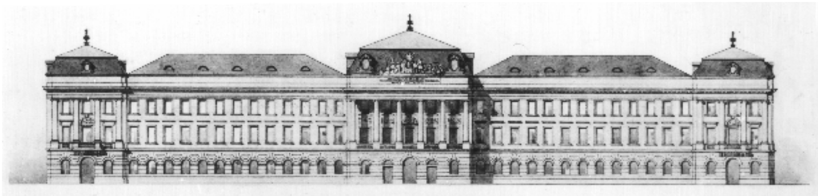
dbai

TECHNICAL REPORT



INSTITUT FÜR LOGIC AND COMPUTATION

ABTEILUNG DATENBANKEN UND ARTIFICIAL INTELLIGENCE

# Expressiveness of SETAFs and Support-Free ADFs under 3-valued Semantics

**DBAI-TR-2020-117**

**Wolfgang Dvořák** **Atefeh Keshavarzi Zafarghandi**
**Stefan Woltran**

Institut für Logic and Computation
Abteilung Datenbanken und
Artificial Intelligence
Technische Universität Wien
Favoritenstr. 9
A-1040 Vienna, Austria
Tel: +43-1-58801-18403
Fax: +43-1-58801-918403
sek@dbai.tuwien.ac.at
www.dbai.tuwien.ac.at

DBAI TECHNICAL REPORT
2020

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

DBAI TECHNICAL REPORT
DBAI TECHNICAL REPORT DBAI-TR-2020-117, 2020

# Expressiveness of SETAFs and Support-Free ADFs under 3-valued Semantics

**Wolfgang Dvořák** [1] **Atefeh Keshavarzi Zafarghandi** [2]
**Stefan Woltran** [3]

**Abstract.** We investigate the expressiveness of argumentation frameworks with collective attacks and argumentation frameworks with a general attack relation (but without supporting links) under 3-valued semantics. To this end we consider SETAFs and Support-Free ADFs.

[1]Institute of Logic and Computation, TU Wien, Austria. E-mail: dvorak@dbai.tuwien.ac.at
[2]Department of Artificial Intelligence, Bernoulli Institute of Mathematics, Computer Science and Artificial Intelligence, University of Groningen, Groningen, The Netherlands E-mail: A.Keshavarzi.Zafarghandi@rug.nl
[3]Institute of Logic and Computation, TU Wien, Austria. E-mail: woltran@dbai.tuwien.ac.at

# Expressiveness of SETAFs and Support-Free ADFs under 3-valued Semantics

Wolfgang Dvořák, Atefeh Keshavarzi Zafarghandi, Stefan Woltran

**Abstract**

We investigate the expressiveness of argumentation frameworks with collective attacks and argumentation frameworks with a general attack relation (but without supporting links) under 3-valued semantics. To this end we consider SETAFs and Support-Free ADFs.

## 1 Introduction

Abstract argumentation frameworks (AFs) as introduced by Dung [7] are a core formalism in formal argumentation. A popular line of research investigates extensions of Dung AFs that allow for a richer syntax (see, e.g. [3]). In this work we investigate two generalisations of Dung AFs that allow for a more flexible attack structure (but do not consider support between arguments). First SETAFs as introduced by Nielsen and Parsons [13] that extend Dung AFs by allowing for collective attacks such that a set of arguments $B$ attacks another argument $a$ but no proper subset of $B$ attacks $a$. Second, Support-Free Abstract Dialectical Frameworks (SFADFs). Abstract Dialectical Frameworks (ADFs) [4] are studied as a general formalism for modeling and evaluating argumentation, that can cover several generalizations of AFs [3]. In this work we will consider the sub-class of Support-Free ADFs (SFADFs), i.e., ADFs where each link between arguments is attacking. We study the expressiveness of SETAFs and SFADFs w.r.t. their 3-valued semantics (a.k.a. labelling-based semantics) in terms of their signatures. A signature of a semantics $\sigma$ contains all sets of 3-valued labellings, interpretations resp., that correspond to the $\sigma$-labellings, $\sigma$-interpretations resp., of some framework.

Argumentation frameworks with collective attacks have received increasing interest in the last years. For instance, semi-stable, stage, ideal, and eager semantics have been adapted to SETAFs in [8, 10]; translations between SETAFs and other abstract argumentation formalisms are studied in [15]; and the expressiveness of SETAFs with two-valued semantics is investigated in [9]. [19] observed that for particular instantiations, SETAFs provide a more convenient target formalism than Dung AFs. ADFs are among the most prominent abstract argumentation formalisms (see e.g. [2]) and recently the support-free fragment of ADFs was identified in a study on the expressiveness of sub-classes of ADFs [6].

In this work we complement the investigations on expressiveness of SETAFs [9] by (a) providing exact characterisations of the 3-valued signatures of SETAFs and (b) by relating it to the expressiveness of ADFs with arbitrary attack relations but no supporting links.

The expressiveness of SETAFs has first been investigated in [12] where different sub-classes of ADFs, i.e. AFs, SETAFs and Bipolar ADFs, are related w.r.t. their signatures of 3-valued semantics. Moreover, they provide an algorithm to decide realizability in one of the formalisms under admissible, preferred, complete, model and stable semantics. However, no explicit characterisations of the signatures are given. Recently Pührer [16] presented explicit characterisations of the signatures of general ADFs (but not for the sub-classes discussed above). In contrast [8] provides explicit characterisations of the two-valued signatures and shows that SETAFs are more expressive than AFs from the perspective of realizability. In both works all arguments are relevant for the signature. On the other hand, in [10] it is shown that when allowing to add extra arguments to an AF which are not relevant for the signature, i.e. the extensions/labellings are projected on common arguments, then SETAFs and AFs are equivalent.

In [17], it is shown that ADFs that only consist of attack links, i.e., support-free ADFs (SFADFs), and where the acceptance condition of each argument is satisfiable, can be equivalently represented as a SETAF. This provides a sufficient condition for rewriting an ADF as SETAF and raises the question whether it is also a necessary condition. We will show that a SFADF has an equivalent SETAF iff all acceptance conditions are satisfiable. Further, given that BADFs are more expressive than SETAFs from [12], raises the question whether SFADFs have the same expressiveness power as SETAFs. We will show that SFADFs are more expressive than SETAFs and characterise the precise difference under admissible, preferred, grounded, complete, stable and two-valued model semantics.
The main contribution of our paper are as follows:

- We embed SETAFs under 3-valued labeling based semantics [10] in the more general framework of ADFs. That is, we show 3-valued labeling based SETAF semantics to be equivalent to the corresponding ADF semantics. By that we show the equivalence of the 3-valued SETAF semantics in [12] and [10].

- We investigate the expressiveness of SETAFs under 3-valued semantics by providing exact characterizations of the signatures.

- We study the relations between SETAFs and support-free ADFs (SFADFs). In particular we give the exact difference in expressiveness between SETAFs and SFADFs.

# 2 Background

In this section we briefly recall the necessary definitions for SETAFs and ADFs.

**Definition 1.** A set argumentation framework (SETAF) is an ordered pair $F = (A, R)$, where $A$ is a finite set of arguments and $R \subseteq (2^A \setminus \{\emptyset\}) \times A$ is the attack relation.

Given a SETAF $(A,R)$, we write $S \mapsto_R b$ if there is a set $S' \subseteq S$ attacking b, i.e. $(S',b) \in R$. We say that in this case also $S$ attacks b. Moreover, we write $S' \mapsto_R S$ if $S' \mapsto_R b$ for some $b \in S$. We drop the subscript in $\mapsto_R$ if the attack relation is clear from the context.

Notions of conflict and defense can be defined for SETAFs in analogy to these notions in the context of AFs. Given a SETAF $F = (A,R)$, a set $S \subseteq A$ is *conflicting* in $F$ if $S \mapsto_R S$; $S \subseteq A$ is *conflict-free* in $F$, if $S$ is not conflicting in $F$, i.e. if $S' \cup \{a\} \not\subseteq S$ for each $(S',a) \in R$. An argument $a \in A$ is *defended* (in $F$) by a set $S \subseteq A$ if for each $B \subseteq A$, such that $B \mapsto_R a$, also $S \mapsto_R B$. A set $T$ of arguments is hence defended (in $F$) by $S$ if each $a \in T$ is defended by $S$ (in $F$).

The semantics of SETAFs can now also be defined similarly to AFs via a characteristic operator. With a slight abuse of notation, we thus define first of all also for a SETAF $F = (A,R)$, $\Gamma_F(S) = \{a \in A \mid a \text{ is defended by } S \text{ in } F\}$; here the notion of "defense" clearly being that defined for SETAFs. For completeness we detail the definitions of all semantics we consider in this work for SETAFs, although the definitions are exactly as those for AFs (modulo the use of the more general notions of attack and the characteristic operator for SETAFs):

**Definition 2.** Let $F = (A,R)$ be a SETAF. A set $S$ which is conflict-free in $F$ is

- *naive* in $F$ iff S is $\subseteq$-maximal among all conflict-free sets;
- *admissible* in $F$ iff $S \subseteq \Gamma_F(S)$;
- *complete* in $F$ iff $S = \Gamma_F(S)$;
- *grounded* in $F$ iff $S$ is the $\subseteq$-least fixed-point of $\Gamma_F$;
- *preferred* in $F$ iff $S$ is $\subseteq$-maximal admissible (resp. complete) in $F$;
- *stable* in $F$ iff for all $a \in A \setminus S$, $S$ attacks $a$.

We next present 3-valued labelling based semantics as in [10].

**Definition 3.** A (3-valued) labelling of a SETAF $F = (A,R)$ is a total function $\lambda : A \mapsto \{\texttt{in}, \texttt{out}, \texttt{undec}\}$. For $x \in \{\texttt{in}, \texttt{out}, \texttt{undec}\}$ we write $\lambda_x$ to denote the sets of arguments $a$ with $\lambda(a) = x$. We sometimes denote labellings $\lambda$ as triples $(\lambda_{\texttt{in}}, \lambda_{\texttt{out}}, \lambda_{\texttt{undec}})$.

**Definition 4.** Let $F = (A,R)$ be a SETAF. A labelling is called conflict-free if (a) for all $(S,a) \in R$ either $\lambda(a) \neq \texttt{in}$ or there is a $b \in S$ with $\lambda(b) \neq \texttt{in}$, and (b) for all $a \in A$ if $\lambda(a) = \texttt{out}$ then there is an attack $(S,a) \in R$ such that $\lambda(b) = \texttt{in}$ for all $b \in S$. A labelling $\lambda$ which is conflict-free in $F$ is

- *naive* iff $\lambda_{\texttt{in}}$ is $\subseteq$-maximal among all conflict-free labellings, i.e. there is no conflict-free $\lambda'$ with $\lambda_{\texttt{in}} \subset \lambda'_{\texttt{in}}$;
- *admissible* in $F$ iff for all $a \in A$ if $\lambda(a) = \texttt{in}$ then for all $(S,a) \in R$ there is a $b \in S$ such that $\lambda(b) = \texttt{out}$;
- *complete* in $F$ iff for all $a \in A$ (a) $\lambda(a) = \texttt{in}$ iff for all $(S,a) \in R$ there is a $b \in S$ such that $\lambda(b) = \texttt{out}$, and (b) $\lambda(a) = \texttt{out}$ iff there is an attack $(S,a) \in R$ such that $\lambda(b) = \texttt{in}$ for all $b \in S$;

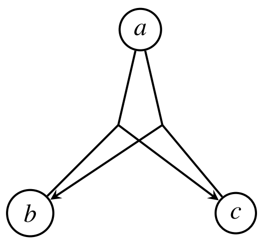



Figure 1: The SETAF of Example 1.

- *grounded* in $F$ iff it is complete and $\lambda_{\texttt{in}}$ is $\subseteq$-minimal among all complete labellings;
- *preferred* in $F$ iff it is complete and $\lambda_{\texttt{in}}$ is $\subseteq$-maximal among all complete labellings, i.e. there is no complete $\lambda'$ with $\lambda_{\texttt{in}} \subset \lambda'_{\texttt{in}}$;
- a *stable* in $F$ iff $\lambda_{\texttt{undec}} = \emptyset$.

**Remark:** Differences to the definitions in [12]: [12] defines admissible, complete, preferred, and mod as 3-valued interpretations (not labellings), i.e. grounded, and stable are missing. preferred is defined via admissible. They already provide equivalence results for their semantics and the corresponding ADF semantics.

The basic definitions of ADFs and semantics of ADFs are derived from those given in [4, 1]. In the following we provide an example of a SETAF to illustrate the concept of labellings and semantics for SETAFs.

**Example 1.** The SETAF $F = (\{a,b,c\},\{(\{a,b\},c),(\{a,c\},b)\})$ is depicted in Figure 1. In $F$, $(\{a,b\},c) \in R$ says that there is a joint attack from $a$ and $b$ to $c$, and $(\{a,c\},b) \in R$ says that there is a joint attack from $a$ and $c$ to $b$. The former attack represents that neither $a$ nor $b$ are strong enough to attack $c$ by themselves. The latter attack indicates that neither $a$ nor $c$ are strong enough to attack $b$ by themselves. The conflict-free labellings of $F$ are $cf(F) = \{\{a \mapsto \texttt{undec}, b \mapsto \texttt{undec}, c \mapsto \texttt{undec}\}, \{a \mapsto \texttt{in}, b \mapsto \texttt{undec}, c \mapsto \texttt{undec}\}, \{a \mapsto \texttt{undec}, b \mapsto \texttt{in}, c \mapsto \texttt{undec}\}, \{a \mapsto \texttt{undec}, b \mapsto \texttt{undec}, c \mapsto \texttt{in}\}, \{a \mapsto \texttt{in}, b \mapsto \texttt{undec}, c \mapsto \texttt{in}\}, \{a \mapsto \texttt{in}, b \mapsto \texttt{out}, c \mapsto \texttt{in}\}, \{a \mapsto \texttt{in}, b \mapsto \texttt{in}, c \mapsto \texttt{undec}\}, \{a \mapsto \texttt{in}, b \mapsto \texttt{in}, c \mapsto \texttt{out}\}, \{a \mapsto \texttt{undec}, b \mapsto \texttt{in}, c \mapsto \texttt{in}\}\}$, the admissible labellings $adm(F) = \{\{a \mapsto \texttt{undec}, b \mapsto \texttt{undec}, c \mapsto \texttt{undec}\}, \{a \mapsto \texttt{in}, b \mapsto \texttt{undec}, c \mapsto \texttt{undec}\}, \{a \mapsto \texttt{in}, b \mapsto \texttt{out}, c \mapsto \texttt{in}\}, \{a \mapsto \texttt{in}, b \mapsto \texttt{in}, c \mapsto \texttt{out}\}\}$, the complete labellings $com(F) = \{\{a \mapsto \texttt{in}, b \mapsto \texttt{undec}, c \mapsto \texttt{undec}\}, \{a \mapsto \texttt{in}, b \mapsto \texttt{out}, c \mapsto \texttt{in}\}, \{a \mapsto \texttt{in}, b \mapsto \texttt{in}, c \mapsto \texttt{out}\}\}$, the unique grounded labelling $grd(F) = \{\{a \mapsto \texttt{in}, b \mapsto \texttt{undec}, c \mapsto \texttt{undec}\}\}$, and the preferred labellings $prf(F) = stb(F) = \{\{a \mapsto \texttt{in}, b \mapsto \texttt{out}, c \mapsto \texttt{in}\}, \{a \mapsto \texttt{in}, b \mapsto \texttt{in}, c \mapsto \texttt{out}\}\}$. Note that, for instance, $\{a \mapsto \texttt{in}, b \mapsto \texttt{undec}, c \mapsto \texttt{in}\}$ is a conflict-free labelling. However, it is not an admissible labelling, since $c$ is mapped to `in` but neither $a$ nor $b$ is mapped to `out`. Further, $\{a \mapsto \texttt{in}, b \mapsto \texttt{undec}, c \mapsto \texttt{undec}\}$ is an admissible and a complete extension,which is not a preferred labelling because $\lambda_{\texttt{in}} = \{a\}$ is not a $\subseteq$-maximal among all complete labellings.

Notice that *Dungs Abstract Argumentation Frameworks (AFs)* [7] and their semantics can be identified with SETAFs whose attacks are restricted a single argument attacking an argument [13]. That is all attacks are of the form $(b,a)$ for some arguments $a,b$ (in the setting of Dung AFs such attacks are then denoted by pairs of arguments $(b,a)$).

We next turn to abstract dialectical frameworks.

**Definition 5.** An abstract dialectical framework (ADF) is a tuple $D = (S,L,C)$ where:

- $S$ is a finite set of arguments (statements, positions);
- $L \subseteq S \times S$ is a set of links among arguments;
- $C = \{\varphi_s\}_{s \in S}$ is a collection of propositional formulas over arguments, called acceptance conditions.

An ADF can be represented by a graph in which nodes indicate arguments and links show the relation among arguments. Each argument $s$ in an ADF is attached by a propositional formula, called acceptance condition, $\varphi_s$ over $par(s)$ such that, $par(s) = \{b \mid (b,s) \in R\}$. The acceptance condition of each argument clarifies under which condition the argument can be accepted. Further, the acceptance conditions indicate the type of links. An *interpretation* $v$ (for $F$) is a function $v : S \mapsto \{\mathbf{t},\mathbf{f},\mathbf{u}\}$, that maps arguments to one of the three truth values true ($\mathbf{t}$), false ($\mathbf{f}$), or undecided ($\mathbf{u}$). Truth values can be ordered via information ordering relation $<_i$ given by $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$ and no other pair of truth values are related by $<_i$. Relation $\leq_i$ is the reflexive and transitive closure of $<_i$. Let $\mathscr{V}$ be the set of all interpretations for an ADF $D$. Then, we call a subset of all interpretations of the ADF, $\mathbb{V} \subseteq 2^{\mathscr{V}}$, an *interpretation-set*. Interpretations can be ordered via $\leq_i$ with respect to their information content. It is said that an interpretation $v$ is an *extension* of another interpretation $w$, if $w(s) \leq_i v(s)$ for each $s \in S$, denoted by $w \leq_i v$. Interpretations $v$ and $w$ are incomparable if neither $w \nleq_i v$ nor $v \nleq_i w$, denoted by $w \not\sim v$.

Semantics for ADFs can also defined via a *characteristic operator* $\Gamma_D$ for an ADF $D$, although given that the semantics of ADFs give interpretations rather than sets of arguments. Given an interpretation $v$ (for $D$), the characteristic operator $\Gamma_D$ for $D$ is defined as

$$\Gamma_D(v) = v' \text{ such that } v'(s) = \begin{cases} \mathbf{t} & \text{if } \varphi_s^v \text{ is irrefutable (i.e., a tautology) ,} \\ \mathbf{f} & \text{if } \varphi_s^v \text{ is unsatisfiable,} \\ \mathbf{u} & \text{otherwise,} \end{cases}$$

where the partial valuation of $\varphi_s$ by $v$, is given by $\varphi_s^v = v(\varphi_s) = \varphi_s[p/\top : v(p) = \mathbf{t}][p/\bot : v(p) = \mathbf{f}]$. Here $p \in par(s)$.

The semantics for ADFs, as defined via the characteristic operator, are provided next in Definition 6.

**Definition 6.** Given an ADF $D = (S,L,C)$, an interpretation $v$ is

- *conflict-free* in $D$ iff $v(s) = \mathbf{t}$ implies $\varphi_s^v$ is satisfiable and $v(s) = \mathbf{f}$ implies $\varphi_s^v$ is unsatisfiable;

- *admissible* in $D$ iff $v \leq_i \Gamma_D(v)$;
- *complete* in $D$ iff $v = \Gamma_D(v)$;
- *grounded* in $D$ iff $v$ is the least fixed-point of $\Gamma_D$;
- *preferred* in $D$ iff $v$ is $\leq_i$-maximal admissible (resp. complete) in $D$;
- a *(two-valued) model* of $D$ iff $v$ is two-valued and for all $s \in S$, it holds that $v(s) = v(\varphi_s)$;
- a *stable model* of $D$ if $v$ is a model of $D$ and $v^{\mathbf{t}} = w^{\mathbf{t}}$, where $w$ is the grounded interpretation of the *stb*-reduct $D^v = (S^v, L^v, C^v)$, where $S^v = v^{\mathbf{t}}$, $L^v = L \cap (S^v \times S^v)$, and $\varphi_s[p/\bot : v(p) = \mathbf{f}]$ for each $s \in S^v$.

As for AFs and SETAFs, the set of all $\sigma$ interpretations for $D$ is denoted by $\sigma(D)$, where $\sigma \in \{cf, adm, com, grd, prf, mod, stb\}$ abbreviates the different semantics in the obvious manner.

Intuitively, the idea of defining stable models of ADFs follows the idea of stable models of logic programming, that breaks self-justify support cycles. In fact, in ADF $D$, a model $v$ is a stable model if there exists a constructive proof for all arguments assigned to true in $v$, if all arguments which are assigned to false in $v$ are actually false. Since in AFs and SETAFs there is no direct support link, stable models and models are equal. The relation among semantics of ADF $D$ are as follows: $stb(D) \subseteq mod(D) \subseteq prf(D) \subseteq com(D) \subseteq adm(D) \subseteq cf(D)$, further, $grd(D) \subseteq com(D)$. The same as AF each ADF contains at least one admissible, preferred, complete, and grounded interpretation, however the existence of stable models, and models respectively, is not guaranteed.

In ADFs links between arguments can be classified into four types, reflecting the relationship of attack and/or support that exists among the arguments. In Definition 7 we consider two-valued interpretations that are only defined over the parents of $a$, that is, only give values to $par(a)$.

**Definition 7.** Let $D = (S, L, C)$ be an ADF. A link $(b, a) \in L$ is called

- *supporting* (in $D$) if for every two-valued interpretation $v$ of $par(a)$, $v(\varphi_a) = \mathbf{t}$ implies $v|^b_{\mathbf{t}}(\varphi_a) = \mathbf{t}$;
- *attacking* (in $D$) if for every two-valued interpretation $v$ of $par(a)$, $v(\varphi_a) = \mathbf{f}$ implies $v|^b_{\mathbf{t}}(\varphi_a) = \mathbf{f}$;
- *redundant* (in $D$) if it is both attacking and supporting;
- *dependent* (in $D$) if it is neither attacking nor supporting.

The classification of the types of the links of ADFs is also relevant for classifying ADFs themselves. Thus, one particularly important subclass of ADFs is that of *bipolar* ADFs or BADFs for short, first defined in [4]. In such an ADF each link is either attacking or supporting (or both; thus, the links can also be redundant). The following example clarifies the role of the acceptance conditions in ADFs and illustrates the functioning of the different semantics in the context of ADFs.

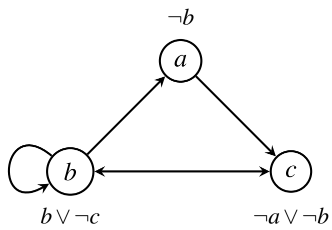


Figure 2: The ADF of Example 2.

**Example 2.** An example of an ADF $D = (S, L, C)$ is shown in Figure 2. To each argument a propositional formula is associated, the acceptance condition of the argument. In the acceptance condition of an argument only the parents of that argument occur as variables. The acceptance condition indicates how the acceptance-status of an argument depends on that of its parents in the ADF. For instance, the acceptance condition of $c$, namely $\varphi_c : \neg a \vee \neg b$, states that $c$ can be accepted in an interpretation where either $a$ or $b$ (or both) are rejected. The acceptance condition $\varphi_a : \neg b$ states that $a$ is acceptable if $b$ is rejected. The acceptance condition $\varphi_b : b \vee \neg c$ indicates that $b$ can be accepted if and only if $b$ is accepted or $c$ is rejected, thus also indicating a form of self-support for $b$.

Since in ADFs an argument appears in the acceptance condition of an argument $a$ if and only if it belongs to the set $par(a)$, the set of links $L$ of an ADF is given implicitly via the acceptance conditions. Thus, there is no need of presenting the links explicitly in the structure of ADFs. For instance, since $b$ appears in $\varphi_b$, $(b, b) \in L$. Moreover, given that for every two-valued interpretation $v$, $v(\varphi_b) = \mathbf{t}$ implies $v|^b_{\mathbf{t}}(\varphi_b) = \mathbf{t}$, the link $(b, b)$ is supporting (in $D$). Further, $D$ is a bipolar ADF in which $L^+ = \{(b, b)\}$ is the set of all supporting links, and $L^- = \{(a, c), (b, c), (c, b), (b, a)\}$ is the set of all attacking links.

In $D$ the interpretation $v = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}, c \mapsto \mathbf{t}\}$ is conflict-free. However, $v$ is not an admissible interpretation, because $\Gamma_D(v) = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}\}$, that is, $v \not\leq_i \Gamma_D(v)$. The interpretation $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{u}\}$ on the other hand is an admissible interpretation. Since $\Gamma_D(v_1) = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}\}$ and $v_1 \leq_i \Gamma_D(v_1)$. Further, $prf(D) = mod(D) = \{\{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{t}\}, \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}\}\}$, but only the first interpretation in this set is a stable model. This is because for $v = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{t}\}$ the unique grounded interpretation $w$ of $D^v$ is $\{a \mapsto \mathbf{t}, c \mapsto \mathbf{t}\}$ and $v^{\mathbf{t}} = w^{\mathbf{t}}$. The interpretation $v' = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}\}$ is not a stable model, since the unique grounded interpretation $w'$ of $D^{v'}$ is $\{b \mapsto \mathbf{u}, c \mapsto \mathbf{t}\}$ and $v'^{\mathbf{t}} \neq w'^{\mathbf{t}}$. Actually, $v'$ is not a stable model because the truth value of $b$ in $v'$ is since of self-support. Moreover, the unique grounded interpretation of $D$ is $v = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}\}$. In addition, we have that for the ADF $D$, $com(D) = prf(D) \cup grd(D)$.

Another subclass of ADFs, having only attacking links, is defined in [11], called *support free ADFs* in the current work, defined formally in Definition 8.

**Definition 8.** A Bipolar ADF $D = (S, L, C)$ is called support-free ADFs (SFADFs) if it neither has supporting nor redundant links, that is, $L = L^-$.

Next consider stable semantics in the context of SFADFs. The motivation behind the reduct in the definition of stable semantics is to avoid cyclic support among arguments. As we now consider support-free ADFs there is no need for such an reduct and thus the models semantics and the stable semantics coincide.

**Lemma 1.** *Let $D = (S, L, C)$ be an ADF, let $v$ be a model of $D$ and let $s \in S$ be an argument such that all parents of $s$ are attackers. Thus, $\varphi_s^v$ is irrefutable if and only if $\varphi_s[p/\bot : v(p) = \mathbf{f}]$ is irrefutable.*

*Proof.* Assume that $D = (S, L, C)$ is an ADF and $v$ is a model of $D$. Further, assume that $s \in S$ such that $\forall p \in par(s)$, $(p, s)$ is an attacking in $D$. Clearly if $\varphi_s[p/\bot : v(p) = \mathbf{f}]$ is irrefutable then also $\varphi_s^v = \varphi_s[p/\top : v(p) = \mathbf{t}][p/\bot : v(p) = \mathbf{f}]$ is irrefutable. It remains to show that if $\varphi_s^v$ is irrefutable then also $\varphi_s[p/\bot : v(p) = \mathbf{f}]$ is irrefutable. Let $\varphi_s' = \varphi_s[p/\bot : v(p) = \mathbf{f}]$. Towards a contradiction, assume that $\varphi_s^v$ is irrefutable and $\varphi_s'$ is not irrefutable. That is, either $\varphi_s'$ is unsatisfiable or it is undecided. In both cases, $\varphi_s'[p/\top : v(p) = \mathbf{t}]$ is unsatisfiable (as all the links are attacking). Thus, $\varphi_s^v = \varphi_s'[p/\top : v(p) = \mathbf{t}]$ is unsatisfiable as well. This is a contradiction with the assumption that $\varphi_s^v$ is irrefutable. □

**Proposition 2.** *For every SFADF $D$ it holds the $mod(D) = stb(D)$.*

*Proof.* Let $D = (S, L, C)$ be a SFADF. Since $stb(D) \subseteq mod(D)$ for each ADF $D$, it remains to show that each model of $D$ is also a stable model of $D$. Toward a contradiction assume that $mod(D) \not\subseteq stb(D)$. Thus, there exists a model $v$ of $D$ which is not a stable model. Let $D^v$ be a $stb$-reduct of $D$ and let $w$ be the unique grounded interpretation of $D^v$. Since it is assumed that $v$ is not a stable model, $v^{\mathbf{t}} \neq w^{\mathbf{t}}$. That is, there exists $s \in S$ such that $v(s) = \mathbf{t}$ and $w(s) \neq \mathbf{t}$. Thus, $\varphi_s[p/\bot : v(p) = \mathbf{f}]$ is not irrefutable. Since, $D$ is a SFADF, all parents of $s$ are attackers. Hence, By Lemma 1, $\varphi_s^v$ is not irrefutable, that is, $v(s) \neq \mathbf{t}$. This is a contradiction by the assumption that $v(s) = \mathbf{t}$. Thus, the assumption that $D$ consists of a model which is not a stable model is incorrect. □

# 3 Embedding SETAFs in ADFs

As observed by Polberg [14] and Linsbichler et.al [12], the notion of collective attacks can also be represented in ADFs by using the right acceptance conditions. We next introduce the class SETADFs of SFADFs that only have collective attacks (including simple binary attacks) in their link structure.

**Definition 9.** An ADF $D = (S, L, C)$ is called SETAF-like (SETADF) if each of the acceptance conditions in $C$ is given by a formula

$$\bigwedge_{cl \in \mathscr{C}} \bigvee_{a \in cl} \neg a.$$

with $\mathscr{C}$ being a set of non-empty clauses $cl$.

That is, in a SETADF each acceptance condition is either $\top$ (if $\mathscr{C}$ is empty) or a proper CNF formula over negative literals.

We next study the relation between SETAFs and SETADFs. We first give a translation from SETAFs to ADFs.

**Definition 10.** Let $F = (A, R)$ be a SETAF. The ADF associated to $F$ is a tuple $D_F = (S, L, C)$ in which, $S = A$, $L = \{(a, b) \mid (B, b) \in \mathrm{R}, a \in B\}$ and $C = \{\varphi_a\}_{a \in S}$ is the collection of acceptance conditions defined, for each $a \in S$, as

$$\varphi_a = \bigwedge_{(B,a) \in R} \bigvee_{a' \in B} \neg a'.$$

Clearly the ADF associated to a SETAF is a SETADF. Also, as we show next, a SETADF can directly be written as a SETAF.

**Definition 11.** Let $D = (S, L, C)$ be a SETADF. We construct the SETAF $F = (A, R)$ in which, $A = S$, and $R$ is constructed as follows. For each argument $s \in S$ with acceptance formula $\bigwedge_{cl \in \mathscr{C}} \bigvee_{a \in cl} \neg a$ we add the attacks $\{(cl, s) \mid cl \in \mathscr{C}\}$.

Notice that $D$ is the ADF associated to the constructed SETAF $F$.

We next deal with the fact that SETAFs semantics are defined as three-valued labellings while semantics for ADFs are defined as three valued interpretations. In order to compare these semantics we associate the *in* label with $t$ the *out* label with $f$ and the *undec* with $u$.

**Definition 12.** The function $\mathit{Lab2Int}(\cdot)$ maps three-valued labellings to three-valued interpretations such that

- $\mathit{Lab2Int}(\lambda)(s) = t$ iff $\lambda(s) = \mathit{in}$,
- $\mathit{Lab2Int}(\lambda)(s) = f$ iff $\lambda(s) = \mathit{out}$, and
- $\mathit{Lab2Int}(\lambda)(s) = u$ iff $\lambda(s) = \mathit{undec}$.

For a labelling $\lambda$ and an interpretation $I$ we write $\lambda \equiv I$ iff $\mathit{Lab2Int}(\lambda) = I$. For a set $\mathscr{L}$ of labellings and a set $\mathbb{V}$ of interpretations we write $\mathscr{L} \equiv \mathbb{V}$ iff $\{\mathit{Lab2Int}(\lambda) \mid \lambda \in \mathscr{L}\} = \mathbb{V}$.

**Theorem 3.** *For a SETAF $F$ and its associated SETADF $D$ we have $\sigma_{\mathscr{L}}(F) \equiv \sigma(D)$ for $\sigma \in \{cf, adm, com, prf, grd, stb\}$.*

*Proof.* Let $F = (A, R)$ be a SETAF and $D = (S, L, C)$ be its corresponding SETADF. We show that $\{\mathit{Lab2Int}(\lambda) \mid \lambda \in \sigma_{\mathscr{L}}(F)\} = \sigma(D)$. Let $\lambda$ be an arbitrary three-valued labelling and let $v = \mathit{Lab2Int}(\lambda)$. We investigate that $\lambda \in \sigma_{\mathscr{L}}(F)$ if and only if $v \in \sigma(D)$.

- Let $\sigma = adm$. We first assume that $\lambda \in adm_{\mathscr{L}}(F)$ and show that $v \in adm(D)$. Consider $s \in S$ and the acceptance condition $\varphi_s = \bigwedge_{(B,s) \in R} \bigvee_{a \in B} \neg a$. If $v(s) = \mathbf{t}$ we have that $\lambda(s) = \mathtt{in}$ and thus that for all $(B, s) \in R$ there exists $b \in B$ s.t. $\lambda(b) = \mathtt{out}$. The latter holds iff for all $(B, s) \in R$ there exists $b \in B$ s.t. $v(b) = \mathbf{f}$ iff partial evaluation of $\varphi_s$ under $v$ is irrefutable iff $\Gamma_D(v)(s) = \mathbf{t}$. If $v(s) = \mathbf{f}$ we have that $\lambda(s) = \mathtt{out}$ and thus that there exists $(B, s) \in R$ s.t. for all $b \in B$: $\lambda(b) = \mathtt{in}$. The latter holds iff there exists $(B, s) \in R$ s.t. for all $b \in B$: $v(b) = \mathbf{t}$ iff $\varphi_s^v$ is unsatisfiable iff $\Gamma_D(v)(s) = \mathbf{f}$. We thus obtain that $v \leq_i \Gamma_D(v)$ and therefore $v \in adm(D)$.

Now we assume $v \in adm(D)$ and show that $\lambda \in adm_{\mathscr{L}}(F)$. That is for each $s$ with $\lambda(s) = \texttt{in}$ we have $\Gamma_D(v)(s) = \mathbf{t}$ and, as argued above, that for all $(B,s) \in R$ there exists $b \in B$ s.t. $\lambda(b) = out$. Moreover for each $s$ with $\lambda(s) = \texttt{out}$ we have $\Gamma_D(v)(s) = \mathbf{f}$ and, as argued above, that there exists $(B,s) \in R$ s.t. for all $b \in B$: $\lambda(b) = \texttt{in}$. We obtain $\lambda \in adm_{\mathscr{L}}(F)$.

- Let $\sigma \in \{com, prf, grd\}$. Let $\lambda \in com_{\mathscr{L}}(F)$ and let $\varphi_s = \bigwedge_{(B,s)\in R} \bigvee_{a\in B} \neg a$ be the acceptance condition of $s \in S$ in $D$. For complete semantics it is enough to show that $\lambda(s) = \texttt{in}$ iff $\Gamma_D(v)(s) = \mathbf{t}$ and $\lambda(s) = \texttt{out}$ iff $\Gamma_D(v)(s) = \mathbf{f}$.

  - It holds that $\lambda(s) = \texttt{in}$ (i.e. $v(s) = \mathbf{t}$) iff for all $(B,s) \in R$ there exists $b \in B$ s.t. $\lambda(b) = \texttt{out}$ iff for all $(B,s) \in R$ there exists $b \in B$ s.t. $v(b) = \mathbf{f}$ iff partial evaluation of $\varphi_s$ under $v$ is irrefutable iff $\Gamma_D(v)(s) = \mathbf{t}$.
  - On the other hand, $\lambda(s) = \texttt{out}$ (i.e. $v(s) = \mathbf{f}$) iff there exists $(B,s) \in R$ s.t. for all $b \in B$: $\lambda(b) = \texttt{in}$ iff there exists $(B,s) \in R$ s.t. for all $b \in B$: $v(b) = \mathbf{t}$ iff $\varphi_s^v$ is unsatisfiable iff $\Gamma_D(v)(s) = \mathbf{f}$.

  Now as complete semantics coincide it is easy to verify that also the maximal, i.e. the preferred, extensions and the minimal, i.e. the grounded, extension coincide.

- Let $\sigma = stb$. Recall that, by Proposition 2, on SETADFs we have that stable and models semantics coincide. We will show that $\lambda \in stb_{\mathscr{L}}(F)$ iff $v \in mod(D)$. That is we show that For complete semantics it is suffice to show that for each $s \in S$ we have (i) $\lambda(s) = \texttt{in}$ iff $v(\varphi_s) = \mathbf{t}$ and (ii) $\lambda(s) = \texttt{out}$ iff $v(\varphi_s) = \mathbf{f}$. To this end let $\varphi_s = \bigwedge_{(B,s)\in R} \bigvee_{a\in B} \neg a$ be the acceptance condition of $s$.

  - It holds that $\lambda(s) = \texttt{in}$ (i.e. $v(s) = \mathbf{t}$) iff for all $(B,s) \in R$ there exists $b \in B$ s.t. $\lambda(b) = out$ iff for all $(B,s) \in R$ there exists $b \in B$ s.t. $v(b) = \mathbf{f}$ iff $v(\varphi_s) = \mathbf{t}$.
  - On the other hand, $\lambda(s) = out$ (i.e. $v(s) = \mathbf{f}$) iff there exists $(B,s) \in R$ s.t. for all $b \in B$: $\lambda(b) = \texttt{in}$ iff there exists $(B,s) \in R$ s.t. for all $b \in B$: $v(b) = \mathbf{t}$ iff $v(\varphi_s) = \mathbf{f}$.

- Finally let $\sigma = cf$. We first assume that $\lambda \in cf_{\mathscr{L}}(F)$ and show that $v \in cf(D)$. Consider $s \in S$ and the acceptance condition $\varphi_s = \bigwedge_{(B,s)\in R} \bigvee_{a\in B} \neg a$. If $v(s) = \mathbf{t}$ we have that $\lambda(s) = \texttt{in}$ and thus that for all $(B,s) \in R$ there exists $b \in B$ s.t. $\lambda(b) \neq \texttt{in}$. The latter holds iff for all $(B,s) \in R$ there exists $b \in B$ s.t. $v(b) \neq \mathbf{t}$ iff $\varphi_s^v$ is satisfiable. If $v(s) = \mathbf{f}$ we have that $\lambda(s) = \texttt{out}$ and thus that there exists $(B,s) \in R$ s.t. for all $b \in B$: $\lambda(b) = \texttt{in}$. The latter holds iff there exists $(B,s) \in R$ s.t. for all $b \in B$: $v(b) = \mathbf{t}$ iff $\varphi_s^v$ is unsatisfiable. We thus obtain that $v \in cf(D)$.

  Now we assume $v \in cf(D)$ and show that $\lambda \in cf_{\mathscr{L}}(F)$. That is for each $s$ with $\lambda(s) = \texttt{in}$ we have $\varphi_s^v$ is satisfiable and, as argued above, that for all $(B,s) \in R$ there exists $b \in B$ s.t. $\lambda(b) \neq \texttt{in}$. Moreover for each $s$ with $\lambda(s) = \texttt{out}$ we have $\varphi_s^v$ is unsatisfiable and, as argued above, that there exists $(B,s) \in R$ s.t. for all $b \in B$: $\lambda(b) = \texttt{in}$. We obtain $\lambda \in cf_{\mathscr{L}}(F)$.

$\square$

Notice that by the above theorem we also have that the 3-valued SETAF semantics introduced in [12] coincide with the 3-valued labelling based SETAF semantics of [10]. Notice that model semantics of [12] corresponds to the stable semantics of [10].

# 4 3-valued Signatures of SETAFs

By Theorem 3 we can use labellings of SETAFs and interpretations of the SETADF class of ADF interchangeable. In this section on 3-valued Signatures of SETAFs/SETADFs, for convenience, we will use the SETAF terminology.

**Definition 13.** We define the signature of SETAFs under a labelling-based semantics $\sigma_{\mathscr{L}}$ as

$$\Sigma^{\sigma}_{SETAF} = \{\sigma_{\mathscr{L}}(F) \mid F \in SETAF\}.$$

**Proposition 4.** *The signature $\Sigma^{stb_{\mathscr{L}}}_{SETAF}$ is given by all sets $\mathbb{L}$ of labellings such that*

1. *all labellings $\lambda \in \mathbb{L}$ have the same domain and $\lambda(s) \neq \texttt{undec}$ for all $\lambda \in \mathbb{L}$ and arguments s.*
2. *if the domain is non-empty each $\lambda \in \mathbb{L}$ assigns at least one argument to* `in`.
3. *for arbitrary $\lambda_1, \lambda_2 \in \mathbb{L}$ with $\lambda_1 \neq \lambda_2$ there is an argument a such $\lambda_1(a) = \texttt{in}$ and $\lambda_2(a) = \texttt{out}$.*

*Proof.* We first show that for each SETAF $F$ the set $stb_{\mathscr{L}}(F)$ satisfies the conditions of the proposition. First clearly all $\lambda \in stb_{\mathscr{L}}(F)$ have the same domain and by the definition of stable semantics do not assign `undec` to any argument. That is the first condition is satisfied. Now, toward a contradiction assume that $\lambda \in stb_{\mathscr{L}}(F)$ assigns all arguments to `out`. Consider an arbitrary argument $a$. By definition of stable semantics $a$ is only labeled `out` if there is an attack $(B,a)$ such that all arguments in $B$ are labeled in `in`, a contradiction. Thus by the above contradiction we obtain there is at least one argument $a$ with $\lambda(a) = \texttt{in}$. For condition 3, toward a contradiction assume that for all arguments $a$ with $\lambda_1(a) = \texttt{in}$ also $\lambda_2(a) = \texttt{in}$ holds. As $\lambda_1 \neq \lambda_2$ there is an $a$ with $\lambda_2(a) = \texttt{in}$ and $\lambda_1(a) = \texttt{out}$. That is, there is an attack $(B,a)$ such that $\lambda_1(b) = \texttt{in}$ for all $b \in B$. But then also $\lambda_2(b) = \texttt{in}$ for all $b \in B$ and by $\lambda_2(a) = \texttt{in}$ we obtain that $\lambda_2 \notin cf_{\mathscr{L}}(F)$, a contradiction.

Now assume that $\mathbb{L}$ satisfies all the conditions. We give a SETAF $F_{\mathbb{L}} = (A_{\mathbb{L}}, R_{\mathbb{L}})$ with $stb_{\mathscr{L}}(F_{\mathbb{L}}) = \mathbb{L}$.

$$A_{\mathbb{L}} = \texttt{Args}_{\mathbb{L}}$$
$$R_{\mathbb{L}} = \{(\lambda_{\texttt{in}}, a) \mid \lambda \in \mathbb{L}, \lambda(a) = \texttt{out}\}$$

We first show $stb_{\mathscr{L}}(F_{\mathbb{L}}) \supseteq \mathbb{L}$: Consider an arbitrary $\lambda \in \mathbb{L}$: By condition 1 there is no $a \in \texttt{Args}_{\mathbb{L}}$ with $\lambda(a) = \texttt{undec}$ and it only remains to show $\lambda \in cf_{\mathscr{L}}(F_{\mathbb{L}})$. We first consider `out` labeled arguments. First, if $\lambda(a) = \texttt{out}$ for some argument $a$ then by construction and condition (2) we have an attack $(\lambda_{\texttt{in}}, a)$ and thus $a$ is legally labeled `out`. Now toward a contradiction assume there is a conflict $(B,a)$ such that $B \cup \{a\} \subseteq$

$\lambda_{\texttt{in}}$. Then, by construction of $R_{\mathbb{L}}$ there is a $\lambda' \in \mathbb{L}$ with $\lambda'_{\texttt{in}} = B$ and $\lambda_{\texttt{in}} \neq B$ (as $a \in \lambda_{\texttt{in}}$). That is, $\lambda'_{\texttt{in}} \subset \lambda_{\texttt{in}}$, a contradiction to (3). Thus, $\lambda \in cf_{\mathscr{L}}(F_{\mathbb{L}})$ and therefore $\lambda \in stb_{\mathscr{L}}(F_{\mathbb{L}})$.

We complete the proof by showing $stb_{\mathscr{L}}(F_{\mathbb{L}}) \subseteq \mathbb{L}$: Consider $\lambda \in stb_{\mathscr{L}}(F_{\mathbb{L}})$: If $\lambda$ maps all arguments to in then there is no attack in $R_{\mathbb{L}}$ which means that $\mathbb{L}$ contains only the labelling $\lambda$. Thus we can assume that $\lambda(a) = \texttt{out}$ for some argument $a$ and there is $(B,a) \in R_{\mathbb{L}}$ with $\lambda(b) = \texttt{in}$ for all $b \in B$. By construction there is $\lambda' \in \mathbb{L}$ such that $\lambda'_{\texttt{in}} = B$. Then by construction we have $(B,c) \in R_{\mathbb{L}}$ for all $c \notin B$ and thus $\lambda'_{\texttt{in}} = B = \lambda_{\texttt{in}}$ and moreover $\lambda'_{\texttt{out}} = \lambda_{\texttt{out}}$ and thus $\lambda = \lambda'$. □

By Theorem 3 we get the corresponding characterisation of $\Sigma^{stb}_{\text{SETADF}}$.

**Proposition 5.** *The signature $\Sigma^{prf_{\mathscr{L}}}_{SETAF}$ is given by all sets $\mathbb{L}$ of labellings such that*

1. *all labellings $\lambda \in \mathbb{L}$ have the same domain.*
2. *if $\lambda \in \mathbb{L}$ assigns one argument to* out *then it also assigns an argument to* in.
3. *for arbitrary $\lambda_1, \lambda_2 \in \mathbb{L}$ with $\lambda_1 \neq \lambda_2$ there is an argument $a$ such $\lambda_1(a) = \texttt{in}$ and $\lambda_2(a) = \texttt{out}$.*

*Proof.* We first show that for each SETAF $F$ the set $prf_{\mathscr{L}}(F)$ satisfies the conditions of the proposition. The first condition is satisfied as clearly all $\lambda \in prf_{\mathscr{L}}(F)$ have the same domain. Now, assume that $\lambda \in prf_{\mathscr{L}}(F)$ assigns an argument $a$ to out. By the definition of conflict-free labellings there is an attack $(B,a)$ such that all arguments $b \in B$ are labeled in. Thus condition 2 is satisfied. For condition 3, toward a contradiction assume that for all arguments $a$ with $\lambda_1(a) = \texttt{in}$ also $\lambda_2(a) = \texttt{in}$ holds. As for preferred semantics the out labels are fully determined by the in labels and $\lambda_1 \neq \lambda_2$ there is an $a$ with $\lambda_2(a) = \texttt{in}$ and $\lambda_1(a) \neq \texttt{in}$. This is in contradiction to the $\subseteq$-maximality of $\lambda_{\texttt{in}}$.

Now assume that $\mathbb{L}$ satisfies all the conditions. We give a SETAF $F_{\mathbb{L}} = (A_{\mathbb{L}}, R_{\mathbb{L}})$ with $prf_{\mathscr{L}}(F_{\mathbb{L}}) = \mathbb{L}$.

$$A_{\mathbb{L}} = \texttt{Args}_{\mathbb{L}}$$
$$R_{\mathbb{L}} = \{(\lambda_{\texttt{in}}, a) \mid \lambda \in \mathbb{L}, \lambda(a) = \texttt{out}\} \cup \{(\lambda_{\texttt{in}} \cup \{a\}, a) \mid \lambda \in \mathbb{L}, \lambda(a) = \texttt{undec}\}$$

We first show $prf_{\mathscr{L}}(F_{\mathbb{L}}) \supseteq \mathbb{L}$: Consider an arbitrary $\lambda \in \mathbb{L}$: We first show $\lambda \in cf_{\mathscr{L}}(F_{\mathbb{L}})$. We first consider out labeled arguments. First, if $\lambda(a) = \texttt{out}$ for some argument $a$ then by construction and condition (2) we have an attack $(\lambda_{\texttt{in}}, a)$ and thus $a$ is legally labeled out. Now toward a contradiction assume there is a conflict $(B,a)$ such that $B \cup \{a\} \subseteq \lambda_{\texttt{in}}$. If $|\mathbb{L}| = 1$, by the construction of $F_{\mathbb{L}}$ there is no $(B,a) \in R_{\mathbb{L}}$ such that $a \in \lambda_{\texttt{in}}$. That is, $a$ is legally labeled in. If $|\mathbb{L}| > 1$, by construction there is a $\lambda' \in \mathbb{L}$ with $\lambda'_{\texttt{in}} = B \setminus \{a\}$, a contradiction to (3). Thus, $\lambda \in cf_{\mathscr{L}}(F_{\mathbb{L}})$. Next we show that $\lambda \in adm_{\mathscr{L}}(F_{\mathbb{L}})$. Consider an argument $a$ with $\lambda(a) = \texttt{in}$ and an attack $(B,a)$. Then, by construction there is a $\lambda' \in \mathbb{L}$ with $\lambda'_{\texttt{in}} = B \setminus \{a\}$ and, by condition (3), an argument $b \in B$ such that $\lambda(b) = \texttt{out}$. Thus, $\lambda \in adm_{\mathscr{L}}(F_{\mathbb{L}})$. Finally we show that $\lambda \in prf_{\mathscr{L}}(F_{\mathbb{L}})$. Towards a contradiction assume that there is a $\lambda' \in adm_{\mathscr{L}}(F_{\mathbb{L}})$ with $\lambda_{\texttt{in}} \subset \lambda'_{\texttt{in}}$. Let $a$ be an argument such that $\lambda'(a) = \texttt{in}$ and $\lambda(a) \in \{\texttt{out}, \texttt{undec}\}$. By construction there is either an attack $(\lambda_{\mathtt{in}}, a)$ or an attack $(\lambda_{\mathtt{in}} \cup \{a\}, a)$. In both cases $\lambda' \notin adm_{\mathscr{L}}(F_{\mathbb{L}})$ a contradiction. Hence, $\lambda \in prf_{\mathscr{L}}(F_{\mathbb{L}})$.

We complete the proof by showing $prf_{\mathscr{L}}(F_{\mathbb{L}}) \subseteq \mathbb{L}$: Consider $\lambda \in prf_{\mathscr{L}}(F_{\mathbb{L}})$: If $\lambda$ maps all arguments to in then there is no attack in $R_{\mathbb{L}}$ which means that $\mathbb{L}$ contains only the labelling $\lambda$. Thus we can assume that $\lambda(a) = \mathtt{out}$ for some argument $a$ and there is $(B, a) \in R_{\mathbb{L}}$ with $\lambda(b) = \mathtt{in}$ for all $b \in B$. By construction there is $\lambda' \in \mathbb{L}$ such that $\lambda'_{\mathtt{in}} = B$. Then by construction we have $(B, c) \in R_{\mathbb{L}}$ for all $c$ with $\lambda'(c) = \mathtt{out}$ and $(B \cup \{c\}, c) \in R_{\mathbb{L}}$ for all $c$ with $\lambda'(c) = \mathtt{undec}$. We obtain that $\lambda'_{\mathtt{in}} = B = \lambda_{\mathtt{in}}$ and thus $\lambda = \lambda'$. □

**Proposition 6.** *The signature $\Sigma^{cf_{\mathscr{L}}}_{SETAF}$ is given by all sets $\mathbb{L}$ of labellings such that*

1. *all $\lambda \in \mathbb{L}$ have the same domain $\mathtt{Args}_{\mathbb{L}}$.*
2. *if $\lambda \in \mathbb{L}$ assigns one argument to* out *then it also assigns an argument to* in*.*
3. *for $\lambda \in \mathbb{L}$ and $C \subseteq \lambda_{\mathtt{in}}$ also $(C, \emptyset, \mathtt{Args}_{\mathbb{L}} \setminus C) \in \mathbb{L}$*
4. *for $\lambda \in \mathbb{L}$ and $C \subseteq \lambda_{\mathtt{out}}$ also $(\lambda_{\mathtt{in}}, \lambda_{\mathtt{out}} \setminus C, \lambda_{\mathtt{undec}} \cup C) \in \mathbb{L}$*
5. *for $\lambda, \lambda' \in \mathbb{L}$ with $\lambda_{\mathtt{in}} \subseteq \lambda'_{\mathtt{in}}$ also $(\lambda'_{\mathtt{in}}, \lambda_{\mathtt{out}} \cup \lambda'_{\mathtt{out}}, \lambda_{\mathtt{undec}} \cap \lambda'_{\mathtt{undec}}) \in \mathbb{L}$.*
6. *for $\lambda, \lambda' \in \mathbb{L}$ and $C \subseteq \lambda_{\mathtt{out}}$ (s.t. $C \neq \emptyset$) we have $\lambda_{\mathtt{in}} \cup C \not\subseteq \lambda'_{\mathtt{in}}$*

*Proof.* We first show that for each SETAF $F$ the set $cf_{\mathscr{L}}(F)$ satisfies the conditions of the proposition. The first condition is satisfied as clearly all $\lambda \in cf_{\mathscr{L}}(F)$ have the same domain. Now, assume that $\lambda \in cf_{\mathscr{L}}(F)$ assigns an argument $a$ to out. By the definition of conflict-free labellings there is an attack $(B, a)$ such that all arguments $b \in B$ are labeled in. Thus condition 2 is satisfied. For condition 3, toward a contradiction assume that $(C, \emptyset, \mathtt{Args}_{\mathbb{L}} \setminus C)$ is not conflict-free. Then there is an attack $(B, a)$ such that $B \cup \{a\} \subseteq C$. But then also $B \cup \{a\} \subseteq \lambda_{\mathtt{in}}$ and thus $\lambda \notin cf_{\mathscr{L}}(F)$, a contradiction. Condition 4, is satisfied as in the definition of conflict-free labellings there are no conditions for label an argument undec. Further, the conditions that allow to label an argument out solely depend on the in labeled arguments. Since $\lambda_{\mathtt{out}} \setminus C \subseteq \lambda_{\mathtt{out}}$, the condition for arguments labeled out is satisfied. For condition 5 consider $\lambda, \lambda' \in cf_{\mathscr{L}}(F)$ with $\lambda_{\mathtt{in}} \subseteq \lambda'_{\mathtt{in}}$ and $\lambda^* = (\lambda'_{\mathtt{in}}, \lambda_{\mathtt{out}} \cup \lambda'_{\mathtt{out}}, \lambda_{\mathtt{undec}} \cap \lambda'_{\mathtt{undec}})$. First there cannot be an attack $(B, a)$ such that $B \cup \{a\} \subseteq \lambda^*_{\mathtt{in}}$ as $\lambda' \in cf_{\mathscr{L}}(F)$. Hence, $\lambda'_{\mathtt{in}} \cap \lambda_{\mathtt{out}} = \emptyset$ and thus $\lambda^*$ is a well-defined labelling. Moreover, for each $a$ with $\lambda^*(a) = \mathtt{out}$ there is an attack $(B, a)$ with $B \subseteq \lambda^*_{\mathtt{in}}$ as either $\lambda(a) = \mathtt{out}$ or $\lambda'(a) = \mathtt{out}$. Thus, $\lambda^* \in cf_{\mathscr{L}}(F)$ and therefore condition 5 holds. For condition 6 consider $\lambda, \lambda' \in cf_{\mathscr{L}}(F)$ and a set $C \subseteq \lambda_{\mathtt{out}}$ containing an argument $a$ such that $\lambda(a) = \mathtt{out}$. That is, there is an attack $(B, a)$ with $B \subseteq \lambda_{\mathtt{in}}$ and thus $\lambda_{\mathtt{in}} \cup C \not\subseteq \lambda'$. That is, condition 6 is satisfied.

Now assume that $\mathbb{L}$ satisfies all the conditions. We give a SETAF $F_{\mathbb{L}} = (A_{\mathbb{L}}, R_{\mathbb{L}})$ with $cf_{\mathscr{L}}(F_{\mathbb{L}}) = \mathbb{L}$.

$$A_{\mathbb{L}} = \mathtt{Args}_{\mathbb{L}}$$
$$R_{\mathbb{L}} = \{(\lambda_{\mathtt{in}}, a) \mid \lambda \in \mathbb{L}, \lambda(a) = \mathtt{out}\} \cup \{(B, b) \mid b \in B, \nexists \lambda \in \mathbb{L} : \lambda_{\mathtt{in}} = B\}$$

We first show $cf_{\mathscr{L}}(F_{\mathbb{L}}) \supseteq \mathbb{L}$: Consider an arbitrary $\lambda \in \mathbb{L}$: First, if $\lambda(a) = \texttt{out}$ for some argument $a$ then by construction and condition (2) we have an attack $(\lambda_{\texttt{in}}, a)$ and thus $a$ is legally labeled out. Now toward a contradiction assume there is a conflict $(B,a)$ such that $B \cup \{a\} \subseteq \lambda_{\texttt{in}}$. By condition (3) it cannot be the case that $a \in B$. Thus, by construction there is a $\lambda' \in \mathbb{L}$ with $\lambda'_{\texttt{in}} = B$, a contradiction to condition (6). Thus, $\lambda \in cf_{\mathscr{L}}(F_{\mathbb{L}})$.

We complete the proof by showing $cf_{\mathscr{L}}(F_{\mathbb{L}}) \subseteq \mathbb{L}$: Consider $\lambda \in cf_{\mathscr{L}}(F_{\mathbb{L}})$: If $\lambda$ maps all arguments to in then there is no attack in $R_{\mathbb{L}}$ which means that $\mathbb{L}$ contains only the labelling $\lambda$. Thus we can assume that $\lambda(a) \in \{\texttt{out}, \texttt{undec}\}$ for some argument $a$. If $\lambda_{\texttt{in}} \neq \lambda'_{\texttt{in}}$ for all $\lambda' \in \mathbb{L}$ then by construction of the second part of $R_{\mathbb{L}}$ there would be attacks $(\lambda_{\texttt{in}}, b)$ for all $b \in \lambda_{\texttt{in}}$, which is in contradiction to $\lambda \in cf_{\mathscr{L}}(F_{\mathbb{L}})$. Thus, there is $\lambda' \in \mathbb{L}$ such that $\lambda'_{\texttt{in}} = \lambda_{\texttt{in}}$. For arguments $a$ with $\lambda(a) = \texttt{out}$ there is an attack $(B,a)$ with $B \subseteq \lambda_{\texttt{in}}$ and, by construction, a $\lambda^* \in \mathbb{L}$ such that $\lambda^*_{\texttt{in}} = B$ and $\lambda^*(a) = \texttt{out}$. By the existence of $\lambda' \in \mathbb{L}$ and condition (5) we have that there exists $\lambda'' \in \mathbb{L}$ such that $\lambda_{\texttt{in}} = \lambda''_{\texttt{in}}$, $\lambda'_{\texttt{out}} \subseteq \lambda''_{\texttt{out}}$ and $a \in \lambda''_{\texttt{out}}$. By iteratively applying this argument for each argument a with $\lambda(a) = \texttt{out}$ we obtain that there is a labelling $\hat{\lambda} \in \mathbb{L}$ such that $\lambda_{\texttt{in}} = \hat{\lambda}_{\texttt{in}}$ and $\lambda_{\texttt{out}} \subset \hat{\lambda}_{\texttt{out}}$. By condition (4) we obtain that $\lambda \in \mathbb{L}$. □

**Remark:** For extension-based semantics we have that cf sets fully determine naive extensions and vice versa. For labelling-based semantics only the former is true.

**Proposition 7.** *The signature $\Sigma^{nai_{\mathscr{L}}}_{SETAF}$ is given by all sets $\mathbb{L}$ of labellings such that*

1. *all $\lambda \in \mathbb{L}$ have the same domain $\texttt{Args}_{\mathbb{L}}$.*
2. *if $\lambda \in \mathbb{L}$ assigns one argument to* out *then it also assigns an argument to* in*.*
3. *for $\lambda \in \mathbb{L}$ and $C \subseteq \lambda_{\texttt{out}}$ also $(\lambda_{\texttt{in}}, \lambda_{\texttt{out}} \setminus C, \lambda_{\texttt{undec}} \cup C) \in \mathbb{L}$*
4. *for $\lambda, \lambda' \in \mathbb{L}$ with $\lambda_{\texttt{in}} = \lambda'_{\texttt{in}}$ also $(\lambda_{\texttt{in}}, \lambda_{\texttt{out}} \cup \lambda'_{\texttt{out}}, \lambda_{\texttt{undec}} \cap \lambda'_{\texttt{undec}}) \in \mathbb{L}$.*
5. *for arbitrary $\lambda, \lambda' \in \mathbb{L}$ we have $\lambda_{\texttt{in}} \not\subset \lambda'_{\texttt{in}}$.*

*Proof.* First we show that for each SETAF $F$ the set $nai_{\mathscr{L}}(F)$ satisfies the conditions of the proposition. Since each naive labelling (in $F$) is a conflict-free labelling (in $F$), the first two conditions are satisfied by Proposition 6. For condition 3, notice that the definition of naive labellings does not require any arguments to be labeled out. Thus, whenever there is a naive labelling $\lambda$ that labels some arguments out there is also another naive labelling $\lambda'$that labels these arguments undec and coincide with $\lambda$ on the other arguments. Given two naive labelling $\lambda, \lambda'$ with $\lambda_{\texttt{in}} = \lambda'_{\texttt{in}}$ we know that for each $a \in \lambda_{\texttt{out}} \cup \lambda'_{\texttt{out}}$ there is an attack $(B,a)$ with $B \subseteq \lambda_{\texttt{in}}$. Thus also $(\lambda_{\texttt{in}}, \lambda_{\texttt{out}} \cup \lambda'_{\texttt{out}}, \lambda_{\texttt{undec}} \cap \lambda'_{\texttt{undec}}) \in nai_{\mathscr{L}}(F)$ and condition 4 is satisfied. Finally condition 5 is by the maximality of $\lambda_{\texttt{in}}$ in naive labelings.

Now assume that $\mathbb{L}$ satisfies all the conditions. We give a SETAF $F_{\mathbb{L}} = (A_{\mathbb{L}}, R_{\mathbb{L}})$ with $cf_{\mathscr{L}}(F_{\mathbb{L}}) = \mathbb{L}$.

$$A_{\mathbb{L}} = \texttt{Args}_{\mathbb{L}}$$
$$R_{\mathbb{L}} = \{(\lambda_{\texttt{in}}, a) \mid \lambda \in \mathbb{L}, \lambda(a) = \texttt{out}\} \cup \{(\lambda_{\texttt{in}} \cup \{a\}, a) \mid \lambda \in \mathbb{L}, \lambda(a) = \texttt{undec}\}$$

We first show $nai_{\mathscr{L}}(F_{\mathbb{L}}) \supseteq \mathbb{L}$: Consider an arbitrary $\lambda \in \mathbb{L}$: We first show $\lambda \in cf_{\mathscr{L}}(F_{\mathbb{L}})$. First, if $\lambda(a) = \mathtt{out}$ for some argument $a$ then by construction and condition (2) we have an attack $(\lambda_{\mathtt{in}}, a)$ and thus $a$ is legally labeled $\mathtt{out}$. Now toward a contradiction assume there is a conflict $(B,a)$ such that $B \cup \{a\} \subseteq \lambda_{\mathtt{in}}$. If $|\mathbb{L}| > 1$, then, by construction there is a $\lambda' \in \mathbb{L}$ with $\lambda'_{\mathtt{in}} = B \setminus \{a\}$, a contradiction to (5). Thus, $\lambda \in cf_{\mathscr{L}}(F_{\mathbb{L}})$. Finally we show that $\lambda \in nai_{\mathscr{L}}(F_{\mathbb{L}})$. Towards a contradiction assume that there is a $\lambda' \in cf_{\mathscr{L}}(F_{\mathbb{L}})$ with $\lambda_{\mathtt{in}} \subset \lambda'_{\mathtt{in}}$. Let $a$ be an argument such that $\lambda'(a) = \mathtt{in}$ and $\lambda(a) \in \{\mathtt{out}, \mathtt{undec}\}$. By construction there is either an attack $(\lambda_{\mathtt{in}}, a)$ or an attack $(\lambda_{\mathtt{in}} \cup \{a\}, a)$. In both cases $\lambda' \notin cf_{\mathscr{L}}(F_{\mathbb{L}})$ a contradiction. Hence, $\lambda \in nai_{\mathscr{L}}(F_{\mathbb{L}})$.

We complete the proof by showing $nai_{\mathscr{L}}(F_{\mathbb{L}}) \subseteq \mathbb{L}$: Consider $\lambda \in nai_{\mathscr{L}}(F_{\mathbb{L}})$: If $\lambda$ maps all arguments to $\mathtt{in}$ then there is no attack in $R_{\mathbb{L}}$ which means that $\mathbb{L}$ contains only the labelling $\lambda$. Thus we can assume that $\lambda(a) \in \{\mathtt{out}, \mathtt{undec}\}$ for some argument $a$ and there is $(B,a) \in R_{\mathbb{L}}$ with $B \subseteq \lambda_{\mathtt{in}} \cup \{a\}$. By construction there is $\lambda' \in \mathbb{L}$ such that $\lambda'_{\mathtt{in}} = B \setminus \{a\}$. By the above $\lambda' \in nai_{\mathscr{L}}(F_{\mathbb{L}})$ and thus $\lambda = \lambda'_{\mathtt{in}}$ (cf. condition 5). Moreover, for each argument $b$ with $\lambda(b) = \mathtt{out}$, by construction, we have a $\lambda^b \in \mathbb{L}$ with $\lambda^b_{\mathtt{in}} = \lambda_{\mathtt{in}}$ and $\lambda^b(b) = \mathtt{out}$. Let us next define the labelling

$$\lambda^* = (\lambda'_{\mathtt{in}}, \lambda'_{\mathtt{out}} \cup \bigcup_{b \in \lambda_{\mathtt{out}}} \lambda^b_{\mathtt{out}}, \lambda'_{\mathtt{undec}} \cap \bigcap_{b \in \lambda_{\mathtt{out}}} \lambda^b_{\mathtt{undec}}).$$

By condition 4 we have that $\lambda^* \in \mathbb{L}$. By the construction of $\lambda^*$ we have $\lambda_{\mathtt{out}} \subseteq \lambda^*_{\mathtt{out}}$ and $\lambda_{\mathtt{in}} = \lambda^*_{\mathtt{in}}$. Thus, by condition 3, $\lambda \in \mathbb{L}$. □

**Proposition 8.** *For each $\mathbb{L} \in \Sigma_{SETAF}^{adm_{\mathscr{L}}}$ we have:*

1. *all $\lambda \in \mathbb{L}$ have the same domain.*
2. *if $\lambda \in \mathbb{L}$ assigns one argument to* $\mathtt{out}$ *then it also assigns an argument to* $\mathtt{in}$.
3. *for $\lambda, \lambda' \in \mathbb{L}$ and $C \subseteq \lambda_{\mathtt{out}}$ (s.t. $C \neq \emptyset$) we have $\lambda_{\mathtt{in}} \cup C \not\subseteq \lambda'_{\mathtt{in}}$*
4. *for arbitrary $\lambda, \lambda' \in \mathbb{L}$ either (a) $(\lambda_{\mathtt{in}} \cup \lambda'_{\mathtt{in}}, \lambda_{\mathtt{out}} \cup \lambda'_{\mathtt{out}}, \lambda_{\mathtt{undec}} \cap \lambda'_{\mathtt{undec}}) \in \mathbb{L}$ or (b) there is an argument a such $\lambda(a) = \mathtt{in}$ and $\lambda'(a) = \mathtt{out}$.*
5. *for $\lambda, \lambda' \in \mathbb{L}$ with $\lambda_{\mathtt{out}} \subseteq \lambda'_{\mathtt{out}}$, and $C \subseteq \lambda_{\mathtt{in}} \setminus \bigcup_{\lambda^* \in \mathbb{L}:\, \lambda^*_{\mathtt{in}} = \lambda'_{\mathtt{in}}} \lambda^*_{\mathtt{out}}$ we have $(\lambda'_{\mathtt{in}} \cup C, \lambda'_{\mathtt{out}}, \lambda'_{\mathtt{undec}} \setminus C) \in \mathbb{L}$.*
6. *for $\lambda, \lambda' \in \mathbb{L}$ with $\lambda_{\mathtt{in}} \subseteq \lambda'_{\mathtt{in}}$, and $C \subseteq \lambda_{\mathtt{out}}$ we have $(\lambda'_{\mathtt{in}}, \lambda'_{\mathtt{out}} \cup C, \lambda'_{\mathtt{undec}} \setminus C) \in \mathbb{L}$.*
7. *for $\lambda, \lambda' \in \mathbb{L}$ with $\lambda_{\mathtt{in}} \subseteq \lambda'_{\mathtt{in}}$ and $\lambda_{\mathtt{out}} \supseteq \lambda'_{\mathtt{out}}$ we have $(\lambda_{\mathtt{in}}, \lambda'_{\mathtt{out}}, \mathit{Args}_{\mathbb{L}} \setminus (\lambda_{\mathtt{in}} \cup \lambda'_{\mathtt{out}})) \in \mathbb{L}$.*
8. *$(\emptyset, \emptyset, \mathit{Args}_{\mathbb{L}}) \in \mathbb{L}$*

*Proof.* We show that for each SETAF $F$ the set $adm_{\mathscr{L}}(F)$ satisfies the conditions of the proposition. Conditions (1)-(3) are by the fact that $adm_{\mathscr{L}}(F) \subseteq cf_{\mathscr{L}}(F)$. For condition 4, let $\lambda, \lambda'$ be admissible labellings such that $\lambda_{\mathtt{in}} \cap \lambda'_{out} = \{\}$ (since each admissible labelling defends itself, $\lambda'_{\mathtt{in}} \cap \lambda_{out} = \{\}$). Thus, $\lambda^* = (\lambda_{\mathtt{in}} \cup \lambda'_{\mathtt{in}}, \lambda_{\mathtt{out}} \cup \lambda'_{\mathtt{out}}, \lambda_{\mathtt{undec}} \cap \lambda'_{\mathtt{undec}})$ is a well-defined labelling. Consider that $\lambda^*(a) = \mathtt{in}$, that is, either $\lambda(a) = \mathtt{in}$

or $\lambda'(a) = \mathtt{in}$. Since $\lambda, \lambda'$ are admissible labellings, for each conflict $(B,a)$ there exists $b \in B$ s.t. $\lambda(b) = \mathtt{out}$ in the former case and $\lambda'(b) = \mathtt{out}$ in the latter case. Thus, for each conflict $(B,a)$ there exists $b \in B$ s.t. $\lambda^*(b) = \mathtt{out}$. Moreover, if $\lambda^*(a) = \mathtt{out}$ there is an attack $(B,a)$ with $B \subseteq \lambda_{\mathtt{in}}$ or $B \subseteq \lambda'_{\mathtt{in}}$, that is, there exists a conflict $(B,a)$ such that $B \subseteq \lambda^*_{\mathtt{in}}$. On the other hand, assume that $\lambda_{\mathtt{in}} \cap \lambda'_{out} \neq \{\}$, for instance, $a \in \lambda_{\mathtt{in}} \cap \lambda'_{out}$. Therefore, $a \in \lambda^*_{\mathtt{in}}$ and $a \in \lambda^*_{\mathtt{out}}$. That is, $\lambda^*$ is not a well-defined labelling.

For condition 5, let $\lambda^* = (\lambda'_{\mathtt{in}} \cup C, \lambda'_{\mathtt{out}}, \lambda'_{\mathtt{undec}} \setminus C)$. By the definition of $C$, it is easy to check that $\lambda^*_{\mathtt{in}} \cap \lambda^*_{\mathtt{out}} = \{\}$, $\lambda^*_{\mathtt{in}} \cap \lambda^*_{\mathtt{undec}} = \{\}$, and $\lambda^*_{\mathtt{out}} \cap \lambda^*_{\mathtt{undec}} = \{\}$ hold. Thus, $\lambda^*$ is a well-defined labelling. In the definition of admissible labelling there is no condition for label an argument undec. Further, $\lambda^*_{\mathtt{out}} = \lambda'_{\mathtt{out}}$, $\lambda'_{\mathtt{in}} \subseteq \lambda^*_{\mathtt{in}}$ and $\lambda'$ is an admissible labelling, therefore, the condition for arguments which are labelled out in $\lambda^*$ are also satisfied. For argument $a$ with $\lambda^*(a) = \mathtt{in}$ either $a \in \lambda'_{\mathtt{in}}$ or $a \mapsto \mathtt{in} \in C \subseteq \lambda_{\mathtt{in}}$. Each of them implies that for each conflict $(B,a)$ there exists $b \in B$ s.t. $\lambda^*(b) = \mathtt{out}$, since $\lambda, \lambda'$ are admissible labelling and $\lambda_{\mathtt{out}} \subseteq \lambda'_{\mathtt{out}}$. Thus, $\lambda^*$ is an admissible labelling.

For condition 6, first we show that $\lambda'_{\mathtt{in}} \cap (\lambda'_{\mathtt{out}} \cup C) = \{\}$. To this end, let $a \in C$ we show that $a \notin \lambda'_{\mathtt{in}}$. Since $C \subseteq \lambda_{\mathtt{out}}$, there exists $(B,a) \in R$ such that $\lambda(b) = \mathtt{in}$ for all $b \in B$. By the assumption of this condition, namely $\lambda_{\mathtt{in}} \subseteq \lambda'_{\mathtt{in}}$, the relation $B \subseteq \lambda'_{\mathtt{in}}$ holds. Thus, $\lambda'(a) \neq \mathtt{in}$. Since $\lambda' \in adm_{\mathscr{L}}(F)$, to show that $\lambda^* \in adm_{\mathscr{L}}(F)$ it is enough to show that each $a \in C$ is actually labelled out in $\lambda^*$. This condition is trivially satisfied, because $C \subseteq \lambda_{\mathtt{out}}$, $\lambda_{\mathtt{in}} \subseteq \lambda'_{\mathtt{in}}$ and $\lambda' \in adm_{\mathscr{L}}(F)$.

For condition 7, it is enough to show that $\lambda_{\mathtt{in}} \cap \lambda'_{out} = \{\}$, $\lambda_{\mathtt{in}} \cap (\mathtt{Args}_{\mathbb{L}} \setminus (\lambda_{\mathtt{in}} \cup \lambda'_{\mathtt{out}})) = \{\}$, and $\lambda_{\mathtt{out}} \cap (\mathtt{Args}_{\mathbb{L}} \setminus (\lambda_{\mathtt{in}} \cup \lambda'_{\mathtt{out}})) = \{\}$. Let $\lambda^* = (\lambda_{\mathtt{in}}, \lambda'_{\mathtt{out}}, \mathtt{Args}_{\mathbb{L}} \setminus (\lambda_{\mathtt{in}} \cup \lambda'_{\mathtt{out}}))$. For $a$ with $\lambda^*(a) = \mathtt{in}$ ($a \in \lambda_{\mathtt{in}}$) it holds that $a \notin \lambda_{out}$, because $\lambda \in adm_{\mathscr{L}}(F)$. Further, since $\lambda'_{\mathtt{out}} \subseteq \lambda_{\mathtt{out}}$, $a \notin \lambda'_{out}$, that is, $a \notin \lambda^*_{\mathtt{out}}$. If $a \in \lambda^*_{\mathtt{out}}$ ($a \in \lambda'_{\mathtt{out}}$), since $\lambda'_{\mathtt{out}} \subseteq \lambda_{\mathtt{out}}$, $a \in \lambda_{\mathtt{out}}$. Therefore, $a \notin \lambda_{\mathtt{in}}$ as $\lambda \in adm_{\mathscr{L}}(F)$. Thus, $a \notin \lambda^*_{\mathtt{in}}$. Moreover, $a$ is included either in $\lambda^*_{\mathtt{in}}$ or $\lambda^*_{\mathtt{out}}$ if and only if $a \notin (\mathtt{Args}_{\mathbb{L}} \setminus (\lambda_{\mathtt{in}} \cup \lambda'_{\mathtt{out}}))$. On the other hand, condition of admissible labelling for arguments labelled out in $\lambda^*$ are trivially satisfied as $\lambda^*_{\mathtt{in}} = \lambda_{\mathtt{in}}$ and $\lambda^*_{\mathtt{out}} \subseteq \lambda_{\mathtt{out}}$. Toward a contradiction, assume that $\lambda^*(a) = \mathtt{in}$ and there exists conflict $(B,a)$ s.t. for each $b \in B$, $\lambda^*(b) \neq \mathtt{out}$, that is, $\lambda^*(b) = \mathtt{in}/\mathtt{undec}$. If $\lambda^*(b) = \mathtt{in}$, then $\lambda(b) = \mathtt{in}$ and if $\lambda^*(b) = \mathtt{undec}$, then $b \notin \lambda'_{\mathtt{out}} \subseteq \lambda_{\mathtt{out}}$. That is, $\lambda(b) \neq \mathtt{out}$ for each $b \in B$. This is a contradiction with the assumption that $\lambda \in adm_{\mathscr{L}}(F)$.

For condition 8 let $\lambda = (\emptyset, \emptyset, \mathtt{Args}_{\mathbb{L}})$. The conditions of admissible labelling for arguments labelled with in or out in $\lambda$ are satisfied, there is no such an argument, and there is no condition for arguments labelled with undec in the conditions of admissible labelling. Thus, $\lambda \in adm_{\mathscr{L}}(F)$. □

The condition in the above proposition are necessary for an labelling-set to be *adm*-realizable, but it remains open whether they are also sufficient.

Finally, we give a characterisation of the signature of grounded semantics.

**Proposition 9.** *The signature $\Sigma^{grd_{\mathscr{L}}}_{SETADF}$ is given by sets $\mathbb{L}$ of labellings such that*

- $|\mathbb{L}| = 1$,
- *if $\lambda \in \mathbb{L}$ assigns one argument to* out *then $\lambda_{\mathtt{in}} \neq \emptyset$.*

*Proof.* We first show that for each SETAF $F$ the set $grd_{\mathscr{L}}(F)$ satisfies the conditions of the proposition. Toward a contradiction assume that there are $\lambda, \lambda' \in grd_{\mathscr{L}}$ with $\lambda \neq \lambda'$. By the definition of grounded labelling $\lambda_{\mathtt{in}}$ $\lambda_{\mathtt{in}}$ are $\subseteq$-minimal among all complete labelling, thus, $\lambda_{\mathtt{in}} = \lambda'_{\mathtt{in}}$. Assume that $\lambda_{\mathtt{out}} \subset \lambda'_{\mathtt{out}}$. Since each grounded labelling is a conflict-free, for each $a$ with $a \in \lambda'_{\mathtt{out}}$ there is $(B,a)$ such that $B \subseteq \lambda'_{\mathtt{in}}$. Since $\lambda_{\mathtt{in}} = \lambda'_{\mathtt{in}}$, $a \in \lambda_{\mathtt{out}}$. Therefore, $\lambda = \lambda'$. Now, assume that $\lambda \in grd_{\mathscr{L}}(F)$ assigns an argument $a$ to out. By the definition of conflict-free labeling there is an attack $(B,a)$ such that $B \subseteq \lambda_{\mathtt{in}}$.

Now assume that $\mathbb{L}$ satisfies all the conditions. We give a SETAF $F_{\mathbb{L}} = (A_{\mathbb{L}}, R_{\mathbb{L}})$ with $grd_{\mathscr{L}}(F_{\mathbb{L}}) = \mathbb{L}$.

$$A_{\mathbb{L}} = \mathtt{Args}_{\mathbb{L}}$$
$$R_{\mathbb{L}} = \{(\lambda_{\mathtt{in}}, a) \mid \lambda \in \mathbb{L}, \lambda(a) = \mathtt{out}\} \cup \{(\lambda_{\mathtt{in}} \cup \{a\}, a) \mid \lambda \in \mathbb{L}, \lambda(a) = \mathtt{undec}\}$$

Consider the unique $\lambda \in \mathbb{L}$ and the unique $\lambda^G \in grd_{\mathscr{L}}(F_{\mathbb{L}}$. For each argument $a \in \lambda_{\mathtt{in}}$ we have that $a$ is not attacked in $F_{\mathbb{L}}$ and thus $a \in \lambda^G_{\mathtt{in}}$. For each argument $a \in \lambda_{\mathtt{out}}$ there is an attack $(\lambda_{\mathtt{in}}, a)$ in $F_{\mathbb{L}}$ and as $\lambda_{\in} \subseteq \lambda^G_{\in}$ by the definition of complete labellings we have $a \in \lambda^G_{\mathtt{out}}$. Finally for each argument $a \in \lambda_{\mathtt{undec}}$ the attack $(\lambda_{\mathtt{in}} \cup \{a\}, a)$ is the only attack towards $a$ in $F_{\mathbb{L}}$. Thus, by the definition of complete labellings, we have that $a$ is neither labelled in nor out in $F_{\mathbb{L}}$ and therefore $a \in \lambda^G_{\mathtt{undec}}$. We obtain that $\lambda^G = \lambda$ and thus $grd_{\mathscr{L}}(F_{\mathbb{L}}) = \mathbb{L}$. □

Notice that the above proof basically exploits that grounded semantics is a unique status semantics and based on admissibility. The results thus immediately extends to other semantics satisfying these two properties, e.g. to ideal or eager semantics [10].

# 5 On the relation of SETAFS and Support-Free ADFs

In order to compare SETAFs with SFADFS we switch to SETADF notation (cf. Theorem 3). We start with the formal definition of the signatures of sub-classes of ADFs.

**Definition 14.** We define the signature of a class of ADFs $\mathscr{D}$ under semantics $\sigma$ as

$$\Sigma^{\sigma}_{\mathscr{D}} = \{\sigma(D) \mid D \in \mathscr{D}\}.$$

We start with the observation that each SETADF can be rewritten as a SETADF that is a SFADF.

**Lemma 10.** *For Each SETADF $D = (S,L,C)$ there is an equivalent SETADF $D' = (S,L',C')$ that is also a SFADF, i.e. for each $s \in S$, $\varphi_s \in C$, $\varphi'_s \in C'$ we have $\varphi_s \equiv \varphi'_s$ over atoms $S$.*

*Proof.* Given a SETADF $D$, by Definition 9, each acceptance condition is a CNF over negative literals and thus does not have any support link which is not redundant. We can thus obtain $L'$ by removing the redundant links from $L$ and $C'$ by, in each acceptance condition, deleting the clauses that are super-sets of other clauses. □

As discussed in [15] in general SETAFs translate to Bipolar ADFs that contain attacking and redundant links. However, when we consider SETAFs without redundant attacks we obtain a SFADF.

We next characterise the acceptance conditions of SFADF that can be rewritten as collective attacks.

**Lemma 11.** *Let $D = (S, L, C)$ be a SFADF. If $s \in S$ has at least one incoming link then the acceptance condition of argument $s$ can be written in conjunctive normal form containing only negative literals.*

*Proof.* Since the acceptance condition of each argument in an ADF is indicated by a propositional formula, it can be transformed to CNF. It remains to show that each of the resulting formulas in CNF can be transformed into a CNF that consists of only negative literals. Let $\varphi_s$ be acceptance condition of an argument $s$ with an incoming link $(t,s)$ in conjunctive normal form that contains $t$ as positive literal. If $t$ doesn't not appear in any model of the formula $\varphi_s$ we can safely delete the literal $t$ from each clause of the CNF to obtain an equivalent CNF without $t$. Otherwise let $v$ be a model of $\varphi_s$ with $v(t) = \mathbf{t}$ then, as $(t,s)$ is attacking, we have that $v|^t_{\mathbf{f}}$ is a model of $\varphi_s$. Again we can safely delete the literal $t$ from each clause of the CNF to obtain an equivalent CNF without $t$. That is, we can iteratively remove positive literals from the CNF to obtain a CNF with only negative literals. $\square$

While we can rewrite each attack relation as a collection of collective attacks this is not true for arguments that have no incoming link. That is, an argument with unsatisfiable acceptance condition cannot be modeled in a SETADF. One can use this fact to show that there are some difference in expressiveness between SETADFs and SFADFs.

For instance, given an interpretation-set $\mathbb{V} = \{\{s \mapsto \mathbf{f}\}\}$ which is *prf*-realizable in SFADFs. It is easy to check that there is no SETADF that realize $\mathbb{V}$ under preferred semantics (cf. Proposition 5). That is we obtain that SFADFs are strictly more expressive than SETADFs for all ADF semantics under our considerations.

**Theorem 12.** $\Sigma^{\sigma}_{SETADF} \subsetneq \Sigma^{\sigma}_{SFADF}$, *for* $\sigma \in \{adm, stb, mod, com, prf, grd\}$.

*Proof.* In the following first we show that $\Sigma^{\sigma}_{\text{SETADF}} \subseteq \Sigma^{\sigma}_{SFADF}$, for $\sigma \in \{adm, stb, mod, com, prf, grd\}$. Let $\mathbb{V}$ be an interpretation-set which is $\sigma$-realizable in SETADFs for $\sigma \in \{adm, stb, mod, com, prf, grd\}$. Thus, there exists a SETADF $D = (S, L, C)$ such that $\mathbb{V} = \sigma(D)$. By Lemma 10 $D$ has an equivalent SFADF and thus $\mathbb{V} \in \Sigma^{\sigma}_{\text{SFADF}}$. Thus, for $\sigma \in \{adm, stb, mod, com, prf, grd\}$, $\Sigma^{\sigma}_{\text{SETADF}} \subseteq \Sigma^{\sigma}_{\text{SFADF}}$.

We complete the proof of theorem by showing $\Sigma^{\sigma}_{\text{SFADF}} \not\subseteq \Sigma^{\sigma}_{\text{SETADF}}$, for $\sigma \in \{adm, stb, mod, com, prf, grd\}$. To investigate, $\Sigma^{adm}_{\text{SETADF}} \subsetneq \Sigma^{adm}_{\text{SFADF}}$, let $\mathbb{V} = \{\{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}\}, \{a \mapsto \mathbf{u}, b \mapsto \mathbf{f}\}, \{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}\}\}$ be an interpretation-set. A witness of *adm*-realizability of $\mathbb{V}$ in SFADFs is $D = (\{a,b\}, \{\varphi_a = \neg a \vee \neg b, \varphi_b = \bot\})$. However, $\mathbb{V}$ is not realizable by any SETADF for admissible (cf. Proposition 8). Thus, $\Sigma^{adm}_{\text{SFADF}} \not\subseteq \Sigma^{adm}_{\text{SETADF}}$. To show $\Sigma^{\sigma}_{\text{SFADF}} \not\subseteq \Sigma^{\sigma}_{\text{SETADF}}$, for $\sigma \in \{stb, mod, com, prf, grd\}$, let $\mathbb{V} = \{\{a \mapsto \mathbf{f}\}\}$. The interpretation $\mathbb{V}$ is $\sigma$-realizable in SFADFs for $\sigma \in \{stb, mod, com, prf, grd\}$, and a witness of $\sigma$-realizability of $\mathbb{V}$ in SFADFs is $D = (\{a\}, \{\varphi_a = \bot\})$. However, $\mathbb{V}$ cannot be realized by any SETADF (cf. Propositions 4–9). Hence, $\Sigma^{\sigma}_{\text{SETADF}} \subsetneq \Sigma^{\sigma}_{SFADF}$, for $\sigma \in \{adm, stb, mod, com, prf, grd\}$. $\square$

The interpretation-sets $\mathbb{V}$ used in the proof of Theorem 12 to show that SFADFs are more expressive than SETADFs, for $\sigma \in \{adm, stb, mod, com, prf, grd\}$, are very special interpretation-sets, which are only $\sigma$-realizable by a SFADF containing an argument with unsatisfiable acceptance condition.

In [17, 18], it is shown that the unsatisfiable condition ($\varphi_a = \bot$) has no direct representation in SETAFs and in SETADFs, as well. However, there are SFADFs with an unsatisfiable acceptance condition that have an equivalent SETADF, i.e. a SETADF that has the same interpretations. For instance, the interpretation-set $\mathbb{V} = \{\{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}\}\}$ can be $\sigma$-realized by SFADF $D = (\{a,b\}, \{\varphi_a = \bot, \varphi_b = \top\})$ under $\sigma \in \{stb, mod, com, prf, grd\}$. The SFADF $D$is a witness of $\sigma$-realizability of $\mathbb{V}$ in SFADF and consists of unsatisfiable condition, however, we cannot conclude that $\mathbb{V}$ is not $\sigma$-realizable in SETADFs. Actually, for the SETADF $D' = (\{a,b\}, \{\varphi_a = \neg b, \varphi_b = \top\})$ we have $\mathbb{V} = \sigma(D')$ and thus that $\mathbb{V} \in \Sigma^{\sigma}_{\text{SETADF}}$ for $\sigma \in \{stb, mod, com, prf, grd\}$.

## 5.1 On the exact difference of SETADFs and SFADFs

We already have seen that SFADFs are strictly more expressive than SETADFs. In this section we investigate the exact difference in the signatures of SFADFs and SETADFs

As each SFADF without unsatisfiable acceptance condition can be translated into a SETADF the interpretations in $\Sigma^{\sigma}_{\text{SFADF}} \setminus \Sigma^{\sigma}_{\text{SETADF}}$ must be based on unsatisfiable acceptance conditions. An argument with unsatisfiable acceptance conditions allows for interpretations that assign an argument to $\mathbf{f}$ without assigning an argument to $\mathbf{t}$. We will denote the set of interpretation-sets containing such an interpretation by $\Delta_\sigma$.

**Definition 15.** Let $\sigma$ be a semantics of ADFs. $\Delta_\sigma$ is a subset of $\Sigma^{\sigma}_{\text{SFADF}}$ such that:

$$\Delta_\sigma = \{\sigma(D) \mid D \in \text{SFADF}, \exists v \in \sigma(D) \text{ s.t. } \forall a\ v(a) \in \{\mathbf{f}, \mathbf{u}\} \wedge \exists a\ v(a) = \mathbf{f}\}.$$

Moreover $\bar{\Delta}_\sigma = \Sigma^{\sigma}_{SFADF} \setminus \Delta_\sigma$.

In this section we will first show that $\Delta_\sigma$ characterises the difference between SFADFS and SETADFs and then further investigate the sets $\Delta_\sigma$ for the different semantics.

We first show that each SFADF realizing an interpretation-set of $\Delta_\sigma$ has an argument with an unsatisfiable acceptance conditions (and thus is not a SETADF).

**Lemma 13.** *Given an interpretation-set $\mathbb{V} \in \Delta_\sigma$, for $\sigma \in \{adm, stb, mod, com, prf, grd\}$. Let $v \in \mathbb{V}$ be a non-trivial interpretation in which $v(a) = \mathbf{f}/\mathbf{u}$, for each argument a. In all SFADFs that realize $\mathbb{V}$ under $\sigma$, the acceptance conditions of all arguments assigned to $\mathbf{f}$ by $v$ are equal to $\bot$.*

*Proof.* let $D$ be a SFADF that realizes $\mathbb{V}$ under $\sigma$, for $\sigma \in \{adm, stb, mod, com, prf, grd\}$. Let $v \in \mathbb{V}$ be an non-trivial interpretation that assigns all arguments either to $\mathbf{f}$ or $\mathbf{u}$. Toward a contradiction, assume that there exists an argument $a$ which is assigned to $\mathbf{f}$ by $v$, and $\varphi_a \neq \bot$ in $D$.

- First we show that $\mathbb{V}$ cannot be *adm*-realizable in SFADFs. Since $a$ is assigned to $\mathbf{f}$ in $v$ the acceptance condition of $a$ cannot be equal to $\top$. By Lemma 11, the acceptance condition of $a$ is in CNF and having only negative literals. Since all $b \in par(a)$ are either assigned to $\mathbf{f}$ or $\mathbf{u}$ by $v$, $\varphi_a^v$ cannot be unsatisfiable. That is, $v(a) \not\leq_i \Gamma_D(v)(a)$. Therefore, $v$ is not an admissible interpretation of $D$. Thus, $\mathbb{V}$ consists of $v$ is not *adm*-realizable in SFADF.

- To complete the proof we show that $\mathbb{V}$ cannot be $\sigma$-realizable for $\sigma \in \{stb, mod, com, prf, grd\}$ in SFADF $D$. Suppose $\mathbb{V} = \sigma(D)$ for $\sigma \in \{stb, mod, com, prf, grd\}$, that is, each interpretation of $\mathbb{V}$ is a $\sigma$ interpretation. Since each $\sigma$ interpretation, for $\sigma \in \{stb, mod, com, prf, grd\}$, is an admissible interpretation, $v$ is an admissible interpretation, as well. This is a contradiction by the previous item in which it is shown that $v$ is not an admissible interpretation.

Hence, if $\mathbb{V} \in \Delta_\sigma$ for $\sigma \in \{adm, stb, mod, com, prf, grd\}$ in all SFADFs that realize $\mathbb{V}$ under $\sigma$, $\varphi_a = \bot$ for all $a$ assigned to $\mathbf{f}$ in $v$, $\square$

From Lemma 13, and the fact that unsatisfiable conditions do not have a direct analogue in SETAFS, via [17], we have the following theorem.

**Theorem 14.** *Given an interpretation-set $\mathbb{V} \in \Delta_\sigma$. The interpretation-set $\mathbb{V}$ is not $\sigma$-realizable in SETADFs, for $\sigma \in \{adm, stb, prf, mod, com, grd\}$.*

*Proof.* Since $\mathbb{V} \in \Delta_\sigma$, the interpretation-set $\mathbb{V}$ is $\sigma$-realizable in SFADFs for $\sigma \in \{adm, stb, prf, mod, com, grd\}$. Let $v \in \mathbb{V}$ be an interpretation in which all arguments are assigned to either $\mathbf{f}$ or $\mathbf{u}$. By Lemma 13, in any SFADF $D$ which is a witness of $\sigma$-realizability of $\mathbb{V}$, that is $\mathbb{V} = \sigma(D)$ for $\sigma \in \{adm, stb, prf, mod, com, grd\}$, the acceptance conditions of arguments which are assigned to $\mathbf{f}$ in $\mathbb{V}$ are unsatisfiable. That is $D$ is not a SETADF. $\square$

By the above we have that all interpretation-sets in $\Delta_\sigma$ are not realizable with SETADFs. It remains to show that all other interpretation-sets in $\Sigma^\sigma_{SFADF}$ can be realized with SETADFs, i.e. we have to show that $\bar{\Delta}_\sigma = \Sigma^\sigma_{\text{SETADF}}$.

Theorem 14, is a motivation to study whether $\Delta_\sigma$ is the only set elements of which cannot be realized by any SETAFs under $\sigma$. We first show this for admissible semantics.

**Lemma 15.** *Given an interpretation-set $\mathbb{V} \in \bar{\Delta}_{adm}$. Each SFADF that realizes $\mathbb{V}$ has no unsatisfiable acceptance condition.*

*Proof.* Given an arbitrary interpretation-set $\mathbb{V} \in \bar{\Delta}_{adm}$ Suppose to the contrary that there exist a SFADF $D = (S, L, C)$ such that $adm(D) = \mathbb{V}$ and there exists $s \in S$ such that $\varphi_s = \bot$. Then, an interpretation $v$ that assigns $s$ to $\mathbf{f}$ and all other arguments of $S$ to $\mathbf{u}$ is an admissible interpretation of $D$. This is a contradiction with the assumption that $\mathbb{V} \notin \Delta_{adm}$. Thus, the acceptance condition of all arguments of all SFADFs that realize $\mathbb{V}$ under $\sigma$ is either $\top$ or the argument has at least one incoming link. $\square$

By Lemma 15 and Lemma 11 each $\mathbb{V} \in \bar{\Delta}_{adm}$ can be realized as SETADF.

**Proposition 16.** $\Sigma^{adm}_{SFADF} = \Sigma^{adm}_{SETADF} \cup \Delta_{adm}$.

Note that Lemma 15, does not hold for $\sigma \in \{stb, mod, prf, com, grd\}$. That is, a witness of $\sigma$-realizability of an interpretation-set $\mathbb{V} \in \bar{\Delta}_\sigma$ may contain arguments with unsatisfiable acceptance condition. For instance, the interpretation-set $\mathbb{V} = \{\{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}\}\}$ that $\mathbb{V} \in \bar{\Delta}_\sigma$ can also be realized by SFADF $D = (\{a,b\}, \{\varphi_a = \neg b, \varphi_b = \bot\})$, for $\sigma \in \{stb, mod, prf, com, grd\}$, in which there is an argument with unsatisfiable acceptance condition.

Proposition 17 shows that under which condition an interpretation-set $\mathbb{V} \in \Sigma^\sigma_{\text{SFADF}}$ is $\sigma$-realizable in SETADFs, for $\sigma \in \{stb, grd, prf, mod, com\}$, beside the fact that a witness of $\sigma$-realizability of $\mathbb{V}$ in SFADFs consists of an argument with unsatisfiable acceptance condition.

**Proposition 17.** *$\Sigma^\sigma_{SFADF} = \Sigma^\sigma_{SETADF} \cup \Delta_\sigma$, for $\sigma \in \{stb, grd, prf, mod, com\}$.*

*Proof.* It remains to show that each $\mathbb{V} \in \bar{\Delta}_\sigma$ can be $\sigma$-realizable in SETADFs. Consider a witness $D = (S, L, C)$ of $\sigma$-realizability of $\mathbb{V}$ in SFADFs. If there is no argument with acceptance condition $\bot$, then $\mathbb{V}$ is $\sigma$-realizable in SETADFs by Lemma 11. Assume there are arguments $s_1, \dots, s_\ell$ with acceptance condition $\bot$ and thus $s_i$ is denied by any $v_i \in \mathbb{V}$. Notice that as $\mathbb{V} \in \bar{\Delta}_\sigma$, we then have that each $v_i \in \mathbb{V}$ assigns at least one argument to $\mathbf{t}$. For each $v_i \in \mathbb{V}$ let $b_i$ be an argument such that $v_i(b_i) = \mathbf{t}$.

We construct a SETADF $F_D = (S, L', C')$ such that $C'$ is a collection of $\varphi'_a$ as follows.

$$\varphi'_a = \begin{cases} \varphi_a & \text{if } \varphi_a \neq \bot \\ \neg a \wedge \bigwedge_{v_i \in \mathbb{V}} \neg b_i & \text{otherwise.} \end{cases}$$

It is now easy to verify that $\mathbb{V} = \sigma(F_D)$ and as, by construction $F_D$, has no argument with acceptance condition $\bot$, by Lemma 11, $\mathbb{V}$ is $\sigma$-realizable in SETADFs. $\square$

In the proof of Proposition 17, it is shown constructively how an interpretation-set that is $\sigma$-realizable in SFADFs for $\sigma \in \{stb, grd, prf, mod, com\}$ and each element of which accepts an argument, can be realized by a SETADF.

**Example 3.** Let $\mathbb{V} = \{\{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, d \mapsto \mathbf{f}\}, \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}, d \mapsto \mathbf{t}\}\}$ be an interpretation-set. A witness of $\sigma$-realizability of $\mathbb{V}$ in SFADFs, for $\sigma \in \{stb, prf, mod\}$ can be $D = (\{a,b,c,d\}, \{\varphi_a = \bot, \varphi_b = \neg a, \varphi_c = \neg d, \varphi_d = \neg c\})$. By the construction in the proof of Proposition 17, a SETADF $F_D$ constructed based on SFADF $D$ is, $F_D = (\{a,b,c,d\}, \{\varphi_a = \neg a \wedge \neg b, \varphi_b = \neg a, \varphi_c = \neg d, \varphi_d = \neg c\})$.

Theorem 18 summarizes or results and shows that $\Delta_\sigma$ is the only set of interpretation-sets that cannot be realized by any SETADF, for $\sigma \in \{adm, stb, mod, com, prf, grd\}$.

**Theorem 18.** *$\Sigma^\sigma_{SETADF} = \bar{\Delta}_\sigma$, for $\sigma \in \{adm, stb, mod, com, prf, grd\}$.*

Given that the difference in $\Sigma^\sigma_{\text{SETADF}}$ and $\Sigma^\sigma_{\text{SFADF}}$ is captured by the sets $\Delta_\sigma$ we further investigate the properties of these sets.

We first show that for $\sigma \in \{stb, mod, prf\}$ the set $\Delta_\sigma$ only contains interpretation-sets that contain only a single interpretation.

**Lemma 19.** *For $\mathbb{V} \in \Delta_\sigma$ and $\sigma \in \{stb, mod, prf\}$ we have $|\mathbb{V}| = 1$. For $\sigma \in \{stb, mod\}$ the unique $v \in \mathbb{V}$ assigns all arguments to* $\mathbf{f}$.

*Proof.* Toward a contradiction assume that there exists $\mathbb{V} \in \Delta_\sigma$, for $\sigma \in \{stb, mod, prf\}$, such that $|\mathbb{V}| > 1$. Let $v \in \mathbb{V}$ be an interpretation that assign all arguments to either **f** or **u** (since $\mathbb{V} \in \Delta_\sigma$, such a $v$ exists). By Lemma 13, the acceptance condition of all arguments that are assigned to **f** by $v$ is equal to $\perp$ in all SFADFs that realize $\mathbb{V}$ under $\sigma \in \{stb, mod, prf\}$. Let $D = (S, L, C)$ be a witness of $\sigma$-realizibility of $\mathbb{V}$ in SFADFs, under $\sigma \in \{stb, mod, prf\}$. If all arguments are assigned to **f** in $v$, the acceptance conditions of all arguments are $\perp$ in SFADF $D$. Thus, $|\sigma(D)| = 1$ for $\sigma \in \{stb, mod, prf\}$. This is a contradiction by the assumption that the cardinal of the interpretation-set $\mathbb{V}$ is more than one.

Assume that $v$ assigns some arguments to **u**. Thus, $V$ cannot be *mod* or *stb*-realized in any ADF. It remains to show that the interpretation-set $\mathbb{V}$ in question is not *prf*-realizable. Let $B \subset S$ such that $v(b) = \mathbf{u}$ for $b \in B$. For each $s \in S \setminus B$, by Lemma 13, $\varphi_s = \perp$ in $D$. Therefore, in all $v' \in \mathbb{V}$, $v'(s) = \mathbf{f}$ for $s \in S \setminus B$. For each $v' \neq v$ in $\mathbb{V}$ there exists at least $b \in B$ such that $v'(b) \neq \mathbf{u}$, therefore, $v < v'$. By the definition of preferred interpretations $v$ cannot be a preferred interpretation. Therefore, the assumption $|\mathbb{V}| > 1$ is not correct. Thus, if $\mathbb{V} \in \Delta_\sigma$, for $\sigma \in \{stb, mod, prf\}$, then $\mathbb{V}$ consist of only one interpretation. □

In other words each interpretation-set which is $\sigma$-realizable in SFADFs and contains at least two interpretation can be realized in SETADFs, for $\sigma \in \{stb, prf, mod\}$

However, this is not a sufficient condition for admissible and complete semantics, shown in Example 4.

**Example 4.** Let $\mathbb{V} = \{\{a \mapsto \mathbf{f}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}\}, \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}\}, \{a \mapsto \mathbf{f}, b \mapsto \mathbf{f}, c \mapsto \mathbf{t}\}\}$. A witness of *com*-realizability of $\mathbb{V}$ in SFADFs can be $D = (\{a, b, c\}, \{\varphi_a = \perp, \varphi_b = \neg c, \varphi_c = \neg b\})$ and $\mathbb{V}' = grd(D) = \{\{a \mapsto \mathbf{f}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}\}\}$. However, there is no SETADF that realize $\mathbb{V}$ under *com* and $\mathbb{V}'$ under *grd*.

Finally we give an alternative characterisation for complete semantics.

**Theorem 20.** *Let $\mathbb{V} \in \Sigma^{com}_{SFADF}$ and $g = \bigsqcap_{v \in \mathbb{V}} v$, then $\mathbb{V} \in \Sigma^{com}_{SETADF}$ if and only if $g$ is grd-realizable in SETADFs.*

*Proof.* Let $D = (S, L, C)$ be a witness of realizability of $\mathbb{V}$ in SFADFs under complete semantics. Assume that $\mathbb{V}$ is *com*-realizable in SETADFs by $D' = (S, L', C')$ then it is clear that $g = grd(D) = grd(D')$ and thus $g$ is *grd*-realizable in SETADFs. To show the if part of the theorem, assume that $g$ is *grd*-realizable in SETADFs. By Theorem 14, either (a) there exists $s \in S$ which is assigned to **t** in $g$ or (b) all $s \in S$ are assigned to **u**. In case (a) this $s$ is assigned to **t** by any $v \in \mathbb{V}$. By the method presented in the proof of Proposition 17 this $\mathbb{V}$ is *com*-realizable in SETAFs. In case (b) we have that in $D$ there is no argument with acceptance condition $\perp$ (or $\top$) and thus by Lemma 11 $D$ is equivalent to a SETADF.

□

## 5.2 SETADFs vs. Symmetric SFADFs

In this section we consider the special subclass of SFADF in which the attack link relation is symmetric in the sense of [5].

**Definition 16.** An ADF $D = (S, L, C)$ is *symmetric* if $L$ is irreflexive and symmetric and $L$ does not contain any redundant links.

**Definition 17.** A support free ADF $D = (S, L, C)$ is a *support free symmetric ADF*(SFSADF for short) if it is symmetric.

In Lemma 21, the sufficient condition under which a SFSADF can be written as a SETADF is investigated. Notice that in symmetric ADFs, due to the lack of redundant links, arguments with unsatisfiable acceptance condition are always isolated arguments.

**Lemma 21.** *Given a SFSADF D which does not contain any isolated argument with unsatisfiable acceptance condition. The SFSADF D can be written as a SETADF.*

*Proof.* Assume that $D = (S, L, C)$ is a SFSADF in which there is no isolated argument $s \in S$ such that $\varphi_s = \bot$. Since each SFSADF is a SFADF, $D$ is a SFADF and by the assumption of the lemma, $D$ does not contain any argument with unsatisfiable acceptance condition. Via Lemma 11, $D$ can be rewritten as a SETADF. $\square$

**Lemma 22.** *Let D be an SFSADF with no isolated argument. The unique grounded interpretation of D is the trivial interpretation,* $v_{\mathbf{u}}$.

*Proof.* We show that for any SFSADF $D = (S, L, C)$ with no isolated argument, $\Gamma_D(v_{\mathbf{u}}) = v_{\mathbf{u}}$. Let $s$ be an argument. Let $v_1$ be an interpretation in which all parents of $s$ are assigned to $\mathbf{t}$ and let $v_2$ be an interpretation in which all $par(s)$ are assigned to $\mathbf{f}$. Since $D$ is an SFSADF, the former interpretation shows that $\varphi_s^{v_{\mathbf{u}}}$ is not irrefutable and the latter interpretation says that $\varphi_s^{v_{\mathbf{u}}}$ is not unsatisfiable. Therefore, for each argument $s$, $\Gamma_D(v_{\mathbf{u}})(s) = \mathbf{u}$. $\square$

We next combine the two above Lemmas to obtain necessary and sufficient conditions for realizability of $\mathbb{V} \in \Sigma^{\sigma}_{\text{SFSADF}}$ in SETADFs for $\sigma \in \{adm, com, grd\}$.

**Proposition 23.** *Given a SFSADF D we have that*

1. *$adm(D) \in \Sigma^{adm}_{SETADF}$ iff D does not contain any argument with unsatisfiable acceptance condition; and*
2. *$\sigma(D) \in \Sigma^{\sigma}_{SETADF}$ for $\sigma \in \{grd, com\}$ iff either (a) D contains an isolated argument with acceptance condition $\top$ or (b) D does not contain any argument with unsatisfiable acceptance condition.*

*Proof.* 1) The "if" direction is immediate by Lemma 21. For the "only if" direction assume that $D$ contains an argument $a$ with unsatisfiable acceptance conditions. Then there is a admissible interpretation that assigns a to $\mathbf{f}$ and all the other arguments to $\mathbf{u}$. By Theorem 18 such a $\sigma(D)$ is not in $\Sigma^{\sigma}_{\text{SETADF}}$.

2) For "if" direction first assume (a) holds, i.e. there is an argument $a$ with $c_s = \top$. Then each complete interpretation assigns $a$ to $\mathbf{t}$ and thus, by Theorem 18, $\sigma(D) \in \Sigma^{\sigma}_{\text{SETADF}}$. Otherwise, (b) holds and $\sigma(D) \in \Sigma^{\sigma}_{\text{SETADF}}$ is immediate by Lemma 21. For the "only if" direction assume that $D$ contains arguments with unsatisfiable acceptance conditions but no isolated arguments with acceptance condition $\top$. By Lemma 22 we then have an interpretation $\lambda \in \mathbb{V}$ that assigns some arguments to $\mathbf{f}$ and all the other arguments to $\mathbf{u}$. By Theorem 18 such a $\sigma(D)$ is not in $\Sigma^{\sigma}_{\text{SETADF}}$. $\square$

On the other hand, the conditions in the above proposition are not necessary for $\sigma \in \{prf, stb, mod\}$ as indicated in Example 5.

**Example 5.** Let $\mathbb{V} = \{\{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}\}, \{a \mapsto \mathbf{f}, b \mapsto \mathbf{f}, c \mapsto \mathbf{t}\}\}$ be an interpretation-set. A witness of $\sigma$-realizability of $\mathbb{V}$ in SFSADFs for $\sigma \in \{stb, mod, prf\}$, is $D = (\{a,b,c\}, \{\varphi_a = \bot, \varphi_b = \neg c, \varphi_c = \neg b\})$. $D$ is a SFADF that contains an argument $a$ such that $\varphi_a = \bot$, however, $\mathbb{V} \notin \Delta_\sigma$. Thus, by Proposition 17, $\mathbb{V}$ can also be realized by a SETADF, a witness of which is $D' = (\{a,b,c\}, \{\varphi_a = \neg a \wedge \neg b \wedge \neg c, \varphi_b = \neg c \wedge \neg a, \varphi_c = \neg b \wedge \neg a\})$.

In [6] it is proven that $\Sigma^\sigma_{\text{SFSADF}} = \Sigma^\sigma_{\text{SFADF}}$, for $\sigma \in \{stb, mod\}$, and $\Sigma^\sigma_{\text{SFSADF}} \subsetneq \Sigma^\sigma_{\text{SFADF}}$, for $\sigma \in \{adm, grd, com, prf\}$. On the other hand, $\Delta_\sigma \not\subseteq \Sigma_{\text{SFSADF}}$ for $\sigma \in \{adm, grd, com, prf\}$. For instance, let $\mathbb{V} = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{f}\}$. It is clear that $\mathbb{V} \in \Delta_\sigma$ and $\mathbb{V} \notin \Sigma^\sigma_{\text{SFSADF}}$, for $\sigma \in \{adm, grd, com, prf\}$. Let $\Delta'_\sigma$ be a subset of $\Delta_\sigma$ that is realizable in $\Sigma^\sigma_{\text{SFSADF}}$, for $\sigma \in \{adm, grd, com, prf\}$. Theorem 24 clarifies the expressiveness of SFSADFs and SETADFs.

**Theorem 24.** *The following properties hold:*

1. $(\Sigma^\sigma_{SFSADF} \setminus \Delta'_\sigma) \subsetneq \Sigma^\sigma_{SETADF}$*, for* $\sigma \in \{prf, adm, com, grd\}$*,*
2. $\Sigma^\sigma_{SETADF} = (\Sigma^\sigma_{SFSADF} \setminus \Delta_\sigma)$*, for* $\sigma \in \{stb, mod\}$*.*

*Proof.* We show the two statements separately.

1) By Theorem 18 we have that $\Sigma^\sigma_{\text{SETADF}} = (\Sigma^\sigma_{\text{SFADF}} \setminus \Delta_\sigma)$, and by the definition of SFSADFs and $\Delta'_\sigma$ we have $\Sigma^\sigma_{\text{SFSADF}} \setminus \Delta'_\sigma \subseteq \Sigma^\sigma_{\text{SFADF}} \setminus \Delta_\sigma$. Combining these two statements we obtain $(\Sigma^\sigma_{\text{SFSADF}} \setminus \Delta'_\sigma) \subseteq \Sigma^\sigma_{\text{SETADF}}$, for $\sigma \in \{prf, adm, com, grd\}$. To complete the proof, let $\mathbb{V} = \{\{a \mapsto \mathbf{u}\}\}$. The interpretation-set $\mathbb{V}$ is $\sigma$-realizable in SETADFs. However, $\mathbb{V} \notin \Sigma^\sigma_{\text{SFSADF}}$. Thus, $\Sigma^\sigma_{\text{SETADF}} \not\subseteq (\Sigma^\sigma_{\text{SFSADF}} \setminus \Delta'_\sigma)$.

2) By Theorem 18 we have $\Sigma^\sigma_{\text{SETADF}} = (\Sigma^\sigma_{\text{SFADF}} \setminus \Delta_\sigma)$ and by [6]we have $\Sigma^\sigma_{\text{SFSADF}} = \Sigma^\sigma_{\text{SFADF}}$ for $\sigma \in \{stb, mod\}$. Combining these two results we obtain $\Sigma^\sigma_{\text{SETADF}} = \Sigma^\sigma_{\text{SFSADF}} \setminus \Delta_\sigma$ for $\sigma \in \{stb, mod\}$. □

The results of comparison of expressiveness of SETADFs, SFSADFs and SFADFs, for $\sigma \in \{adm, prf, stb, mod, grd, com\}$, are depicted in Figure 3. In both figures it is shown that the set $\Sigma^\sigma_{\text{SETADF}}$, depicted by vertical lines, is equal to the set $\bar{\Delta}_\sigma$, for all semantics. In addition, the expressiveness of SFSADFs is equal to SFADFs, for $\sigma \in \{stb, mod\}$. However, SFADFs are more expressive than SFSADFs, for $\sigma \in \{adm, prf, com, grd\}$. Further, some of the interpretation-sets of $\Delta_\sigma$ are not realizable in SFSADFs, for $\sigma \in \{adm, prf, com, grd\}$.

# References


[1] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes P. Wallner, and Stefan Woltran. Abstract dialectical frameworks revisited. In *Proc. IJCAI*, pages 803–809, 2013.

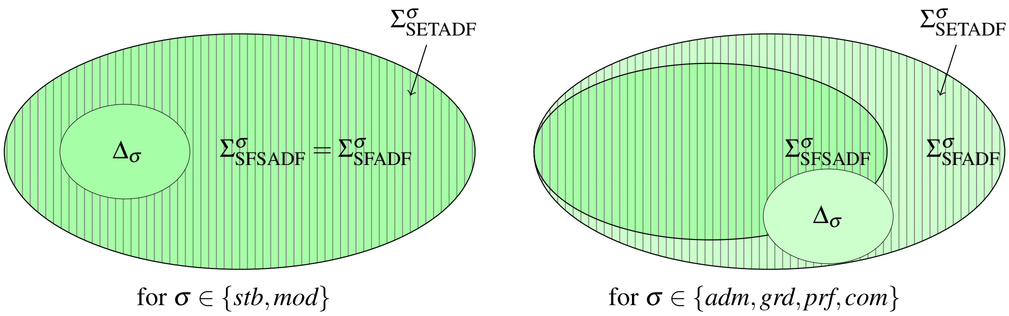



Figure 3: Expressiveness SETADFs, SFSADFs and SFADFs for $\sigma \in \{adm, prf, mod, stb, grd, com\}$


[2] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes P. Wallner, and Stefan Woltran. Abstract Dialectical Frameworks: An Overview. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 5. College Publications, February 2018.

[3] Gerhard Brewka, Sylwia Polberg, and Stefan Woltran. Generalizations of dung frameworks and their role in formal argumentation. *IEEE Intelligent Systems*, 29(1):30–38, 2014.

[4] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In *Proc. KR*, pages 102–111, 2010.

[5] Martin Diller, Atefeh Keshavarzi Zafarghandi, Thomas Linsbichler, and Stefan Woltran. Investigating subclasses of abstract dialectical frameworks. In Sanjay Modgil, Katarzyna Budzynska, and John Lawrence, editors, *Proceedings of COMMA 2018: Computational Models of Argument*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, pages 61–72. IOS Press, 2018.

[6] Martin Diller, Atefeh Keshavarzi Zafarghandi, Thomas Linsbichler, and Stefan Woltran. Investigating subclasses of abstract dialectical frameworks. *Argument & Computation*, 11(1), 2020.

[7] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–357, 1995.

[8] Wolfgang Dvorák, Jorge Fandinno, and Stefan Woltran. On the expressive power of collective attacks. *Argument & Computation*, 10(2):191–230, 2019.

[9] Wolfgang Dvořák, Jorge Fandinno, and Stefan Woltran. On the expressive power of collective attacks. In *COMMA*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, pages 49–60. IOS Press, 2018.

[10] Giorgos Flouris and Antonis Bikakis. A comprehensive study of argumentation frameworks with sets of attacking arguments. *International Journal of Approximate Reasoning*, 2019.

[11] Atefeh Keshavarzi Zafarghandi. Investigating subclasses of abstract dialectical frameworks. Master's thesis, TU Wien, 2017.

[12] Thomas Linsbichler, Jörg Pührer, and Hannes Strass. A uniform account of realizability in abstract argumentation. In *Proc. ECAI*, pages 252–260. IOS Press, 2016.

[13] Søren Holbech Nielsen and Simon Parsons. A generalization of dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In Nicolas Maudet, Simon Parsons, and Iyad Rahwan, editors, *Argumentation in Multi-Agent Systems*, pages 54–73, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[14] Sylwia Polberg. Understanding the abstract dialectical framework. In *Proc. JELIA*, volume 10021 of *Lecture Notes in Computer Science*, pages 430–446, 2016.

[15] Sylwia Polberg. *Developing the abstract dialectical framework*. PhD thesis, TU Wien, Institute of Information Systems, 2017.

[16] Jörg Pührer. Realizability of three-valued semantics for abstract dialectical frameworks. *Artif. Intell.*, 278, 2020.

[17] Johannes Peter Wallner. Structural constraints for dynamic operators in abstract argumentation. In *COMMA*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, pages 73–84. IOS Press, 2018.

[18] Johannes Peter Wallner. Structural constraints for dynamic operators in abstract argumentation. *Argument & Computation*, Pre-press(Pre-press):1–40, 2019.

[19] Bruno Yun, Srdjan Vesic, and Madalina Croitoru. Toward a more efficient generation of structured argumentation graphs. In Sanjay Modgil, Katarzyna Budzynska, and John Lawrence, editors, *Computational Models of Argument - Proceedings of COMMA 2018, Warsaw, Poland, 12-14 September 2018*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, pages 205–212. IOS Press, 2018.