity results for the verification problem of the grounded semantics from [Wal14].

Theorem 74. *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.*

Proposition 75. *$Ver_{acy\text{-}cf}$ is in D^{P}.*

We close this section with the analysis of the complexity of credulous reasoning for our semantics. A (pd–acyclic) conflict–free extension is in fact nothing more than a non self–blocking (acyclic) pd–evaluation. Therefore, using Propositions 63 and 64, we can show that our problems fall into the NP and Σ^{P}_2 classes for standard conflict–free and pd–acyclic conflict–free semantics respectively. The answer to the skeptical problem is always no, since $\emptyset$ is trivially a (pd–acyclic) conflict–free extension.

Proposition 76. *$Cred_{cf}$ is in NP.*

Proposition 77. *$Cred_{acy\text{-}cf}$ is in Σ^{P}_2.*

4.3 Grounded Semantics

In order to verify whether a given set is the grounded or acyclic grounded extension of a given framework, we can reuse the iterative approaches from Proposition 28 and Definition 29. This puts our problems in D^{P} and Δ_3^{P} respectively, though please note that these results are likely to be improved in the future.

Proposition 78. *Ver_{grd} is in D^{P}.*

Proposition 79. *$Ver_{acy\text{-}grd}$ is in Δ_3^{P}.*

Due to the fact that every framework produces a unique (acyclic) grounded extension, the credulous and skeptical problems are one and the same. In the case of the standard grounded semantics, we can exploit its correspondence to the labeling–based semantics (see Theorem 45). In other words, an argument is credulously accepted in a grounded extension iff it is credulously accepted in a grounded labeling. Therefore, the complexity of this particular problem is the same as in the labeling–based case [Wal14] and is thus coNP–complete. Unfortunately, we cannot proceed in the same manner in the acyclic grounded case. In order to establish the upper bound for the difficulty of the credulous (skeptical) reasoning for these semantics, we exploit the fact that it is the least aa–complete extension. Thus, it suffices to check whether there exists such an complete extension not containing the desired argument (we will analyze this further in Section 4.5).

Proposition 80. *$Cred_{grd}$ is coNP–complete.*

Proposition 81. *$Cred_{acy\text{-}grd}$ is in Π_2^{P}.*

4.4 Admissible Semantics

We can now start with the verification problem for our five types of admissible semantics. We will heavily depend on the results we have presented concerning the evaluations and ranges in Section 4.1. The proofs for the cc and ca_1–semantics follow the same pattern; we start with verifying whether the desired set is conflict–free to start with, compute the required discarded set and verify the decisiveness of the accepted arguments. When it comes to the semantics acyclic on the inside, we guess a pd–function and make sure we are able to create an acyclic pd–evaluation s.t. its blocking set is contained in the desired discarded set. It is worth noting that based on Theorem 42, a given set is a ca_2–admissible extension iff there exists a corresponding admissible labeling. We can therefore exploit the results from [Wal14] in order to show that the verification problem for ca_2–admissible semantics is in Σ_2^{P}. By using Lemma 21, Propositions 52, 53, 54, 66, 71 and 73 we obtain the following results:

Proposition 82. *$Ver_{cc\text{-}adm}$ is in Δ_2^{P}.*

Proposition 83. *$Ver_{ac\text{-}adm}$ is in Σ_2^{P}.*

Proposition 84. *$Ver_{aa\text{-}adm}$ is in Σ_2^{P}.*

Proposition 85. *$Ver_{ca_1\text{-}adm}$ is in Δ_3^{P}.*

Proposition 86. *$Ver_{ca_2\text{-}adm}$ is in Σ_2^{P}.*

We can observe that the ca_1–admissibility membership result is in a higher class than in the case of other semantics. This is a result of the fact that for this semantics, the arguments in the discarded set are not necessarily decisively in w.r.t. the acyclic range. Hence, we cannot use Lemma 21 to verify that the blocking set of the evaluation representing our extension is in the discarded set and need to resort to Proposition 71.

Let us now move on to the credulous reasoning problem. In all of the cases, with the exception of the ca_1–admissibility, our problem is in Σ_2^{P}. The fact that we jump to Σ_3^{P} for the ca_1–approach is caused by similar reasons as in Proposition 85.

Proposition 87. *$Cred_{cc\text{-}adm}$, $Cred_{ac\text{-}adm}$, $Cred_{aa\text{-}adm}$ and $Cred_{ca_2\text{-}adm}$ are in Σ_2^{P}.*

Proposition 88. *$Cred_{ca_1\text{-}adm}$ is in Σ_3^{P}.*

We can now analyze the hardness of the credulous reasoning. In order to do so, we will modify the method presented in [Wal14]. The proof will depend on reduction from $QBF_{\exists,w} - VALIDITY$ and will use the following ADF, depicted in Figure 6. Since it is an AADF$^+$, our results will hold for any semantics.

Lemma 89. *Let X, Y be two disjoint sets of propositional variables, and ψ a Boolean formula over $X \cup Y$. Define the ADF $D_\psi = (S, C)$ as follows.*

- *$S = X \cup \overline{X} \cup Y \cup \{f\}$;*
- *$\varphi_x = \neg\overline{x}$ for $x \in X$;*
- *$\varphi_{\overline{x}} = \neg x$ for $\overline{x} \in \overline{X}$; and*
- *$\varphi_y = \neg y$ for $y \in Y$; and*
- *$\varphi_f = \psi$.*

The ADF D_ψ is an AADF$^+$.

Proposition 90. *Let $x, y \in \{a, c\}$. $Cred_{xy\text{-}adm}$ is Σ_2^{P}–hard.*

This gives us the completeness result for the credulous reasoning under the cc, ac, aa and ca_2–admissible semantics. Just like in the case of conflict–free semantics, $\emptyset$ is an admissible extension of any type. Consequently, no argument can ever be skeptically accepted in these semantics and no further analysis is required.

Proposition 91. *$Cred_{cc\text{-}adm}$, $Cred_{ac\text{-}adm}$, $Cred_{aa\text{-}adm}$ and $Cred_{ca_2\text{-}adm}$ are Σ_2^{P}–complete.*

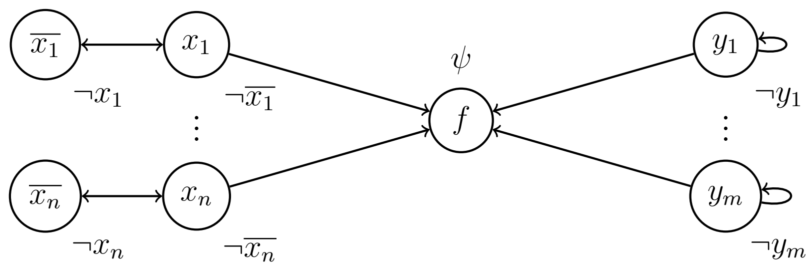


Figure 6: Constructed AADF+ for Lemma 89

4.5 Complete Semantics

Let us now move on to the complete semantics. The proofs for ac and cc–admissible semantics depended on the computation of the standard discarded set; consequently, adding one more NP call (see Proposition 56) in order to verify that no further arguments can be accepted, does not affect our results. Therefore, the membership results for the ac and cc–complete semantics remain the same as in the case of admissibility. The same holds for the ca_1–semantics; the lack of decisiveness in the acyclic range of conflict–free sets made it impossible to exploit Lemmas 19 and 21. The proofs for aa and ca_2–complete semantics are similar to the admissible ones, however, we need a more precise range analysis. Hence, rather than guessing a pd–function for the set we want to verify, we guess one covering all arguments. We can then construct evaluations for arguments outside the range to verify that no arguments can be rejected by the range (see Lemmas 22 and 20). Checking that no elements can be accepted simply adds an NP call as in the ac and cc–complete semantics.

Proposition 92. *$Ver_{\textit{cc-cmp}}$ is in Δ_2^{P}.*

Proposition 93. *$Ver_{\textit{ac-cmp}}$ is in Σ_2^{P}.*

Proposition 94. *$Ver_{\textit{aa-cmp}}$ is in Σ_2^{P}.*

Proposition 95. *$Ver_{\textit{ca}_1\textit{-cmp}}$ is in Δ_3^{P}.*

Proposition 96. *$Ver_{\textit{ca}_2\textit{-cmp}}$ is in Σ_2^{P}.*

Based on Theorems 37 and 38, we can observe that every cc, ac and aa–admissible extension will be contained in some cc, ac and aa–complete extension. Moreover, any complete set is also admissible. Consequently, the credulous reasoning problem for completeness is the same as for admissibility in the cc, ac and aa–approach. Since we are working with finite frameworks only, the same can be inferred for the ca_1 and ca_2–semantics due to Theorems 37 and 39:

Proposition 97. *$Cred_{\textit{cc-cmp}}$, $Cred_{\textit{ac-cmp}}$ are $Cred_{\textit{aa-cmp}}$ are Σ_2^{P}–complete.*

Proposition 98. *$Cred_{ca_2\text{-}cmp}$ is Σ_2^{P}–complete. $Cred_{ca_1\text{-}cmp}$ is in Σ_3^{P} and is Σ_2^{P}–hard.*

Finally, we can observe that due to the structure of the ac, cc and aa–complete semantics (see Theorem 38), the problem of skeptical reasoning boils down to checking whether a given argument is contained in the least complete extension, which depending on the approach happens to be the grounded or acyclic grounded. Therefore, the complexity of this problem is the same as in Propositions 80 and 81. Unfortunately, for the remaining ca_1 and ca_2–approaches, we need to verify whether there exists a complete extension not containing the argument in question.

Proposition 99. *$Skept_{cc\text{-}cmp}$ and $Skept_{ac\text{-}cmp}$ are coNP–complete.*

Proposition 100. *$Skept_{aa\text{-}cmp}$ is in Π_2^{P}.*

Proposition 101. *$Skept_{ca_1\text{-}cmp}$ is in Π_3^{P}.*

Proposition 102. *$Skept_{ca_2\text{-}cmp}$ is in Π_2^{P}.*

4.6 Preferred Semantics

Let us move on to the preferred semantics. In order to see whether a given set is preferred, we first verify that it is admissible and then that there is no "bigger" admissible set in the framework. This brings us to the following results:

Proposition 103. *$Ver_{cc\text{-}prf}$ is in Π_2^{P}.*

Proposition 104. *$Ver_{ac\text{-}prf}$, $Ver_{aa\text{-}prf}$ and $Ver_{ca_2\text{-}prf}$ are in $\mathrm{D}_2^{\mathsf{P}}$.*

Proposition 105. *$Ver_{ca_1\text{-}prf}$ is in Π_3^{P}.*

The credulous reasoning problem for a given x–preferred semantics is the same as for the associated x–admissible semantics. This depends on the fact that for every admissible extension there exists a preferred one containing it, which is always true for finite frameworks, even in the case of ca_1 and ca_2–approaches (see Theorems 37 and 39). Hence, what is left to analyze is the complexity of skeptical reasoning. We focus on the co–problem, i.e. the existence of a preferred extension of a given type that does not contain the argument we want.

Proposition 106. *$Skept_{cc\text{-}prf}$ is in Π_3^{P}.*

Proposition 107. *$Skept_{ac\text{-}prf}$, $Skept_{aa\text{-}prf}$ and $Skept_{ca_2\text{-}prf}$ are in Π_3^{P}.*

Proposition 108. *$Skept_{ca_1\text{-}prf}$ is in Π_4^{P}.*

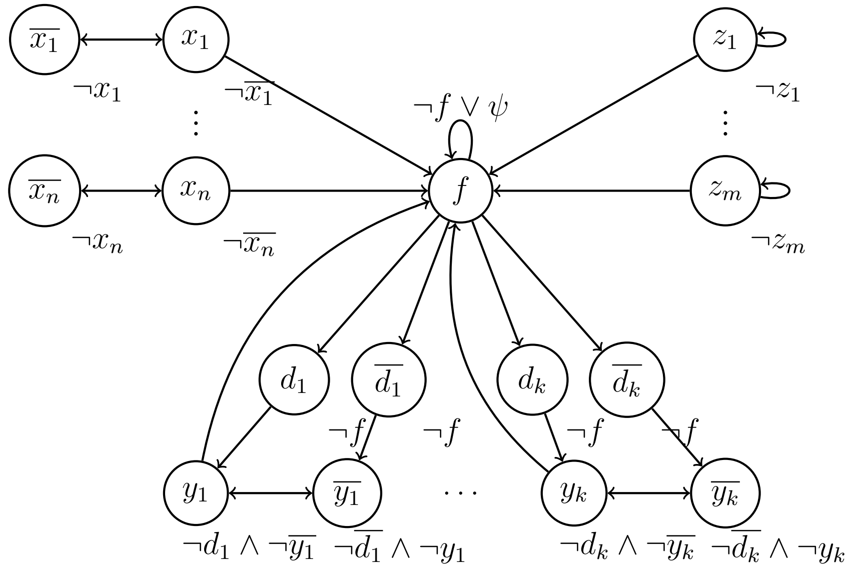


Figure 7: Constructed AADF+ for Lemma 110

We can now analyze the hardness of the skeptical reasoning for our preferred semantics. We will modify the method presented in [Wal14], similarly as we did in the case of the admissible semantics, and use the following framework in our proof. Since it is an AADF$^+$, our results will hold for any semantics.

Lemma 109. *Let $\phi \in QBF_{\forall,3}$ be a closed QBF of the form $\phi = \forall X \exists Y \forall Z \psi$. We define the ADF $D_{pref(\psi)}$ with $S = X \cup \overline{X} \cup Y \cup \overline{Y} \cup D \cup \overline{D} \cup Z \cup \{f\}$ and the acceptance conditions as follows:*

- $\varphi_{x_i} = \neg \overline{x_i}$ *for* $x_i \in X$
- $\varphi_{\overline{x_i}} = \neg x_i$ *for* $\overline{x_i} \in \overline{X}$
- $\varphi_{d_i} = \neg f$ *for* $d_i \in D$
- $\varphi_{\overline{d_i}} = \neg f$ *for* $\overline{d_i} \in \overline{D}$
- $\varphi_{y_i} = \neg d_i \wedge \neg \overline{y_i}$ *for* $y_i \in Y$
- $\varphi_{\overline{y_i}} = \neg \overline{d_i} \wedge \neg y_i$ *for* $\overline{y_i} \in \overline{Y}$
- $\varphi_{z_i} = \neg z_i$ *for* $z_i \in Z$
- $\varphi_f = \neg f \vee \psi$

Then $D_{pref(\psi)}$ is an AADF$^+$.

Proposition 110. *$Skept_{xy-pref}$ is Π_3^{P}–hard for ADFs for $x, y \in \{a, c\}$.*

This brings us to our final results:

Proposition 111. *$Skept_{cc\text{-}prf}$, $Skept_{ac\text{-}prf}$, $Skept_{aa\text{-}prf}$ and $Skept_{ca_2\text{-}prf}$ are Π_3^{P}–complete. $Skept_{ca_1\text{-}prf}$ is Π_3^{P}–hard and in Π_4^{P}.*

Proposition 112. *$Cred_{cc\text{-}prf}$, $Cred_{ac\text{-}prf}$ and $Cred_{aa\text{-}prf}$ are Σ_2^{P}–complete.*

Proposition 113. *$Cred_{ca_2\text{-}prf}$ is Σ_2^{P}–complete. $Cred_{ca_1\text{-}prf}$ is in Σ_3^{P} and is Σ_2^{P}–hard.*

4.7 Summary

The new and existing results concerning the computational complexity of ADF semantics can be seen in Table 2. The blue and the green entries represent our findings, with the latter distinguishing the complete results. We can observe that the xy–semantics are in principle more complex than their labeling–based counterparts. There are two reasons for this; one is that labelings store more information (in particular, concerning the **f** arguments), which the extension–based semantics need to reconstruct. The other reason is that the xy–semantics are more specialized than the labeling based ones and require evaluation analysis, not just decisiveness as in the characteristic operator. Nevertheless, many of the skeptical and credulous reasoning results carry over to our setting as well.

	σ	Ver_σ	$Exists_\sigma$	$Cred_\sigma$	$Skept_\sigma$
ST	CF	**in P**	trivial	**in NP**	trivial
	MOD	in P	NP–c	NP–c	coNP–c
	GRD	coNP–h, **in** D^P	trivial	**coNP–c**	**coNP–c**
	GRD–LAB	D^P–c	trivial	coNP–c	coNP–c
ACY	CF	**in** D^P	trivial	**in** Σ_2^P	trivial
	STB	coNP–c	Σ_2^P–c	Σ_2^P–c	Π_2^P–c
	GRD	**in** Δ_3^P	trivial	**in** Π_2^P	**in** Π_2^P
ADM	AA	**in** Σ_2^P	trivial	Σ_2^P**–c**	trivial
	AC	**in** Σ_2^P	trivial	Σ_2^P**–c**	trivial
	CC	**in** Δ_2^P	trivial	Σ_2^P**–c**	trivial
	CA_1	**in** Δ_3^P	trivial	Σ_2^P**–h, in** Σ_3^P	trivial
	CA_2	**in** Σ_2^P	trivial	Σ_2^P**–c**	trivial
	LAB	coNP–c	trivial	Σ_2^P–c	trivial
$COMP$	AA	**in** Σ_2^P	trivial	Σ_2^P**–c**	**in** Π_2^P
	AC	**in** Σ_2^P	trivial	Σ_2^P**–c**	**coNP–c**
	CC	**in** Δ_2^P	trivial	Σ_2^P**–c**	**coNP–c**
	CA_1	**in** Δ_3^P	trivial	Σ_2^P**–h, in** Σ_3^P	**in** Π_3^P
	CA_2	**in** Σ_2^P	trivial	Σ_2^P**–c**	**in** Π_2^P
	LAB	D^P–c	trivial	Σ_2^P–c	coNP–c
$PREF$	AA	**in** D_2^P	trivial	Σ_2^P**–c**	Π_3^P**–c**
	AC	**in** D_2^P	trivial	Σ_2^P**–c**	Π_3^P**–c**
	CC	**in** Π_2^P	trivial	Σ_2^P**–c**	Π_3^P**–c**
	CA_1	**in** Π_3^P	trivial	Σ_2^P**–h, in** Σ_3^P	Π_3^P**–h, in** Π_4^P
	CA_2	**in** D_2^P	trivial	Σ_2^P**–c**	Π_3^P**–c**
	LAB	Π_2^P–c	trivial	Σ_2^P–c	Π_3^P–c

Table 2: The currently known complexity classes concerning the verification, existence and skeptical (credulous) acceptance problems for extension and labeling–based semantics of ADFs. The blue and the green entries represent our findings, with the latter distinguishing the complete results.

References

[BES+13] 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 (IJ-*

CAI 2013), pages 803–809. AAAI Press, 2013.

[BG09] Pietro Baroni and Massimiliano Giacomin. Semantics of abstract argument systems. In Guillermo Simari and Iyad Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 25–44. Springer, 2009.

[BW10] 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.

[Ell12] Stefan Ellmauthaler. Abstract dialectical frameworks: properties, complexity, and implementation. Master's thesis, Faculty of Informatics, Institute of Information Systems, Vienna University of Technology, 2012.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Theoretical computer science. Addison-Wesley, 1994.

[Pol14] Sylwia Polberg. Extension–based semantics of abstract dialectical frameworks. Technical Report DBAI-TR-2014-85, Institute for Information Systems, Technical University of Vienna, 2014.

[Pol15] 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.

[PWW13] Sylwia Polberg, Johannes Peter Wallner, and Stefan Woltran. Admissibility in the abstract dialectical framework. In Paolo Torroni Leon van der Torre João 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 *LNCS*, pages 102–118. Springer, 2013.

[Str13a] Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence*, 205:39 – 70, 2013.

[Str13b] Hannes Strass. Instantiating knowledge bases in abstract dialectical frameworks. In Paolo Torroni Leon van der Torre João 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 *LNCS*, pages 86–101. Springer, 2013.

[SW15] Hannes Strass and Johannes Peter Wallner. Analyzing the Computational Complexity of Abstract Dialectical Frameworks via Approximation Fixpoint Theory. *Artificial Intelligence*, 226:34–74, 2015.

[Wal14] Johannes P. Wallner. *Complexity Results and Algorithms for Argumentation - Dung's Frameworks and Beyond*. Phd thesis, Vienna University of Technology, Institute of Information Systems, Vienna, Austria, 2014.

5 Proof Appendix

5.1 Abstract Dialectical Frameworks

Lemma 19. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a conflict–free set of arguments and v a two–valued interpretation s.t. $v^{\mathbf{t}} = E$. If every argument in $v^{\mathbf{f}}$ is decisively out w.r.t. v, then the partially acyclic range v_E^a of E is a completion of v. For any conflict–free set E there is an interpretation v with $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} \subseteq E^{p+}$ s.t. all arguments in $v^{\mathbf{f}}$ are decisively out w.r.t. v.*

Proof. Let v_E^a be the partially acyclic range of E. Assume it is not a completion of v. As the $\mathbf{t}$ mappings of both interpretations are the same, this means there is an argument $a \in A$ s.t. $v(a) = \mathbf{f}$ and v_E^p is not defined for a. Therefore, the argument a has a partially acyclic evaluation $(F, (a_0, ..., a_n), B)$ s.t. $B \cap E = \emptyset$ and $F \subseteq E$, and yet is decisively out w.r.t. v.

From decisiveness it follows that v is in "conflict" with any decisively in interpretation v' for a, i.e. for every such v' there is an argument $x \in A$ s.t. $v'(x) \neq v(x)$. If it were not the case and there was no argument mapped to opposite values by the interpretations, then the union of these interpretations would be a completion of both. It would evaluate the condition of x to in our out. Thus, it would be impossible either for v to be decisively out or for v' to be decisively in for x.

Since $F \subseteq E$, it cannot be the case that any argument in F is decisively out w.r.t. v. Let us now focus on a_0 and its decisively in interpretation v_{a_0} with which it entered the partially acyclic evaluation. Since $v_{a_0}^{\mathbf{f}} \cap E = \emptyset$ and $v_{a_0}^{\mathbf{t}} \subseteq E$, it is not possible that a_0 is decisively out w.r.t. v. As $v_{a_1}^{\mathbf{f}} \cap E = \emptyset$ and $v_{a_1}^{\mathbf{t}} \subseteq E \cup \{a_0\}$ and $v(a_0) \neq \mathbf{f}$, it is not possible that a_1 is decisively out w.r.t. v. We can continue in the same manner until we reach a_n and conclude that if a_n could not have been decisively out w.r.t. v. Thus, if v maps to $\mathbf{f}$ a given argument, then so does v_E^p and the partially acyclic range is a completion of v.

The fact that there exists an interpretation v with $v^{\mathbf{f}} \subseteq E^{p+}$ arguments that are decisively out follows from Proposition 33. $\square$

Lemma 20. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a conflict–free set of arguments and $X \subseteq A \setminus E$ a set of arguments. Let v be a two–valued interpretation s.t. $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} = X$. Then v is the partially acyclic range interpretation of E iff all arguments in X are decisively out w.r.t. v and every $x \in A \setminus (E \cup X)$ has a partially acyclic evaluation (F^x, G^x, B^x) s.t. $B^x \cap E = \emptyset$ and $F^x \subseteq E$.*

Proof. Let us focus on the if–then direction. If v is the partially acyclic range interpretation, then the decisiveness of the arguments in X holds by Proposition 33. If an argument x is not in the range, then by Definition 17 it means it has an unblocked partially acyclic evaluation with the pd–set contained in E. Therefore, this direction holds.

Let us focus on the other way. If all arguments in X are decisively out w.r.t. v, then the partially acyclic range interpretation of E is a completion of v and v has a chance of being the actual range by Lemma 19. Since every other argument has an unblocked partially acyclic evaluation with a pd–set in E, then by the Definition 17 it cannot be the case that there is an argument mapped to $\mathbf{f}$ by the range but not by v. Thus, v is the acyclic range interpretation of E. $\square$

Lemma 21. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a conflict–free set of arguments and v a two–valued interpretation s.t. $v^{\mathbf{t}} = E$. If every argument in $v^{\mathbf{f}}$ is decisively out w.r.t. v, then the acyclic range v_E^a of E is a completion of v. If E is pd–acyclic conflict–free, then there is an interpretation v with $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} \subseteq E^{a+}$ s.t. all arguments in $v^{\mathbf{f}}$ are decisively out w.r.t. v.*

Proof. The fact that v_E^a is a completion of v follows from Lemmas 18 and 19 The fact that there exists an interpretation v with $v^{\mathbf{f}} \subseteq E^{a+}$ arguments that are decisively out follows from Proposition 33. □

Lemma 22. *Let $D = (A, C)$ be an ADF, $E \subseteq A$ a pd–acyclic conflict–free set of arguments and $X \subseteq A \setminus E$ a set of arguments. Let v be a two–valued interpretation s.t. $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} = X$. Then v is the acyclic range interpretation of E iff all arguments in X are decisively out w.r.t. v and every $x \in A \setminus (E \cup X)$ has an acyclic evaluation (F^x, B^x) s.t. $B^x \cap E = \emptyset$.*

Proof. Let v be an interpretation s.t. $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} = X$. Let us show that v is the acyclic range interpretation of E iff all arguments in X are decisively out w.r.t. v and every $x \in A \setminus (E \cup X)$ has an acyclic pd–evaluation (F^x, B^x) s.t. $B^x \cap E = \emptyset$. If v is the acyclic range interpretation, then decisiveness of the arguments holds by Proposition 33 and the fact that every other argument has an unblocked acyclic pd–evaluation comes from the Lemma 16. If the arguments are decisively out, then the acyclic range interpretation of E is a completion of v and v has a chance of being the actual range by Lemma 21. Since every other argument has an unblocked acyclic pd–evaluation, then by Lemma 16 it cannot be the case that there is an argument mapped to $\mathbf{f}$ by the range but not by v. Thus, v is the acyclic range interpretation of E. □

Theorem 39. *Let $D = (A, C)$ be an ADF s.t. A is finite. Given that $x \in \{ca_1, ca_2\}$, D possesses at least one x–preferred extension and for every x–admissible set E of D, there exists an x–preferred extension E' of D s.t. $E \subseteq E'$.*

Proof. Since the set of arguments is finite, the collection of x–admissible extensions is finite as well. Therefore, it will always possess maximal elements, which is precisely what x–preferred extensions are. It is now easy to verify that every x–admissible set will be contained in some x–preferred one. □

5.2 Complexity Analysis of Basic Components

Proposition 52. *Ver^{in}_{dec} is* coNP*-complete.*

Proof. Let us consider an argument a and its condition $C_a = \varphi_a$, where φ_a is a propositional function. Let v be an interpretation defined on a set E w.r.t. which we want to check for decisiveness. An interpretation is decisively in for an argument a iff there is no set of arguments $S \subseteq A$ s.t. $v^{\mathbf{t}} \subseteq S$, $S \cap v^{\mathbf{f}} = \emptyset$ and $C_a(S \cap par(a)) = out$. We can therefore consider the co–problem. We guess a set $S \subseteq A$ s.t. $v^{\mathbf{t}} \subseteq S$ and $S \cap v^{\mathbf{f}} = \emptyset$. Verifying that $C_a(S \cap par(a)) = out$ is achieved in polynomial time for propositional acceptance conditions. This problem is in NP, and therefore our main question is in coNP.

In order to show hardness, we consider the problem of checking whether a Boolean formula is a tautology, which is coNP–complete. Let φ be a Boolean formula and X the set of propositional variables that φ is based on. Without the loss of generality, let us assume that $b \notin X$. We construct an ADF (A, C) the following way: $A = X \cup \{b\}$, for every $x_i \in X$, $C_{x_i} = x$, and $C_b = \varphi$. Let v be an empty interpretation. We will show that φ is a tautology iff b is decisively in w.r.t. v in (A, C).

Assume that b is decisively in w.r.t. v, but φ is not a tautology. This means that there exists a set $S \subseteq X$ s.t. $S \not\models \varphi$. Therefore, the t–completion of v to S evaluates $C_b = \varphi$ to out. Hence, v could not have been decisively in for a and we reach a contradiction with our assumptions.

Let us now assume that φ is a tautology, but v is not decisively in for a. Due to the fact that v is empty, this means that there exists a set of arguments $S \subseteq par(a)$ s.t. $C_a(S) = out$. Since $par(a) = X$ by the construction of our ADF, then it has to be the case that $S \not\models \varphi$. Therefore, φ could not have been a tautology and we reach a contradiction.

Thus, we can conclude that our problem is both in coNP and coNP–hard. Hence, it is coNP–complete. ☐

Proposition 53. *Ver^{out}_{dec} is* coNP*-complete.*

Proof. This proposition can be proved in a similar manner as Proposition 52. Instead of checking if the formula with substituted values is a tautology, we verify that it is unsatisfiable. ☐

Proposition 54. *Let $D = (A, C)$ be an ADF, $X \subseteq A$ a set of arguments and $V_X : X \to v_x$ a function assigning partial two–valued interpretations on subsets of A to arguments in X. Verifying every interpretation is decisively in for its argument is coNP–complete.*

Proof. We need to check that $\bigwedge_{x \in X} Ver^{in}_{dec}(D, x, V_X(x)) = yes$. The problem $Ver^{in}_{dec}(D, x, V_X(x))$ was shown to be coNP-complete in Proposition 52. Hence, by the fact that the class coNP is closed under conjunction, membership in coNP follows [6]

In order to show hardness, we consider the problem of checking whether a Boolean formula is a tautology, which is coNP–complete. Let $\varphi = \psi_1 \wedge \psi_2 ... \wedge \psi_n$ be a Boolean formula. Let X_i be the set of propositional variables that ψ_i is based on and $X = \bigcup_{i=1}^{n} X_i$ the set of variables for φ. The sets do not need to be disjoint. Please note that every ψ_i is a Boolean formula of its own; neither ψ_i nor φ have to be in CNF. Without the loss of generality, let us assume that $b_1, ..., b_n \notin X$. We construct an ADF (A, C) the following way: $A = X \cup \{b_1, ..., b_n\}$, for every $x_j \in X$, $C_{x_j} = x_j$, and for every b_i, $C_{b_i} = \psi_i$. Let v be an empty interpretation. We will show that φ is a tautology iff every b_i is decisively in w.r.t. v in (A, C).

Assume that every b_i is decisively in w.r.t. v, but φ is not a tautology. This means that there exists a set $S \subseteq X$ s.t. $S \not\models \varphi$. Consequently, there must also be a formula ψ_i s.t. $S \cap X_i \not\models \psi_i$. However, this implies that $C_{b_i}(S \cap par(b_i)) = out$, which violates the decisiveness of b_i w.r.t. v. Thus, we reach a contradiction.

Let us now assume that φ is a tautology, but there is a b_i s.t. v is not decisively in for it. Due to the fact that v is empty, this means that there exists a set of arguments $S \subseteq par(b_i)$ s.t.

[6] A complexity class $\mathcal{C}$ is closed under conjunctions iff for any problem $\Gamma \in \mathcal{C}$, the problem of deciding whether for a finite set of instances of Γ each of these instances is a yes-instance is also in $\mathcal{C}$.

$C_{b_i}(S) = out$. Since $par(b_i) = X_i$ by the construction of our ADF, then it has to be the case that $S \not\models \psi_i$ and therefore $S \not\models \varphi$. Therefore, φ could not have been a tautology and we reach a contradiction.

We can conclude that our problem is both in coNP and coNP–hard. Hence, it is coNP–complete. □

Proposition 55. *Let $D = (A, C)$ be an ADF, $X \subseteq A$ a set of arguments and $V_X : X \to v_x$ a function assigning partial two–valued interpretations on subsets of A to arguments in X. Verifying that every interpretation is decisively out for its argument is coNP–complete.*

Proof. Checking that every interpretation in V_X is decisively out requires $\mid X \mid$ many independent coNP checks by Proposition 53. Therefore, our problem is in coNP. In order to show hardness, we can adapt the construction from the proof of Proposition 54 (instead of conjunction, we use a disjunction). Thus, our problem is coNP–complete. □

Proposition 56. *Let $D = (A, C)$ be an ADF and v an interpretation defined on $E \subseteq A$. Verifying that no argument in $A \setminus E$ is decisively in w.r.t. v is NP–complete.*

Proof. From Proposition 52 we know that checking whether an argument is decisively in is in coNP. Hence, checking whether an argument is not decisively in is in NP. Now, we need to consider a conjunction of these NP problems. Since NP is closed under conjunction, membership follows.

In order to show hardness, we consider a reduction from the co–problem of Ver^{in}_{dec}, which given an ADF $D' = (A', C')$, argument $a \in A'$ and an interpretation v' defined over a set $E' \subseteq A'$, verifies whether a is not decisively in w.r.t. v' in D'. This problem is NP–complete. We construct the ADF $D = (A', C)$, where the conditions are as follows: $C_a = C'_a$, for every $b \in E$, $C_b = b$, and for every $c \in A' \setminus (E' \cup \{a\})$, $C_c = \neg c$. We will show that a is not decisively in w.r.t. v' in D' iff no argument in $A' \setminus E'$ is decisively in w.r.t. v' in D.

Let a be not decisively in w.r.t. v' in D'. By the construction of D, it follows that it is not decisively in w.r.t. v' in D either. Moreover, for any argument $c \in A' \setminus (E' \cup \{a\})$, $C_c = \neg c$. Since v' is not defined for any such c, we can observe that the $\mathbf{t}$–completion of v' to $E' \cup \{c\}$ will evaluate C_c to out, while the $\mathbf{f}$–completion of v' to $E' \cup \{c\}$ will evaluate it to in. Therefore, no argument in $A' \setminus E'$ can be decisively in w.r.t. v' in D.

If no argument is decisively in w.r.t. v' in D, then by the construction of D, it cannot be the case that a is decisively in w.r.t. v' in D'. Hence, our problem is NP–hard, and based on the membership result, NP–complete as well.

□

Proposition 57. *$Ver^{w}_{st\text{-}pd}$ is Σ_2^{P}-complete.*

Proof. Let $D = (A, C)$ be an ADF and (F, B) a pair with $F, B \subseteq A$. We can guess an interpretation assignment $V = \{v_x \mid x \in F\}$ s.t. for every v_x, $v_x^{\mathbf{t}} \subseteq F$. Verifying that every v_x is decisively in for its x is in coNP by Proposition 54. Checking if the requirements of the standard evaluation hold is achievable in polynomial time. This puts our problem in Σ_2^{P}.

For hardness we reduce from satisfiability of a quantified Boolean formula (QBF) in prenex normal form with one quantifier alternation, starting with an existential quantifier. Let $\phi = \exists X \forall Y \psi$ be an arbitrary closed QBF of this form (w.l.o.g. we assume that $X \cap Y = \emptyset$).

Construct $D' = (A', C')$ in the following manner, as depicted in Figure 8:

- $A' = X \cup Y \cup \{y'\} \cup \{d\}$
- $C'_x = \varphi'_x = x$ for $x \in X$
- $C'_y = \varphi'_y = y$ for $y \in Y$
- $C'_{y'} = \varphi'_{y'} = \psi$
- $C'_d = \varphi'_d = \bigwedge_{x \in X} \neg x$

Further let $F = X \cup \{y'\} \cup \{d\}$ and $B = X$. We claim that (F, B) is a weak standard pd–evaluation for d in D iff ϕ is satisfiable.

Assume that ϕ is satisfiable. Then there is a set $E \subseteq X$ s.t. for any $S \subseteq Y$, we have $E \cup S \models \psi$. Let v be an interpretation s.t. $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} = (X \setminus E)$. Further, let u be an interpretation such that $u^{\mathbf{t}} = \emptyset$ and $u^{\mathbf{f}} = X$. Finally let w be an interpretation such that $w^{\mathbf{t}} = X$ and $w^{\mathbf{f}} = \emptyset$. We can observe that for any $x \in X$, w is a decisively in interpretation for x. The argument d is decisively in w.r.t. u. The argument y' is decisively in w.r.t. v; we can observe that $\varphi_{y'} = \psi$ and v assigns E to true and $X \setminus E$ to false and ψ is a tautology under this partial assignment by assumption. Furthermore, for each $q \in \{v, u, w\}$ it holds that $q^{\top} \subseteq F$. Moreover, $q^{\mathbf{f}} \subseteq B = X$. Since $u^{\mathbf{f}} = X$ it follows that (F, B) is a weak standard pd–evaluation for d in D.

Assume now that (F, B) is a standard pd–evaluation for d in D. Therefore, there is a weak pd–function $pd = \{(a, v_a) \mid a \in F\}$ w.r.t. which (F, B) is weak standard pd–evaluations. Thus, there is an interpretation v s.t. y' is decisively in w.r.t. v. Note that $(v^{\mathbf{t}} \cup v^{\mathbf{f}}) \cap Y = \emptyset$, since $(F \cup B) \cap Y = \emptyset$ by assumption. Let $v^{\mathbf{t}} \cap X = E$ and $v^{\mathbf{f}} \cap X = E'$. Since v is decisively in for y', then for each set $S \subseteq Y$ we have $E \cup S \models \varphi_{y'} = \psi$. This directly implies that ϕ is satisfiable.

The above shows that our problem is Σ_2^{P}–hard. As it is also in Σ_2^{P}, it holds that Ver^{w}_{st-pd} is Σ_2^{P}–complete. $\square$

Proposition 58. *$Ver^{w}_{acy\text{-}pd}$ is in Σ_2^{P}.*

Proof. Let $D = (A, C)$ be an ADF and (F, B) a pair with $F = (a_0, ..., a_n)$ a sequence of distinct elements of A and $B \subseteq A$. We guess an interpretation assignment $V = \{v_x \mid x \in F\}$ s.t. for every v_x, $v_x^{\mathbf{t}} \subseteq F$. Verifying that every v_x is decisively in for its x is in coNP by Proposition 54. Checking if requirements of the acyclic pd–evaluation hold is achievable in polynomial time. Therefore, the problem is in Σ_2^{P}. $\square$

Proposition 59. *$Ver^{w}_{par\text{-}pd}$ is in Σ_2^{P}.*

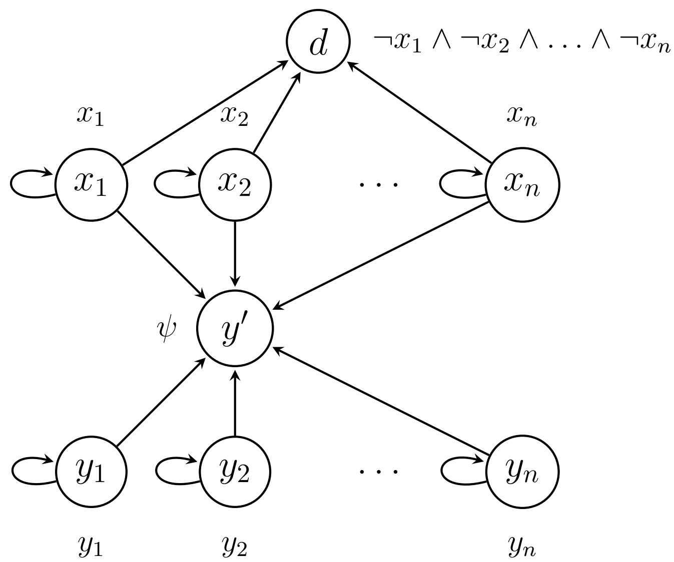


Figure 8: ADF for QBF formula $\exists X \forall Y \psi$

Proof. Let $D = (A, C)$ be an ADF and (F, G, B) a tuple with $G = (a_0, ..., a_n)$ a sequence of distinct elements of A and $F, B \subseteq A$. Guess an interpretation assignment $V = \{v_x \mid x \in F \cup G\}$ s.t. for every v_x, $v_x^{\mathbf{t}} \subseteq F \cup G$. . Verifying that every v_x is decisively in for its x is in coNP by Proposition 54. Checking if requirements of the partially acyclic evaluation hold is achievable in polynomial time. □

Proposition 60. *$Exists^{w}_{st\text{-}pd}$ is in NP.*

Proof. Let $D = (A, C)$. We can guess a set of arguments $F \subseteq A$ s.t. $a \in F$ and an interpretation assignment $V = \{v_x \mid x \in F\}$ s.t. every v_x is a full two–valued interpretation. Since the interpretations are full, verifying that they are decisively in for their arguments can be done in polynomial time – it simply boils down to verifying whether $v_x^{\mathbf{t}}$ is a model of the propositional acceptance condition of x. Checking if conditions of standard evaluations hold is also achievable in polynomial time. □

Proposition 61. *$Exists^{w}_{acy\text{-}pd}$ is in NP.*

Proof. Let $D = (A, C)$. Guess a set of arguments $F \subseteq A$ s.t. $a \in F$ and an interpretation assignment $V = \{v_x \mid x \in F\}$ s.t. every v_x is a full two–valued interpretation. Since the interpretations are full, verifying that they are decisively in for their arguments can be done in polynomial time. Checking if conditions of weak pd–functions and acyclic evaluations hold is also achievable in polynomial time. In particular, ordering the set into a sequence can be done by first identifying those decisively in interpretation that have empty $\mathbf{t}$ parts and putting their respective arguments into the sequence. We then find those interpretations s.t. their $\mathbf{t}$ parts are contained in the arguments we have already ordered, add them to the sequence and repeat the procedure until we are done. $\square$

Proposition 63. *nsb-$Exists^{w}_{st\text{-}pd}$ is in NP.*

Proof. We can guess a set of arguments $F \subseteq A$ s.t. $a \in F$ and an interpretation assignment s.t. every $x \in F$ has an interpretation v_x s.t. $v_x^{\mathbf{t}} = F$ and $v_x^{\mathbf{f}} = A \setminus F$. We receive full interpretations, and therefore verifying that every interpretation is decisively in for its associated argument can be done in polynomial time. Checking that we can form a weak standard evaluation with this weak pd–function is also achievable in polynomial time. $\square$

Proposition 64. *nsb-$Exists^{w}_{acy\text{-}pd}$ is in Σ_2^{P}.*

Proof. We can guess a set of arguments $F \subseteq A$ s.t. $a \in F$ and an interpretation assignment s.t. every $x \in F$ has an interpretation v_x s.t. $v_x^{\mathbf{t}} \subseteq F$ and $v_x^{\mathbf{f}} \cap F = \emptyset$. Verifying that every interpretation is decisively in for its associated argument is in coNP by Proposition 54. Checking that we can form non self–blocking acyclic evaluation from the assignment can be done in polynomial time. $\square$

Proposition 66. *Ver^{dis}_{st} is in Δ_2^{P}.*

Proof. We verify whether X is the standard discarded set by computing the actual set E^+ and comparing it with X. Based on Definition 13 and Proposition 53, we need a polynomially many calls to a coNP oracle in order to compute E^+. Since the range computation proceeds in “waves”, not all of the calls are independent and the construction of the discarded set is in Δ_2^{P}. We can then compare E^+ with X and return yes if they agree. $\square$

Proposition 67. *sub-Ver^{dis}_{st} is in Δ_2^{P}.*

Proof. One way of showing that X is contained in the discarded set is verifying that all arguments in X have all standard pd–evaluations blocked through the blocking set by E (see Lemma 14). It is easy to see we can focus on weak evaluations instead of only normal ones.

We consider the co–problem, i.e. whether there exists an argument in X that possesses a weak standard evaluation (F, B) s.t. $B \cap E = \emptyset$. We can guess a set of arguments Z containing at least one argument from X and an interpretation assignment $V = \{v_z \mid z \in Z\}$ s.t. for every v_z, $v_z^{\mathbf{t}} \subseteq Z$ and $v_z^{\mathbf{f}} \cap E = \emptyset$. Verifying that all of the interpretations are decisively in for their arguments is in coNP by Proposition 54. They can be easily formed into a standard evaluation in polynomial time. Therefore, our problem is in Π_2^{P} with this approach.

Another way of showing that X is contained in the standard discarded set is simply computing the actual set and verifying that X is contained in it. This puts our problem in Δ_2^{P} by Proposition 66, which is a better result. $\square$

Proposition 68. *Ver^{dis}_{par} is in Σ_2^{P}.*

Proof. Let v be an interpretation s.t. $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} = X$. By Lemma 20, v is the partially acyclic range interpretation of E iff all arguments in X are decisively out w.r.t. v and every $x \in A \setminus (E \cup X)$ has a partially acyclic evaluation (F^x, G^x, B^x) s.t. $B^x \cap E = \emptyset$ and $F^x \subseteq E$. We will use this to prove the membership of our problem.

Verifying that all arguments in X are decisively out w.r.t. v is in coNP by Proposition 55. Checking that no other argument qualifies for the discarded set can be done in the following manner. We can observe that if every argument in $A \setminus (E \cup X)$ possesses a partially acyclic evaluation meeting our requirements, then there exists an unblocked partially acyclic evaluation containing all of these arguments that meets our requirements as well. Moreover, if such an evaluation exists, then from it we can easily extract unblocked partially acyclic evaluations that satisfy our conditions. Without the loss of generality, we can focus on the weak evaluations. Therefore, no argument from $A \setminus (E \cup X)$ qualifies for the partially acyclic discarded set iff we can construct a partially acyclic evaluation (F, G, B) s.t. $F = E$, G a sequence of elements on $A \setminus (E \cup X)$ and $B \subseteq A \setminus E$. Hence, we guess an interpretation assignment for arguments in $F \cup A \setminus X$ s.t. the $\mathbf{f}$ part of every interpretation is in $A \setminus E$). Verifying that all interpretations are decisively in for their arguments is in coNP by Proposition 54. Ordering the elements from $A \setminus (E \cup X)$ into a suitable pd–sequence with $F = E$ as the pd–set can be done in polynomial time. Based on the guessed interpretations, it is easy to see that the blocking set requirements are also met. This, along with the previous decisiveness verification, puts our problem in Σ_2^{P}. □

Proposition 69. *$sub\text{-}Ver^{dis}_{par}$ is in Π_2^{P}.*

Proof. X is a subset of the partially acyclic range iff no argument in X has a partially acyclic evaluation that would be unblocked and with a pd–set in E (see Definition 17). It is easy to see we can take into account also weak evaluations instead of only normal ones. Consider the co–problem: there is an argument in X that possesses a partially acyclic evaluation (F, G, B) s.t. $F \subseteq E$ and $B \cap E = \emptyset$. We can guess a set of arguments Z containing at least one argument from X and an interpretation assignment s.t. for every interpretation v_z for $z \in Z$, $v_z^{\mathbf{f}} \subseteq A \setminus E$. Verifying that all of the interpretations are decisively in for their arguments is in coNP by Proposition 54. Checking that the interpretations can form a partially acyclic evaluation with a pd–set in E can be done in polynomial time. Moreover, the evaluation is unblocked by construction. Therefore, our problem is in Π_2^{P}. □

Proposition 70. *Ver^{dis}_{acy} is in $\mathrm{D}_2^{\mathsf{P}}$.*

Proof. In order to show that $X = E^{a+}$, it suffices to prove that $X \subseteq E^{a+}$ and that there is no argument in $x \in A \setminus (E \cup X)$ that would qualify for the discarded set, i.e. for every such x there is an acyclic pd–evaluation not blocked by E (see Lemma 16).

Verifying that $X \subseteq E^{a+}$ is in Π_2^{P} by Lemma 71. Now we need to check that for every argument $x \in A \setminus (E \cup X)$ there is a (weak) acyclic pd–evaluation with a blocking set disjoint from E. It is equivalent to checking if there is an acyclic pd–evaluation containing all the arguments from $A \setminus (E \cup X)$ that has a blocking set disjoint from X. We can guess a set F s.t. $A \setminus (E \cup X) \subseteq F$ and an interpretation assignment $V = \{v_f \mid f \in F\}$ s.t. for every v_f, $v_f^{\mathbf{f}} \subseteq A \setminus E$. Verifying that all of the interpretation are decisively in for their arguments is in coNP by Proposition 54. Checking that we can form a (weak) acyclic pd–evaluation with our assignments can be done in polynomial time. The evaluation is unblocked by construction. This part of the problem can be done in Σ_2^{P}. Therefore, verifying whether a given set is the acyclic discarded set is in $\mathrm{D}_2^{\mathsf{P}}$. $\square$

Proposition 71. *sub-Ver_{acy}^{dis} is in Π_2^{P}.*

Proof. We can observe that $X \subseteq E^{a+}$ iff all arguments in X have all acyclic pd–evaluations blocked through the blocking set by E (see Lemma 16). It is easy to see we can consider weak evaluations instead of only normal ones. We can consider the co–problem: there is an argument in X that possesses a (weak) acyclic pd–evaluation (F, B) s.t. $B \cap E = \emptyset$. We can guess a set of arguments Z containing at least one argument from X and an interpretation assignment for the elements in Z s.t. for every interpretation v_z for $z \in Z$, $v_z^{\mathbf{f}} \subseteq A \setminus E$. Verifying that all of the interpretations are decisively in for their arguments is in coNP by Proposition 54. Checking that we can form (weak) acyclic pd–evaluation can be done in polynomial time. The evaluation is unblocked by construction. Therefore, our problem is in Π_2^{P}. $\square$

Proposition 72. *acy-Ver_{acy}^{dis} is in Σ_2^{P}.*

Proof. Let v be an interpretation s.t. $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} = X$. By Lemma 22, v is the acyclic range interpretation of E iff all arguments in X are decisively out w.r.t. v and every $x \in A \setminus (E \cup X)$ has an acyclic pd–evaluation (F^x, B^x) s.t. $B^x \cap E = \emptyset$.

Verifying that all arguments in X are decisively out w.r.t. v in coNP (see Proposition 55). Independently of this, we can verify that no other argument qualifies for the discarded set. We can repeat the analysis of Proposition 70 in order to show that this problem is in Σ_2^{P}. Thus, the verification is in total in Σ_2^{P}. $\square$

5.3 Complexity Analysis of Conflict–free and Grounded Semantics

Proposition 73. *Ver_{cf} is in P.*

Proof. Create an interpretation v which maps arguments in E to $\mathbf{t}$ and arguments in $A \setminus E$ to $\mathbf{f}$. Verifying that the interpretation evaluates the conditions of arguments in E to in can be done in polynomial time. $\square$

Proposition 75. *$Ver_{acy\text{-}cf}$ is in D^{P}.*

Proof. In order to verify that E is a pd–acyclic conflict–free extension, we will use Theorem 74 and the approach from [BES$^+$13]. Constructing the reduct D^E can be done in polynomial time. Since the grounded extension and the grounded labeling correspond by Theorem 45 and in this case the $\mathbf{f}$ mappings are irrelevant, checking that E is the grounded extension is in D^{P} by [Wal14, Theorem 4.1.4]. $\square$

Proposition 76. *$Cred_{cf}$ is in NP.*

Proof. A conflict–free extension can be viewed as a standard evaluation that is not self–blocking. Consequently, asking whether there exists a conflict–free extension containing a given argument is the same as asking whether there is a non self–blocking standard pd–evaluation containing this argument, which is in NP by Proposition 63. □

Proposition 77. $Cred_{acy\text{-}cf}$ *is in* Σ_2^{P}.

Proof. A pd–acyclic conflict–free extension can be viewed as an acyclic evaluation that is not self–blocking. Consequently, asking whether there exists a pd–acyclic conflict–free extension containing a given argument is the same as asking whether there is a non self–blocking acyclic pd–evaluation containing this argument. Thus, based on Proposition 64, our problem is in Σ_2^{P}. □

Proposition 78. Ver_{grd} *is in* D^{P}.

Proof. Let $D = (A, C)$ be an ADF and $E \subseteq A$ the set we want to verify. Based on Theorem 45, we can observe that the grounded extension and the grounded labeling of D correspond. Therefore, in order to verify that E is the grounded extension of D, we exploit the problem of credulous acceptance $Cred_{grd-lab}$ in the grounded labelings, which is in coNP. For every argument $a \in E$, we want $Cred_{grd-lab}(D, a) = yes$ and for every $b \in A \setminus E$, we want $Cred_{grd-lab}(D, a) = no$. Due to the fact that both coNP and NP classes are closed under conjunction, it means that the first issue lies in coNP while the other in NP. Consequently, our problem is in D^{P}. □

Proposition 79. $Ver_{acy\text{-}grd}$ *is in* Δ_3^{P}.

Proof. We can compute the actual grounded extension and compare it with the set we want to verify. In order to get the real set, we will follow the algorithm from Definition 29. We start with an empty interpretation v and map to $\mathbf{t}$ those arguments that are decisively in w.r.t. v and to $\mathbf{f}$ those that have all of their acyclic evaluations blocked by v. We then perform the search again w.r.t. the extended v and repeat the procedure until no further arguments can be added. Thus, we perform polynomially many steps in which every argument not covered by the range is checked for being decisively in (this is in coNP by Proposition 52) or for belonging to the acyclic discarded set (this is in Π_2^{P} by Lemma 71). Therefore, the algorithm from Definition 29 is in Δ_3^{P}. □

Proposition 80. $Cred_{grd}$ *is coNP–complete.*

Proof. By Theorem 45, an argument is credulously accepted in a grounded extension iff it is credulously accepted in a grounded labeling. Thus, the complexity follows from [Wal14, Proposition 4.1.3]. Since grounded semantics always produces only one extension, the credulous and skeptical problems are the same and are coNP–complete. □

Proposition 81. $Cred_{acy\text{-}grd}$ *is in* Π_2^{P}.

Proof. By Theorem 38, the acyclic grounded extension is the least aa–complete extension. Therefore, in order to verify whether an argument $a \in A$ is skeptically accepted w.r.t. the acyclic grounded semantics, we can check if it is accepted in every aa–complete extension. We can consider the co–problem, i.e. whether there exists an aa–complete extension $E \subseteq A$ s.t. $a \notin E$. We can adapt the proof of Proposition 94 in a way that not only the discarded set, but also the accepted set E is guessed. Therefore, verifying that there is an appropriate aa–complete extension is in Σ_2^{P}, which puts our original problem in Π_2^{P}. □

5.4 Complexity Analysis of Admissible Semantics

Proposition 82. *$Ver_{cc\text{-}adm}$ is in Δ_2^{P}.*

Proof. First, we need to verify that E is even conflict–free. This problem is in P by Proposition 73. Computing the standard range requires polynomially many calls to a coNP oracle and is in Δ_2^{P} by the proof of Proposition 66. By Proposition 54, verifying that all arguments in E are decisively in w.r.t. the standard range simply adds one more call to the coNP oracle. This puts our problem in Δ_2^{P}. □

Proposition 83. *$Ver_{ac\text{-}adm}$ is in Σ_2^{P}.*

Proof. We guess an interpretation assignment for every argument in E s.t. the $\mathbf{t}$ mappings are in E and the $\mathbf{f}$ mappings are outside E. Verifying every interpretation is decisively in for the argument it is assigned to is in coNP by Proposition 54. Due to the decisiveness and the construction of these interpretations, it is easy to see that E is conflict–free. Computing the standard discarded set requires polynomially many calls to a coNP oracle (see the proof of Proposition 66). Finally, checking whether we can form an acyclic pd–evaluation with these interpretations and verifying that the $\mathbf{f}$ part of every interpretation (i.e. the blocking set) is contained in the discarded set we have just computed can be done in polynomial time. Therefore, our problem is in Σ_2^{P}. □

Proposition 84. *$Ver_{aa\text{-}adm}$ is in Σ_2^{P}.*

Proof. Let us guess an interpretation assignment s.t. every argument $x \in E$ is associated with a partial interpretation v_x s.t. $v_x^{\mathbf{t}} \subseteq E$ and $v_x^{\mathbf{f}} \cap E = \emptyset$. Verifying that all of the interpretations are decisively in for their arguments is in coNP by Proposition 54. Checking that the interpretations form a (weak) acyclic pd–evaluation can be done in polynomial time. By construction, this evaluation is not self–blocking, which means that our set is pd–acyclic conflict–free. We can now create a two–valued interpretation v s.t. $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} = \bigcup v_x^{\mathbf{f}}$. By Lemma 21, it suffices to show that every argument in $v^{\mathbf{f}}$ is decisively out w.r.t. v, which can be done in coNP by Proposition 53. This puts our problem in Σ_2^{P}. □

Proposition 85. *$Ver_{ca_1\text{-}adm}$ is in Δ_3^{P}.*

Proof. We first need to verify that E is even conflict–free. This problem is in P by Proposition 73. We then need $\mid A \setminus E \mid$ many calls to a Π_2^{P} oracle in order to determine which arguments from $A \setminus E$ are in the acyclic discarded set (we simply use Proposition 71 for all single–argument sets from $A \setminus E$). We then need one call to a coNP oracle in order to verify that all arguments that are in E, are decisively in w.r.t. the acyclic range (see Proposition 52. This puts our problem in Δ_3^{P}. □

Proposition 86. *$Ver_{ca_2\text{-}adm}$ is in Σ_2^{P}.*

Proof. We will exploit the relation between ca$_2$–admissible extensions and admissible labelings from Theorem 42. In other words, E is ca$_2$–admissible iff there exists a corresponding admissible labeling. Therefore, we can guess a set $X \subseteq A \setminus E$ and create a three valued interpretation v s.t. $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} = X$. Verifying that this interpretation is an admissible labeling can be done in coNP by [Wal14, Proposition 4.1.9]. Therefore, our problem is in Σ_2^{P}. □

Proposition 87. *$Cred_{cc\text{-}adm}$, $Cred_{ac\text{-}adm}$, $Cred_{aa\text{-}adm}$ and $Cred_{ca_2\text{-}adm}$ are in Σ_2^{P}.*

Proof. Let a be the desired argument. In the case of aa–admissible semantics, we adapt proof of Proposition 84 in order to show that the credulous reasoning is in Σ_2^{P}. The guess for interpretation assignment is simply extended in order to include a set of arguments $E \subseteq A$ s.t. $a \in E$. Therefore, the problem remains in Σ_2^{P}. The same follows for the ac–admissible semantics.

In the case of cc–semantics, we can guess a set of arguments E s.t. $a \in E$. Verifying that it is conflict–free is in P. Computing the standard discarded set E^+ requires polynomially many calls to a coNP oracle (see the proof of Proposition 66). We now have the standard range at hand as well. Checking that this range is a decisively in interpretation for all arguments in E is in coNP by Proposition 54. Thus, the credulous reasoning for the cc–admissible semantics is in Σ_2^{P}.

Due to the relation between the ca_2–admissible semantics and admissible labelings stated in Theorem 42, the problem of credulous reasoning for both of the semantics is the same and is in Σ_2^{P} by [Wal14, Proposition 4.1.11]. □

Proposition 88. *$Cred_{ca_1\text{-}adm}$ is in Σ_3^{P}.*

Proof. In order to show that the credulous reasoning for ca_1–admissible semantics is in Σ_3^{P}, we do the following: we can guess a set of arguments X containing the desired argument and an interpretation assignment for this set s.t. for every $x \in X$, $v_x^{\mathbf{t}} \subseteq X$ and $v_x^{\mathbf{f}} \cap X = \emptyset$. This ensures conflict–freeness. Verifying that all of the interpretations are decisively in for their arguments is in coNP by Proposition 54. Checking that we can form a standard evaluation with our assignment can be done in polynomial time. Verifying that its blocking set is a subset of the acyclic discarded set is in Π_2^{P} by Lemma 71. Consequently, our problem is in Σ_3^{P}. □

Lemma 89. *Let X, Y be two disjoint sets of propositional variables, and ψ a Boolean formula over $X \cup Y$. Define the ADF $D_\psi = (S, C)$ as follows.*

- *$S = X \cup \overline{X} \cup Y \cup \{f\}$;*
- *$\varphi_x = \neg\overline{x}$ for $x \in X$;*
- *$\varphi_{\overline{x}} = \neg x$ for $\overline{x} \in \overline{X}$; and*
- *$\varphi_y = \neg y$ for $y \in Y$; and*
- *$\varphi_f = \psi$.*

The ADF D_ψ is an AADF$^+$.

Proof. Every argument $x_i \in X$ has exactly one minimal decisively in interpretation $\{\overline{x_i} : \mathbf{f}\}$. Similarly, for every $\overline{x_i} \in \overline{X}$ we have $\{x_i : \mathbf{f}\}$ and for every $y_i \in Y$, $\{f : \mathbf{f}, y_i : \mathbf{f}\}$. Consequently, all of these arguments satisfy the starting argument requirement for acyclic pd–evaluations and any standard evaluations involving only them can be made acyclic by putting the pd–set arguments into a sequence in any order. The f argument can have an arbitrary number of decisively in interpretations - it can even not possess one at all. However, any decisively in interpretation, provided it exists, maps to $\mathbf{t}$ only $x_i, \overline{x_i}, y_i$ arguments. In other words, f does not possess a minimal decisively in interpretation that would map f itself to $\mathbf{t}$. Since we know that all of the $x_i, \overline{x_i}, y_i$ arguments satisfy the starting argument requirement, any standard evaluation in D_ψ containing f in its pd–set can be reordered into an acyclic one by setting $a_n = f$ and putting all remaining arguments in arbitrary order. In conclusion, every standard evaluation in D_ψ can be made acyclic and D_ψ is an AADF$^+$. $\square$

Proposition 90. *Let $x, y \in \{a, c\}$. $Cred_{xy\text{-}adm}$ is Σ_2^{P}–hard.*

Proof. Now, we will re–adapt the hardness proof of [Wal14, Proposition 4.1.11]) for our purposes. We provide a reduction from the $QBF_{\exists,w} - VALIDITY$. Let $\phi \in QBF_{\exists,2}$ be a closed QBF of the form $\phi = \exists X \forall Y \psi$. Construct an ADF $D_\psi = (X \cup \overline{X} \cup Y, C)$ for ψ described by Lemma 89 and depicted in Figure 6. The ADF D_ψ can be constructed in polynomial time w.r.t. the size of ϕ. We exploit the fact that D_ψ is an AADF$^+$ and thus all aa/ac/cc/ca$_1$ and ca$_2$–admissible extensions coincide according to Theorem 48. We now prove that $Cred_{xy-adm}(f, D) = yes$ iff ϕ is valid.

Assume that $Cred_{xy-adm}(f, D) = yes$. Then there exists an xy–admissible extension $E \subseteq X \cup \overline{X}$ s.t. $f \in E$. Since D is an AADF$^+$, we can focus on cc–admissibility only. We can observe that none of the y_i can be in the discarded set of E – every one of them has a simple standard evaluation $(\{y_i\}, \{y_i\})$, and as $y_i \notin E$, the evaluation remains unblocked (see Lemma 14). Since f is decisively in w.r.t. the standard range of E, it means that any completion of the range satisfies the condition of f. As only elements of X and Y can be parents of f, this means that the condition ψ of f is satisfied by any set $(E \cap X) \cup Y'$, where $Y' \subseteq Y$. Thus, $(E \cap X) \cup Y' \models \psi$ and as it holds for arbitrary $Y' \subseteq Y$, ϕ is valid.

Now assume that ϕ is valid. This means there exists a set $X' \subseteq X$ s.t. for any $Y' \subseteq Y$ we have $X' \cup Y' \models \psi$. We now show that $E = X' \cup \{\overline{x_i} \mid x_i \notin X'\} \cup \{f\}$ is xy–admissible in D_ψ. Since D_ψ is an AADF$^+$, it suffices to focus on the cc–admissible case. First, let us focus on $E' = E \setminus \{f\}$. Since we only accept $\overline{x_i}$ in E if x_i is not present, we can conclude that all arguments in E' have their acceptance conditions satisfied and that the set is conflict–free. We can observe that for any $x_i \in E'$, $\overline{x_i} \in E'^+$ and for any $\overline{x_i} \in E'$, $x_i \in E'^+$. Thus, the decisiveness of arguments in E' w.r.t. the standard range follows easily and we can conclude that the set is cc–admissible. Let us now focus on f. Due to the validity of ϕ and the fact that only elements of X are parents of f, condition ψ is satisfied by E'. As only remaining parents of f are in Y and as f is in w.r.t. $E' \cup Y'$ for any $Y' \subseteq Y$ due to the validity of ϕ, then ψ is in w.r.t. any completion of the range of E' and we can conclude that f is decisively in w.r.t. the range of E'. Thus, by Fundamental Lemma 35, $E = E' \cup \{f\}$ is cc–admissible.

We can finally conclude that the credulous reasoning under xy–admissible semantics is Σ_2^{P}–hard. $\square$

Proposition 91. *$Cred_{cc\text{-}adm}$, $Cred_{ac\text{-}adm}$, $Cred_{aa\text{-}adm}$ and $Cred_{ca_2\text{-}adm}$ are Σ_2^{P}–complete.*

Proof. Σ_2^{P}–completeness for the credulous reasoning under cc, ac, aa and ca$_2$–admissible semantics follows from Proposition 87 and Proposition 90. $\square$

5.5 Complexity Analysis of Complete Semantics

Proposition 92. *$Ver_{cc\text{-}cmp}$ is in Δ_2^{P}.*

Proof. We first need to show that E is cc–admissible. As seen in the proof of Proposition 82, we start by verifying that E is conflict–free (which is in P by Proposition 73) and then need polynomially many calls to a coNP oracle in order to compute the standard range (Proposition 66) and to check that all arguments in E are decisively in w.r.t. to it (Proposition 54). The proof needs to be extended with checking that there is no other argument that is decisively in w.r.t. the range, which requires a call to an NP oracle by Proposition 56. This puts our problem in Δ_2^{P}. $\square$

Proposition 93. *$Ver_{ac\text{-}cmp}$ is in Σ_2^{P}.*

Proof. We first need to verify that E is ac–admissible. We start by guessing an interpretation assignment $V = \{v_x \mid x \in E\}$ s.t. for every v_x, $v_x^{\mathbf{t}} \subseteq E$ and $v_x^{\mathbf{f}} \subseteq A \setminus E$. Verifying that these interpretations are decisively in for their argument is in coNP by Proposition 54. This also ensures conflict–freeness. We then need polynomially many calls to a coNP oracle in order to compute the standard discarded set and verify the decisiveness of the accepted arguments w.r.t. this range. Checking if our interpretations form an acyclic pd–evaluation that meets our requirements is done in polynomial time. In order to prove completeness, we need one call to an NP oracle in order to check that there is no other argument that is decisively in w.r.t. the range (see Proposition 56). Our problem is thus still in Σ_2^{P}. $\square$

Proposition 94. *$Ver_{aa\text{-}cmp}$ is in Σ_2^{P}.*

Proof. Let E be the set of arguments we want to verify. We guess a set of arguments $T \subseteq A \setminus E$ and an interpretation assignment $V = \{v_x \mid x \in A\}$ s.t. for every $x \in E$, $v_x^{\mathbf{t}} \subseteq E$ and $v_x^{\mathbf{f}} = T$, and for every $x \notin E \cup T$, $v_x^{\mathbf{t}} \subseteq A \setminus T$ and $v_x^{\mathbf{f}} \subseteq A \setminus E$. Verifying that all of the interpretations are decisively in for their arguments is in coNP by Proposition 54. Checking that we can form an acyclic pd–evaluation with the assignments for arguments in E can be done in polynomial time. The same holds for the assignments for arguments in $A \setminus T$. Based on the construction, we can observe that the evaluation created for arguments in E cannot be self–blocking. Thus, E is pd–acyclic conflict–free. We now construct an interpretation v s.t. $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} = T$. By Lemma 21, in order to show that our set is aa–admissible, it suffices to check that every argument in $v^{\mathbf{f}}$ is decisively out w.r.t. v. This is in coNP by Proposition 53. In order to prove aa–completeness, we need to check that no argument outside of $v^{\mathbf{t}} \cup v^{\mathbf{f}}$ should be included in it. Verifying that no argument outside the range should be mapped to $\mathbf{t}$ requires one call to an NP oracle (see Proposition 56). The fact that no argument outside the range should be mapped to $\mathbf{f}$ is already proved by the existence of an acyclic pd–evaluation for arguments in $A \setminus T$. Thus, our problem is in Σ_2^{P}. $\square$

Proposition 95. *$Ver_{ca_1\text{-}cmp}$ is in Δ_3^{P}.*

Proof. We first verify that E is ca$_1$–admissible. As seen in the proof of Proposition 85, we start by making sure that E is conflict–free, which is in P. We then need $\mid A \setminus E \mid$ many calls to a Π_2^{P} oracle in order to determine which arguments from $A \setminus E$ are in the discarded set (we simply use

Proposition 71 for all single–argument sets from $A \setminus E$). We then need one call to a coNP oracle in order to verify that all arguments in E are decisively in w.r.t. the acyclic range (see Proposition 52. We then need a call to an NP oracle in order to check that no other argument is decisively in w.r.t. the range (see Proposition 56). Therefore, the verification problem still remains in Δ_3^{P}. □

Proposition 96. *$Ver_{ca_2\text{-}cmp}$ is in Σ_2^{P}.*

Proof. Let E be the set of arguments we want to verify. We guess a set of arguments $T \subseteq A \setminus E$ and an interpretation assignment $V = \{v_x \mid x \in A\}$ s.t. for every $x \in E$, $v_x^{\mathbf{t}} = E$ and $v_x^{\mathbf{f}} = T$, and for every $x \notin E \cup T$, $v_x^{\mathbf{t}} \subseteq A \setminus T$ and $v_x^{\mathbf{f}} \subseteq A \setminus E$. Verifying that all of the interpretations are decisively in for their arguments is in coNP by Proposition 54. This also shows that E is conflict–free and that every $x \in E$ is decisively in w.r.t. an interpretation v s.t. $v^{\mathbf{t}} = E$ and $v^{\mathbf{f}} = T$. By Lemma 19, in order to show that our set is ca$_2$–admissible, it suffices to check that every argument in $v^{\mathbf{f}}$ is decisively out w.r.t. v. This is in coNP by Proposition 53. In order to prove ca$_2$–completeness, we need to check that no argument outside of the range should be included in it. Checking that no argument outside the range is decisively in w.r.t. this range requires one call to an NP oracle (see Proposition 56). The fact that no argument outside the range should be mapped to **f** is proved by showing that with the guessed interpretations, we can order arguments in $A \setminus (E \cup T)$ into a sequence G s.t. $(E, G, \bigcup_{x \in A \setminus T} v_x^{\mathbf{f}})$ is a partially acyclic pd–evaluation. Since this can be done in polynomial time, our verification problem is in Σ_2^{P}. □

Proposition 97. *$Cred_{cc\text{-}cmp}$, $Cred_{ac\text{-}cmp}$ are $Cred_{aa\text{-}cmp}$ are Σ_2^{P}–complete.*

Proof. Follows from Theorems 37, 38 and 91. □

Proposition 98. *$Cred_{ca_2\text{-}cmp}$ is Σ_2^{P}–complete. $Cred_{ca_1\text{-}cmp}$ is in Σ_3^{P} and is Σ_2^{P}–hard.*

Proof. Follows from Theorems 38 and 39, Propositions 88, 90 and 91, and the fact that we are dealing with finite frameworks. □

Proposition 99. *$Skept_{cc\text{-}cmp}$ and $Skept_{ac\text{-}cmp}$ are coNP–complete.*

Proof. Since the grounded extension is the least ac and cc–complete one by Theorem 38, the skeptical reasoning for these semantics is the same as for the grounded one and is coNP–complete by Proposition 80. □

Proposition 100. *$Skept_{aa\text{-}cmp}$ is in Π_2^{P}.*

Proof. Since the acyclic grounded extension is the least aa–complete one by Theorem 38, the skeptical reasoning for this semantics is the same as for the acyclic grounded one and is in Π_2^{P} by Proposition 81.

□

Proposition 101. *$Skept_{ca_1\text{-}cmp}$ is in Π_3^{P}.*

Proof. Let $a \in A$ be the argument we want to check for skeptical acceptance. In other words, we want to verify if all ca$_1$–complete extensions contain a. Let us now focus on the co–problem, i.e. existence of a ca$_1$–complete extension that does not include a.

We first guess a set of arguments $E \subseteq A \setminus \{a\}$ and verify that it is conflict–free, which is in P. We then need $\mid A \setminus E \mid$ many calls to a Π_2^{P} oracle in order to determine which arguments from $A \setminus E$ are in the discarded set (we simply use Proposition 71 for all single–argument sets from $A \setminus E$). We then need one call to a coNP oracle in order to verify that all arguments in E are decisively in w.r.t. the acyclic range (see Proposition 52) and call to an NP oracle in order to check that no other argument is decisively in w.r.t. the range (see Proposition 56).

Our co–problem is in Σ_3^{P}; hence, the main one is in Π_3^{P}. □

Proposition 102. *$Skept_{ca_2\text{-}cmp}$ is in Π_2^{P}.*

Proof. Let $a \in A$ be the argument we want to consider. We will focus on the co–problem, i.e. existence of a ca$_2$–complete extension that does not contain a. This can be done by modifying the guess in the proof of Proposition 96. In this case, we not only guess a set T, but also a set E s.t. $a \notin E$. Therefore, our co–problem is in Σ_2^{P}, which puts the main issue in Π_2^{P}. □

5.6 Complexity Analysis of Preferred Semantics

Proposition 103. *$Ver_{cc\text{-}prf}$ is in Π_2^{P}.*

Proof. Verifying that E is cc–admissible is in Δ_2^{P} by Proposition 82. Now, it is cc–preferred iff there is no other cc–admissible extension that would contain it. Let us now consider the co–problem, i.e. there exists a cc–admissible extension E' s.t. $E \subset E'$. We can guess a set of arguments E' along with the interpretation assignment and proceed as in the proof of Proposition 87 to verify that E' is cc–admissible, which puts this part of the problem in Π_2^{P}. Since $\Delta_2^{\mathsf{P}} \subseteq \Pi_2^{\mathsf{P}}$, the verification problem for the cc–preferred semantics is in Π_2^{P}. □

Proposition 104. *$Ver_{ac\text{-}prf}$, $Ver_{aa\text{-}prf}$ and $Ver_{ca_2\text{-}prf}$ are in $\mathrm{D}_2^{\mathsf{P}}$.*

Proof. Let $x \in \{ac, aa, ca_2\}$. Verifying that E is x–admissible can be done in Σ_2^{P} by Propositions 83, 84 and 86. It is x–preferred iff there is no other x–admissible extension that would contain it. Let us now consider the co–problem, i.e. there exists an x–admissible extension E' s.t. $E \subset E'$. We can guess a set of arguments E' along with the interpretation assignment and proceed as in the proof of Proposition 87 to verify that E' is ac/aa/ca$_2$–admissible. Thus, this part of the problem is in Π_2^{P} and the general verification problem is in $\mathrm{D}_2^{\mathsf{P}}$. □

Proposition 105. *$Ver_{ca_1\text{-}prf}$ is in Π_3^{P}.*

Proof. Verifying that E is ca$_1$–admissible is in Δ_3^{P} by Proposition 85. It is ca$_1$–preferred iff there is no other ca$_1$–admissible extension that would contain it. Let us now consider the co–problem, i.e. there exists a ca$_1$–admissible extension E' s.t. $E \subset E'$. We can guess a set of arguments E' along with the interpretation assignment and proceed as in the proof of Proposition 88 to verify that E' is ca$_1$–admissible, which puts this part of the problem in Π_3^{P}. Thus, the verification problem is in Π_3^{P}. □

Proposition 106. *$Skept_{cc\text{-}prf}$ is in Π_3^{P}.*

Proof. In order to answer yes we need to check if every cc–preferred extension contains a given argument. Let us consider the co–problem, i.e. there exists a cc–preferred extension E' which does not contain the desired argument. Let a be the argument in question. We can guess a set of arguments $E' \subseteq A \setminus \{a\}$ that does not contain a; verifying it is indeed cc–preferred is in Π_2^{P} by Proposition 103. This puts our problem in Π_3^{P}. $\square$

Proposition 107. *$Skept_{ac\text{-}prf}$, $Skept_{aa\text{-}prf}$ and $Skept_{ca_2\text{-}prf}$ are in Π_3^{P}.*

Proof. Let us start with the ac–preferred semantics. We consider the co–problem, i.e. deciding whether there exists an ac–preferred extension E that does not contain the desired argument, say $a \in A$. We guess a set $E \subseteq A \setminus \{a\}$ and an interpretation assignment $V = \{v_x \mid x \in E\}$ s.t. $v_x^{\mathbf{t}} \subseteq E$ and $v_x^{\mathbf{f}} \subseteq A \setminus E$. We then require polynomially many calls to a coNP oracle in order to verify that E is an ac–admissible extension (see the proof of Proposition 83). In order to verify that it is ac–preferred, we consider the co–problem, i.e. we check if there exists an ac–admissible extension E' s.t. $E \subset E'$. Again, this can be done by guessing a set and an interpretation assignment and proceeding as in the previous step. Hence, the problem of skeptical reasoning for ac–preferred semantics is in Π_3^{P}.

The same analysis can be carried out for the aa–preferred semantics, thus putting this skeptical reasoning problem in Π_3^{P} as well. We are left with the ca$_2$–approach. We consider the co-problem again. First, we guess a partial two–valued interpretation v s.t. $v(a) \neq \mathbf{t}$. We exploit the relation between the ca$_2$–admissible extensions and admissible labelings (see Theorem 42) in order to show that $v^{\mathbf{t}}$ is ca$_2$–admissible, which requires a call to coNP oracle by [Wal14, Proposition 4.1.9]. In order to show that $v^{\mathbf{t}}$ is ca$_2$–preferred, we guess another interpretation z s.t. $v^{\mathbf{t}} \subset z^{\mathbf{t}}$ and proceed as before. Thus, verifying that there exists a ca$_2$–preferred extension not containing the argument a is in Σ_3^{P}, which puts the skeptical reasoning problem for this semantics in Π_3^{P}. $\square$

Proposition 108. *$Skept_{ca_1\text{-}prf}$ is in Π_4^{P}.*

Proof. In order to analyze the complexity of verifying that every ca$_1$–preferred extension contains the desired argument, let us consider the co–problem, i.e. deciding whether there exists a ca$_1$–preferred extension E' which does not contain the desired argument. Let a be the argument in question. We can guess a set of arguments $E' \subseteq A \setminus \{a\}$ and an interpretation assignment for E'. Verifying that this guess qualifies for a ca$_1$–admissible extension is in Π_2^{P} (see the proof of Proposition 88). Checking that it is in indeed ca$_1$–preferred (i.e. that there is no ca$_1$–admissible extension greater than E') is in Π_3^{P} (see the proof of Proposition 105). Thus, the skeptical reasoning for ca$_1$–preferred semantics is in Π_4^{P}. $\square$

Lemma 109. *Let $\phi \in QBF_{\forall,3}$ be a closed QBF of the form $\phi = \forall X \exists Y \forall Z \psi$. We define the ADF $D_{pref(\psi)}$ with $S = X \cup \overline{X} \cup Y \cup \overline{Y} \cup D \cup \overline{D} \cup Z \cup \{f\}$ and the acceptance conditions as follows:*

- $\varphi_{x_i} = \neg\overline{x_i}$ *for* $x_i \in X$
- $\varphi_{\overline{x_i}} = \neg x_i$ *for* $\overline{x_i} \in \overline{X}$

- $\varphi_{d_i} = \neg f$ *for* $d_i \in D$
- $\varphi_{\overline{d_i}} = \neg f$ *for* $\overline{d_i} \in \overline{D}$
- $\varphi_{y_i} = \neg d_i \wedge \neg \overline{y_i}$ *for* $y_i \in Y$
- $\varphi_{\overline{y_i}} = \neg \overline{d_i} \wedge \neg y_i$ *for* $\overline{y_i} \in \overline{Y}$
- $\varphi_{z_i} = \neg z_i$ *for* $z_i \in Z$
- $\varphi_f = \neg f \vee \psi$

Then $D_{pref(\psi)}$ *is an AADF*$^+$*.*

Proof. Straightforward: see the proof of Lemma 89 for details. All arguments in $X, \overline{X}, Y, \overline{Y}, D, \overline{D}$ and Z have a single minimal decisively in interpretation that maps parents of a given argument to $\mathbf{f}$. Consequently, they have trivial acyclic evaluations. Argument f possesses at least one minimal decisively in interpretation mapping f to $\mathbf{f}$. Consequently, it also has a trivial acyclic evaluation. Analysis of the other interpretations follows the same line of reasoning as in Lemma 89. □

Proposition 110. $Skept_{xy-pref}$ *is* Π_3^{P}*–hard for ADFs for* $x, y \in \{a, c\}$*.*

Proof. We re–adapt the hardness proof of [Wal14, Theorem 4.1.17] for our purposes. Consider a reduction from a closed QBF $\phi = \forall X \exists Y \forall Z \psi$, the construct $D_{pref(\phi)}$ as in Lemma 109 s.t. $D = \{d_i \mid y_i \in Y\}$ and $\overline{D} = \{\overline{d_i} \mid \overline{y_i} \in \overline{Y}\}$. We know prove that $Skept_{xy-pref}(f, D_{pref(\phi)}) = yes$ iff ϕ is valid.

Assume that ϕ is valid. Consider an arbitrary $X' \subseteq X$. Since ϕ is valid, we know that there is a $Y' \subseteq Y$ s.t. for any $Z' \subseteq Z$ we have $X' \cup Y' \cup Z' \models \psi$. Let $E = X' \cup \{\overline{x_i} \mid x_i \notin X'\} \cup Y' \cup \{\overline{y_i} \mid y_i \notin Y'\} \cup \{f\}$ be a set of arguments. We will show that E is xy–preferred. Since $D_{pref(\phi)}$ is an AADF$^+$ by Lemma 109, all xy–preferred extensions coincide and it suffices to focus on the cc case. We first show it is conflict–free. Since $X' \cup Y' \cup Z' \models \psi$ for arbitrary $Z' \subseteq Z$, then $X' \cup Y' \models \psi$ and thus the acceptance condition of f is easily satisfied w.r.t. E. As no parents of $x_i, \overline{x_i}, y_i, \overline{y_i} \in E$ are present in E, their conditions are also satisfied. Thus, E is conflict–free. Let us now focus on admissibility. Due to the presence of f in E, all $d_i, \overline{d_i}$ are in the discarded set. Similarly, for any x_i in E, $\overline{x_i}$ is in the discarded set, for any $\overline{x_i} \in E$, x_i is in the discarded set, for any $y_i \in E$, $\overline{y_i}$ is in the discarded set, and for any $\overline{y_i} \in E$, y_i is in the discarded set. Therefore, the decisiveness of these arguments follows easily. We can observe that only z_i arguments are not in the discarded set of E (their conditions are satisfied w.r.t. E). However, as $X' \cup Y' \cup Z' \models \psi$ for arbitrary $Z' \subseteq Z$, then it follows that any completion of the standard range will not change the outcome of the acceptance condition of f. Therefore, f is also decisively in w.r.t. the range and E is cc–admissible. By this analysis we can also observe that it is cc–preferred; all arguments in $D_{pref(\phi)}$ that are not in E and Z are in the discarded set. Thus, their conditions will evaluate to out w.r.t. any set containing E by Proposition 33 and Lemma 34. Moreover, no set containing z_i can be conflict–free. Consequently, no set E' s.t. $E \subset E'$ can possibly be conflict–free, let alone cc–preferred.

Now we need to show that all sets of arguments E' s.t. $E' \cap X = X'$ either contain f or are not cc–preferred. Thus, supposed that E' does not contain f and is cc–preferred. We can observe that as long as f is not presented in a given set of arguments, its condition is in w.r.t. this set. Consequently, it cannot be the case that f is in the discarded set of E'. From this follows that no d_i and $\overline{d_i}$ can possibly be decisively in w.r.t. the range of E' – therefore, if they are present in the set, it cannot be even cc–admissible. They also cannot be decisively out w.r.t. the range of E' – as long as f is not present in the set, their conditions are in. Consequently, it is not possible for E' to contain any of the y_i, $\overline{y_i}$, as the completions of range evaluate their conditions to different values depending on what is assigned to d_i's and $\overline{d_i}$'s. Additionally, none of the z_i can possibly be in the standard range – a set that does not contain these arguments evaluates to true, thus decisive outing is not possible, and a set that does contain them cannot be conflict–free. Therefore, it is only possible that E' contains some of the arguments from $\overline{X}$. We can observe that $E' = X'$ itself is easily cc–admissible; every $\overline{x_i}$ for $x_i \in E'$ is trivially in the discarded set. If $\overline{x_i} \in E'$ when $x_i \in E'$, we breach conflict–freeness of E'. If $x_i \notin E'$ and $\overline{x_i} \notin E'$, then x_i is in the discarded set of $E'' = E' \cup \{\overline{x_i}\}$, which makes E'' cc–admissible. Since $E' \subset E''$, it cannot be the case that E' is cc–preferred. This brings us to the conclusion that E' is of the form $X' \cup \{\overline{x_i} \mid x_i \notin X'\}$. However, E' is clearly a subset of the cc–preferred extension E described before. Consequently, it cannot be the case that E' does not contain f and is cc–preferred. We can conclude that if ϕ is valid, then $Skept_{xy-pref}(f, D_{pref(\phi)}) = yes$.

Now assume that $Skept_{xy-pref}(f, D_{pref(\phi)}) = yes$. Let us now consider how the preferred extensions look like in our framework. An arbitrary set $X' \subseteq X$ is cc–admissible; it is conflict–free, and every x_i has the power to decisively out its $\overline{x_i}$. We can give the same analysis for $\overline{X'} \subseteq \overline{X}$. Consequently, it is easy to see that a set $X' \cup \{\overline{x_i} \mid x_i \notin X'\}$ will also be cc-admissible. By this and Theorems 37 and 39 it follows that there exists a cc–preferred extension E s.t. $E \cap X = X'$ for arbitrary X'. As any cc–preferred extension E also contains f, it means that arguments in $D \cup \overline{D}$ will be in the discarded set of E. Now assume that for a given i, neither y_i nor $\overline{y_i}$ are in E. We can observe that $E' = E \cup \{\overline{y_i}\}$ is cc–admissible; it is easily conflict–free, and by Lemma 34 the arguments in E are decisively in w.r.t. the range of E'. Moreover, by this lemma arguments in $D \cup \overline{D}$ are also in the discarded set and due to the presence of $\overline{y_i}$, so is y_i. Consequently, $\overline{y_i}$ has to be decisively in w.r.t. the range of E'. Consequently, if E does not contain y_i, then $E \cup \{\overline{y_i}\}$ is cc–admissible, making it impossible for E to be cc–preferred in the first place. Similar analysis can be done for y_i, assuming its addition does not affect f. The point is, however, that for any cc–preferred extension the standard range is defined for arguments in Y and $\overline{Y}$. Let again $X' \subseteq X$ be an arbitrary set of arguments . There exists a cc–preferred extension E s.t. $E \cap X = X'$. Let $Y' = E \cap Y$. Since the range of E cannot be defined for any z_i and f is still decisively in w.r.t. the range, we can conclude that $X' \cup Y' \cup Z' \cup \{f\} \models C_f$ for arbitrary $Z' \subseteq Z$. Consequently, $X' \cup Y' \cup Z' \models \psi$, and as it holds for arbitrary X', ϕ is valid. □

Proposition 111. *$Skept_{cc\text{-}prf}$, $Skept_{ac\text{-}prf}$, $Skept_{aa\text{-}prf}$ and $Skept_{ca_2\text{-}prf}$ are Π_3^{P}–complete. $Skept_{ca_1\text{-}prf}$ is Π_3^{P}–hard and in Π_4^{P}.*

Proof. Follows from Propositions 106 and 107 and Proposition 110. □

Proposition 112. *$Cred_{cc\text{-}prf}$, $Cred_{ac\text{-}prf}$ and $Cred_{aa\text{-}prf}$ are Σ_2^{P}–complete.*

Proof. Follows from Theorem 37 and Proposition 91. □

Proposition 113. *$Cred_{ca_2\text{-}prf}$ is Σ_2^{P}–complete. $Cred_{ca_1\text{-}prf}$ is in Σ_3^{P} and is Σ_2^{P}–hard.*

Proof. Follows from Theorems 38 and 39, Propositions 88, 90 and 91, and the fact that we are dealing with finite frameworks. □