FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Investigating Subclasses of Abstract Dialectical Frameworks

MASTERARBEIT

zur Erlangung des akademischen Grades

Master of Science

im Rahmen des Studiums

Computational Logic

eingereicht von

Atefeh Keshavarzi
Matrikelnummer 01607385

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.techn. Stefan Woltran
Mitwirkung: Dipl.-Ing. Dr.techn. Thomas Linsbichler
Martin Diller, MSc

Wien, 6. September 2017

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Investigating Subclasses of Abstract Dialectical Frameworks

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Computational Logic

by

Atefeh Keshavarzi
Registration Number 01607385

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Stefan Woltran
Assistance: Dipl.-Ing. Dr.techn. Thomas Linsbichler
Martin Diller, MSc

Vienna, 6th September, 2017

Erklärung zur Verfassung der Arbeit

Atefeh Keshavarzi
Address

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 6. September 2017

Acknowledgements

First of all, I would like to thank my supervisor Prof. Stefan Woltran for his enormous help and patience throughout this project. I am happy that I had the opportunity to work with Thomas Linsbichler and Martin Diller on this thesis and would like to thank them for this collaboration. I am grateful to my family for supporting me by assuming my responsibilities and providing me the opportunity to travel: my father Nouri, my brother Ali, and Maryam my aunt whose presence has helped us to cope with life after the departure of my mother. Most importantly, I would like to commemorate my mother Akram who taught me to be perseverant, patient, and hopeful in life. May her soul rest in piece.

Kurzfassung

Das Forschungsgebiet der Argumentation, inbesonders formale Modelle von Argumentation, wurde in letzter Zeit zu einem wichtigen Thema in der künstlichen Intelligenz. Dies ergibt sich durch die Verbindung zu - sowie diversen Anwendungen in - anderen Disziplinen wie Philosophie, Rechtswissenschaften, Logik, und Medizin, aber auch durch die thematische Nähe zu anderen Formlismen der KI, insbesonders aus dem Bereich der Wissensrepräsentation. Obwohl eine Vielzahl von Formalismen für Argumentation vorgeschlagen wurde, sticht ein Ansatz hervor, nämlich die von Dung eingeführten abstrakten Argumentation Frameworks (AFs). Ein AF ist einfach ein gerichteter Graph, wobei die Knoten Argumente repräsentieren, und die Kanten Konflikte. Diese Konflikte werden dann mittels Semantiken aufgelöst. Obwohl AFs sehr populär sind, stößt ihre Aussdrucksstärke in verschiedenen Anwendungen an ihre Grenzen. Daher finden sich in der Literatur zahlreiche Erweiterungen, wobei Abstract Dialectical Frameworks (ADFs) weit verbreitet sind. ADFs erlauben flexible Beziehungen zwischen Argumenten, welche mittles aussagenlogischer Formalen spezifiziert werden.

In dieser Arbeit wollen wir einige Lücken in der Forschung über ADFs schließen. So werden wir z.B. das Fundamentale Lemma von Dung auf ADFs erweitern. Weiters wollen wir untersuchen, ob sich gewisse Eigenschaften, die für spezielle Subklassen von AFs gelten, sich mittels geeigneter Definition von Subklassen auf ADFs ubertragen. Hierfür definieren wir verschiedene neue Klassen (symmetric ADFs, acyclic ADFs, attack symmetric ADFs, acyclic support ADFs, complete ADFs) und untersuchen deren Eigenschaften. Ein weiterer Aspekt dieser Arbeit sind Resultate zur Aussdrucksstärke dieser Klassen in bezug auf das Konzept der sogenannten Realisierbarkeit. Abschließend stellen wir eine Implementierung eines Generators für die genannten ADF Subklassen vor und untersuchen damit inwiefern sich existierende Systeme für ADFs in der Handhabung von Zyklen verhalten.

Abstract

Argumentation, and in particular computational models of argumentation, has recently become a main topic within artificial intelligence. This is not only because of its crucial importance and wide applications in other fields of science like philosophy, law, logic, and medicine but also because of its connection to other areas of AI, in specific, knowledge representation. Although there exists a wide variety of formalisms of argumentation, one popular, prominent and simple formalism stands out, namely abstract argumentation frameworks (AFs) first introduced by Dung. Intuitively, an AF is a directed graph in which nodes represent arguments and directed links represent conflicts between arguments. The conflicts between the arguments are resolved on the semantical level. Although AFs are very popular tools in argumentation because of their conceptual simplicity, they are not expressive enough to define different kind of relations. Several generalizations of AFs exist, in particular, abstract dialectical frameworks (ADFs), a powerful generalization of AFs, are widely studied. ADFs, first defined by Brewka and Woltran, are capable to express arbitrary relations between arguments with no need of defining a new type of relations and by assigning an acceptance condition to each argument in the form of a propositional formula.

In the current work we close some gaps in existing research on ADFs. More specifically, we investigate whether some main results carry over from AFs to ADFs. For instance, we reformulate Dung's Fundamental Lemma and we study under which conditions all semantics of an ADF coincide. We also study whether particular properties which are known to hold for certain subclasses can be extended to the world of ADFs by defining related subclasses of ADFs. To do so, we introduce several such classes (symmetric ADFs, acyclic ADFs, attack symmetric ADFs, acyclic support symmetric ADFs, complete ADFs) and investigate their properties. A central aspect of our work is comparing the expressivity of subclasses of AFs and ADFs from the perspective of realizability. At the end we introduce an implementation of a generator to produce such subclasses of ADFs. We use this generator in order to evaluate the effect of cycles on the performance of existing solvers for ADFs.

Contents

CHAPTER 1

Introduction

To investigate the importance of argumentation it is enough to mention that reasoning has been a specific topic in philosophy since the time of Aristotle. According to Leibniz, "the only way to rectify our reasonings is to make them as tangible as those of the Mathematicians, so that we can find our error at a glance, and when there are disputes among persons, we can simply say: Let us calculate [calculemus], without further ado, to see who is right". Argumentation and in particular computational models of argumentation has recently become a main topic within artificial intelligence (AI) [6, 9, 13, 38].

Based on the definition explained by Bench-Capon and Dunne [6] argumentation can be considered as, "concerned with how assertions are proposed, discussed, and resolved in the context of issues upon which several diverging opinions may be held." Argumentation is a method of negotiating beliefs among agents and an argument usually presents beliefs and reasonable justifications. The importance of understanding argumentation and its role in human reasoning in different fields like philosophy [44], law [7], politics [2, 12], decision support [3] to name just a few, motivated AI scientists to analyze the structure of arguments and to define different formalizations that deal with the situation when mutually conflicting arguments have to be taken into account. Moreover, recently argumentation has become a center of attention in AI because of its connection to other areas of AI, in specific, knowledge representation [8], game theory [37] and non-monotonic reasoning [24].

There are a number of ways to define an argument. One can consider an argument as a pair s.t. the first item in this pair is a minimal set of consistent formulas and the second item is a logical consequence of the first part. To explain an argument formally consider Δ to be a knowledge base which is a set of propositional formulas. An argument is a pair $\langle \phi, \alpha \rangle$ in which ϕ is a consistent subset of Δ s.t. ϕ is a model of α, $\phi \models \alpha$, and for all $\psi \subset \phi$, $\psi \not\models \alpha$. For instance, let $\Delta = \{s, w, r, s \rightarrow \neg r, w \rightarrow \neg s, r \rightarrow \neg w\}$ be a knowledge base in which s, w and r represent sunny, windy and rainy day, respectively.

Then, $\phi_1 = \{s, s \rightarrow \neg r\}$ is a consistent subset of Δ and $\phi_1 \models \neg r$ and none of the strict subsets of ϕ_1 is a model of $\neg r$. That is, $\langle \phi_1, \neg r \rangle$ is an argument of Δ. Another argument of Δ is $\langle \phi_2, \neg s \rangle$ s.t. $\phi_2 = \{w, w \rightarrow \neg s\}$. In addition, there is another argument $\langle \phi_3, \neg w \rangle$ in which $\phi_3 = \{r, r \rightarrow \neg w\}$. Then, there is a conflict between these two arguments. Formally, we say that there is an attack from $\langle \phi_i, \alpha_i \rangle$ to $\langle \phi_j, \alpha_j \rangle$ whenever there exists a $\mu \in \Delta$ s.t. $\alpha_i \models \neg \mu$ and $\mu \in \phi_j$. When there is an attack form $\langle \phi_i, \alpha_i \rangle$ to $\langle \phi_j, \alpha_j \rangle$ it is depicted in a diagram by an arrow from $\langle \phi_i, \alpha_i \rangle$ to $\langle \phi_j, \alpha_j \rangle$. The diagram depicted in Figure 1.1 show the attacks among the three arguments of Δ.

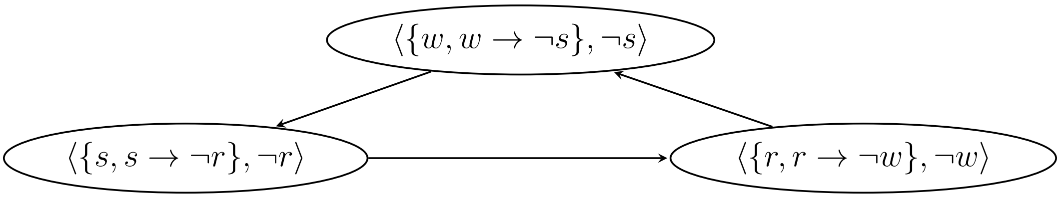

Figure 1.1: Conflicts among arguments of knowledge base $\Delta = \{s, w, r, s \rightarrow \neg r, w \rightarrow \neg s, r \rightarrow \neg w\}$

Argumentation frameworks (AFs for short) first introduced by Dung [24] provide a formal tool that abstracts from the internal structure (content) of arguments. It is mainly defined based on a set of arguments and a binary relation between arguments which represents conflicts (attacks). An AF corresponding to the above example which abstracts away from the content of arguments is depicted in Figure 1.2 in which, $a_i = \langle \phi_i, \alpha_i \rangle$.

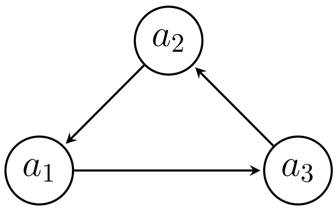

Figure 1.2: AF modeling the diagram in Figure 1.1

More formally, an AF is a pair (A, R) such that A is a set of arguments and $R \subseteq A \times A$ is a binary relation representing conflict (attack) between arguments. In AFs statements (called arguments) are formulated together with a relation (attack) between them and the conflicts between the arguments are resolved on the semantical level. Several ways of defining semantics of AFs are possible. One of them which is used in the current work is extension-based in which the idea is finding sets of arguments which are accepted jointly. Further semantics are defined in [4, 5, 20]. The concept of realizability is defined in [25] which is focusing on analyzing whether an extension-set is in correspondence to the set of outcomes of a semantics of an AF. Because of the crucial role of AFs in AI and the fact that there are many reasoning problems in argumentation frameworks, efficient solving techniques for reasoning tasks within AFs are needed. In [21], an overview of different methods for solving abstract argumentation tasks on AFs are given.

Although AFs are a very popular tool in argumentation because of their conceptual simplicity, they are not powerful enough because of several reasons. For instance, AFs defined based on single relation attack are not expressive enough to express different kind of relations among arguments, like support. Many studies focus on generalizing AFs by defining and adding a positive relation among arguments, named support relations. This generalization is called Bipolar AFs [14, 18, 22, 34]. Bipolar AFs have their own drawbacks. For instance, we need to define a new set of relations to define support relations. In addition, what if one argument a is not strong enough to attack b and another argument also is needed to jointly attack b. This kind of relation is very common, however AFs are not powerful enough to model them directly. There are several generalizations of AFs, see e.g. [18] for an overview. Currently *abstract dialectical frameworks*, ADFs for short, first introduced by Brewka and Woltran [16] and further refined in [17], are widely studied as a generalization of AFs. ADFs try to unify several generalizations of AFs. ADFs are powerful enough to express arbitrary relations between arguments with no need of defining a new type of links. An ADF is a tuple (S, L, C) in which S is a set of arguments, L is a set of links and C is a set of propositional formulas. The meaning of links in ADFs are very flexible: they can be support, attack (or both or neither). Each argument $s \in S$ is in correspondence with a propositional formula in C which is called acceptance condition of s denoted by φ_s.

It is shown in [39] that semantics of ADFs are proper generalizations of AFs. Semantics of ADFs are mainly defined based on three-valued interpretations which are similar to labelling-based semantics defined for AFs [20]. The concept of realizability in ADFs has been studied in [40, 41, 35]. In addition, it is shown in [17] that ADFs are more expressive than AFs. Various results on ADF complexity have been studied in [28, 42, 43]. While ADFs are a powerful generalization of AFs, this capability comes for ADFs by increasing one level up in the polynomial hierarchy compared to AFs. Given the power and role of ADFs in AI and the fact that there are many reasoning problems with high computational complexity, implementation methods are an important research topic as well. Solvers for ADFs can be classified based on the target-formalism: answer set programming (ASP) and Quantified Boolean Formula (QBFs). DIAMOND [26, 27] and YADF [19] are based on ASP and QADF [23] is based on QBFs. Solvers for ADFs are highly relevant because ADFs are able to capture many of the other generalizations of Dung's frameworks that exists [18, 33]. ADFs have themselves also received increasing attention as a representation formalism recently [36, 15, 1].

Dung's argumentation frameworks have been extensively investigated. For example it is shown in [24] that when the framework is well-founded, all semantics are equivalent, and in [22] it is illustrated that symmetric AFs are coherent and relatively grounded. To the best of our knowledge, it has not been investigated under which conditions these results carry over to ADFs. The main purpose of this thesis as follows.

- Reformulating and proving Dung's Fundamental lemma [24] for ADFs.

- Specifying a subclass of ADFs in which different semantics can collapse to the same set of interpretations.
- Clarifying whether symmetric ADFs are coherent and relatively grounded.
- Study properties of more fine-grained subclasses of ADFs, namely, attack symmetric ADFs, acyclic support symmetric ADFs and complete ADFs.
- Comparing the expressivity of different argumentation formalisms (studied in the current work) from the perspective of realizability.
- Providing a generator to produce acyclic ADFs, attack symmetric ADFs and acyclic support symmetric ADFs for a given undirected graph as input.
- Using this generator to illustrating the effect of cycles on the performance of solvers for ADFs.

This thesis is organized as follows. In Chapter 2 we recall some relevant background. In particular, we provide a short recap on AFs, ADFs and a well-known subclass of ADFs, BADFs. The presentation of this chapter is based on [24, 16, 17]. In addition, Dung's Fundamental lemma [24] is reformulated in this chapter. In Chapter 3 we show that some main results carry over from AFs to ADFs, for instance, the conditions under which semantics of ADFs are collapsing into a unique semantics are studied. In addition, it is investigated whether properties of symmetric AFs, namely coherency and relatively groundedness, carry over to ADFs. To illustrate under which conditions these properties of AFs hold for ADFs we introduce and study subclasses of ADFs, namely, attack symmetric ADFs, acyclic support symmetric ADFs and complete ADFs. In Chapter 4 the notion of realizability of an extension-set and an interpretation-set in argumentation formalisms and expressiveness of different formalisms are described along the lines of [25, 40, 41, 35]. Then, the expressiveness of formalisms defined in Chapter 3 are studied. In Chapter 5 we first describe the generator by which acyclic ADFs, attack symmetric ADFs and acyclic support symmetric ADFs are generated. Then, to illustrate the effect of cycles on the performance of solvers for ADFs we carried out some practical experiments. Finally, in Chapter 6 we will summarize and conclude the presented results and refer to the open questions we would like to address next.

CHAPTER 2

Background

In this chapter we briefly survey the notions of Argumentation Frameworks (AFs) and Abstract Dialectical Frameworks (ADFs) which are used in the rest of the work. In particular, in Section 2.1 we provide a short recap of AFs defined by Dung [24] which are modified in a way we use in our work. In Section 2.2 we introduce the syntax and semantics of ADFs as well as bipolar ADFs (BADFs), mainly based on [16, 17]. Moreover, in Section 2.2 we reformulate the Fundamental Lemma of AFs in the context of ADFs. The latter is a new result, which we nevertheless consider better placed in this chapter.

2.1 Argumentation Frameworks

Dung's argumentation frameworks (AFs for short) are a standard formalism of studying abstract argumentation, first defined in [24] and then refined by several authors. AFs nowadays become a central formalisms in artificial intelligence (AI) and it is widely used and studied among other argumentation firmalisms. It is a field that studies how arguments relate to each other regarding to directed conflicts which are named attack relations. AFs can be considered as a knowledge representation formalism, since it is used to represent the knowledge about the arguments and the relations between them. The basic definitions of AFs explained by [24] are described as follows:

Definition 1. An *argumentation framework* (AF) is a pair (A, R) such that A is a set of arguments and $R \subseteq A \times A$ is a binary relation representing attacks between arguments.

Remark 1. For each $a, b \in A$, $R(a, b)$ or $(a, b) \in R$ means that there is an attack from a to b.

In another word, an argumentation framework AF $F = (A, R)$ is a directed graph in which nodes represent arguments and links denote that one argument attacking another one. In Example 1 the graph corresponds to the given AF is depicted.

Example 1. Let $F = (\{a, b, c, d\}, \{(a, c), (c, c), (b, c), (d, c), (d, a), (c, d)\})$ be an AF. The graph depicted in Figure 2.1 is corresponding to AF F.

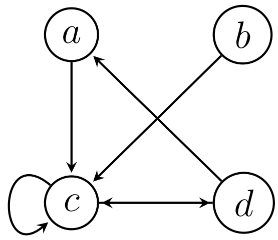

Figure 2.1: AF used in Example 1

Definition 2. Let $F = (A, R)$ be an AF, it is named finite argumentation framework when A is a finite set of arguments.

We say $S \subseteq A$ attacks $b \in A$ if there exists $a \in S$ such that $(a, b) \in R$. For instance, for a given AF $F = (\{a, b, c, d\}, \{(a, c), (c, c), (b, c), (d, c), (d, a), (c, d)\})$ in Example 1, $S = \{a, c\}$ attacks d, since $(c, d) \in R$.

Definition 3. Let $F = (A, R)$ be an AF. An argument $a \in A$ is *defended* by $S \subseteq A$ (or it is *acceptable* with respect to S) (in F) if for each argument $c \in A$: if $(c, a) \in R$ then there exists $b \in S$ such that $(b, c) \in R$.

For instance, in the AF of Example 1, argument a is defended by $S = \{a, c\}$. Since S defenses a by c against the attack of d.
The idea of argumentation is that whether an argument can argue successfully against attacking arguments. AF is concerned about ways to pick subsets of the arguments, *extensions*, that can all be accepted. The methods to clarify whether an argument is accepted in an AF is named semantics. Roughly speaking, in AFs a semantics is a function from the set of AFs to sets of sets arguments.

Definition 4. Let $\mathcal{F}_A$ denote the set of all AFs over A. An extension semantics is a function $\sigma : \mathcal{F}_A \to 2^{2^A}$. That is, for every AF $F = (A, R)$ we have $\sigma(F) \in 2^{2^A}$. The elements of $\sigma(F)$ are called extensions.

In the following we explain some of the properties and semantics of AFs based on [24]. An extension is a set of jointly acceptable arguments from a certain point of view. Semantics give rise to extensions, i.e. a semantics maps each AF to a set of extensions. Example 3 clarifies the following definitions by computing all semantics of the given AF.

Definition 5. Let $F = (A, R)$ be an AF. The set $S \subseteq A$ is a *conflict-free* set in F if there is no $a, b \in S$ such that $(a, b) \in R$. The set of all conflict-free sets is denoted by *cf*(F).

A naive extension of F is a maximal conflict-free set of F w.r.t. set inclusion which is defined as follows:

Definition 6. Let $F = (A, R)$ be an AF. A conflict-free set ($cf(F)$) of arguments $S \subseteq A$ is a *naive extension* of F if for each conflict-free set of arguments $T \subseteq A$ it is not the case that $S \subsetneq T$. The set of naive extensions is denoted by $nai(F)$.

Definition 7. Let $F = (A, R)$ be an AF. A set of arguments $S \subseteq A$ is *admissible* in F if S is a conflict-free set in F and for each $a \in S$, a is defended by S in F. The set of admissible sets in F is denoted by $adm(F)$.

A preferred extension of F is a maximal admissible set of F w.r.t. set inclusion.

Definition 8. Let $F = (A, R)$ be an argumentation framework. An admissible set of arguments S of F is a *preferred extension* of F if for each admissible set of arguments $T \subseteq A$ it is not the case that $S \subsetneq T$. The set of preferred extensions of F is denoted by $prf(F)$.

Definition 9. Let $F = (A, R)$ be an AF. The unique *grounded extension* of F is a set S which is an output of the following algorithm:

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

The unique grounded extension is denoted by $grd(F)$.

Definition 10. Let $F = (A, R)$ be an AF. An admissible set $S \subseteq A$ of F is a *complete extension* of F if each argument $a \in A$ defended by S in F is contained in S. That is, S contains all the arguments that are defended by S in F. The set of complete extensions of F is denoted by $com(F)$.

Theorem 1. *[24] In any argumentation framework $F = (A, R)$ the grounded extension of F is a subset-minimal complete extension of F.*

Definition 11. Let $F = (A, R)$ be an argumentation framework. A conflict-free set $S \subseteq A$ of F is a *stable extension* of F if for each $a \in A \setminus S$ there exists $b \in S$ such that $(b, a) \in R$. That is, a conflict-free set S is a stable extension iff S attacks all arguments which do not belong to S. The set of stable extensions of F is denoted by $stb(F)$.

Lemma 1. *Let $F = (A, R)$ be an argumentation framework. A set of arguments $S \subseteq A$ is a stable extension if and only if $S = \{a \in A| \quad a \text{ is not attacked by } S\}$.*

Lemma 2. *For any argumentation framework $F = (A, R)$ the following holds:*

- *Each stable extension of F is a preferred extension of F.*
- *Each preferred extension of F is a complete extension of F.*
- *Each complete extension of F is an admissible extension of F.*

Since in each AF $F = (A, R)$ empty set is an admissible extension, any AF possesses at least a σ-extension, $\sigma \in \{adm, prf, com, nai, grd\}$, however, there is no guarantee of possessing of stable extension. That is, the set of all stable extensions could be empty. This fact is illustrated by the following example.

Example 2. Let $F = (\{a, b, c, d\}, \{(a, b), (b, c), (d, c), (d, d)\})$ be an AF, depicted in Figure 2.2 . To compute the stable extensions of F first we compute the set of conflict-free sets: $cf(F) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, c\}\}$. None of the sets of $cf(F)$ attacks all arguments which do not belongs to it. Therefore, $stb(F) = \emptyset$.

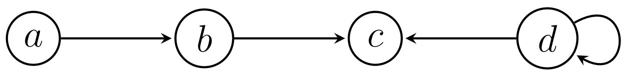

Figure 2.2: AF used in Example 2

By Theorem 1 each AF contains a unique grounded extension. However, the set of σ-extensions, for $\sigma \in \{adm, prf, com, stb, nai\}$, could be a multiple set. To illustrate the definitions explained in this subsection all the extensions studied in this work are computed for the AF in Example 3.

Example 3. Consider an AF $F = (\{a, b, c, d, e\}, \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\})$ depicted in Figure 2.3.

- The set of conflict-free extensions of F is: $cf(F) = \{\{a, c\}, \{a, d\}, \{b, d\}, \{a\}, \{b\}, \{c\}, \{d\}, \emptyset\}$,
- The set of naive extensions of F is: $nai(F) = \{\{a, c\}, \{a, d\}, \{b, d\}\}$,
- The set of admissible extensions of F is: $adm(F) = \{\{a, c\}, \{a, d\}, \{a\}, \{c\}, \{d\}, \emptyset\}$,
- The set of preferred extensions of F is: $prf(F) = \{\{a, c\}, \{a, d\}\}$,
- The set of complete extensions of F is: $com(F) = \{\{a, c\}, \{a, d\}, \{a\}\}$,
- The unique grounded extension of F is: $grd(F) = \{a\}$,
- The stable extension of F is: $stb(F) = \{\{a, d\}\}$.

One of the crucial lemma in AFs proven by Dung in [24] is the Fundamental Lemma. To the best of our knowledge the Fundamental Lemma has not been reformulated in ADFs.

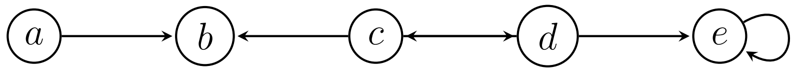

Figure 2.3: AF used in Example 3

Lemma 3. *[24] Let $F = (A, R)$ be an AF and $S \subseteq A$ be an admissible set of arguments for F, and a and a' be arguments which are acceptable w.r.t. A in F. Then,*

1. *$A' = A \cup \{a\}$ is admissible for F, and*
2. *a' is acceptable for F w.r.t. A'.*

2.2 Abstract Dialectical Frameworks

In the previous section we studied the syntax and semantics of AFs which are mainly based on a single relation among arguments, namely attack. Although AFs are popular and well-studied in argumentation, it is not powerful enough to express technical arguments. To over come to this impotency of AFs as a representation formalism there exists quite a number of generalizations of AFs, for an overview see e.g. [18]. Abstract dialectical frameworks (ADFs for short) introduced first by Brewka and Woltran in [16] can be considered as one of the powerful generalizations of AFs in which both arguments and links are abstract. ADFs can be considered as a knowledge representation formalism which is used to express knowledge about argument and relations between them. While in AFs all relations are attacks in ADFs the meaning of relations in ADFs are flexible. In ADFs each argument is associated with a propositional formula over its parents arguments which is called the acceptance condition of an argument. That is, in ADFs when to accept each argument is clarified by acceptance condition explicitly. An ADF is a tuple (S, L, C) in which S is a set of arguments, L is a set of links and C is a set of propositional formulas. However, in ADFs the meaning of relations between arguments can be support or attack (both or neither). We will explain the type of links at the end of this section. Various results on ADF semantics and complexity have been studied in [39, 43]. ADFs is defined formally as follows:

Definition 12. [16] An *abstract dialectical framework* (ADF) is a tuple $D = (S, L, C)$ where,

- S is a set of nodes (argument, statement, positions),
- $L \subseteq S \times S$ is a set of links,
- $C = \{\varphi_s\}_{s \in S}$ is a set of propositional formulas (acceptance conditions).

Definition 13. (Finite ADF)
A finite ADF is an abstract dialectic framework $D = (S, L, C)$ in which S is a finite set.

To visualize the structure of ADFs we use a directed graph such that nodes represent arguments and links represent relations among arguments. The meaning of relations in ADFs is very flexible and the nodes' status only depends on their parent's status. Parents of a node s is the set of nodes with a direct link into s and are denoted by par(s), i.e. $par(s) = \{a \in S \mid (a, s) \in L\}$. Each node s has an acceptance condition φ_s in which conditions under which s is accepted are explicitly defined. In another word, acceptance condition φ_s is a propositional formula such that an argument a occurs in φ_s if and only if $a \in par(s)$. Since acceptance condition φ_s specifies the parents of s implicitly, there is no need to give the links in ADFs explicitly. Therefore, one can define an ADF as a tuple $D = (S, C)$ where S and C is the same as above and $(a, s) \in L$ if and only if $a \in par(s)$.

Remark 2. From now on if not explicitly stated we assume that all acceptance conditions are written in conjunctive normal form, CNF.

Example 4. Let $D = (\{a, b, c, d, e\}, \{\varphi_a : \top, \varphi_b : b, \varphi_c : a \wedge \neg b, \varphi_d : \neg b, \varphi_e : \bot\})$ be an ADF in which each acceptance condition is defined by a propositional formula, depicted in Figure 2.4. Intuitively φ_a states that a should always be accepted, φ_b says its acceptance depends on b, the acceptance condition of c depends on acceptance of a and b, φ_d is defined by negation of b and φ_e states that e should always be rejected.
The set of parents of c is $\{a, b\}$, a and e does not have any parent, b is a parent of b and d. It is clear that there is no need to define the set of links among arguments explicitly. For example $\varphi_c : a \wedge \neg b$ implicitly says that the correspondence graph of D contains links (a, c) and (b, c).

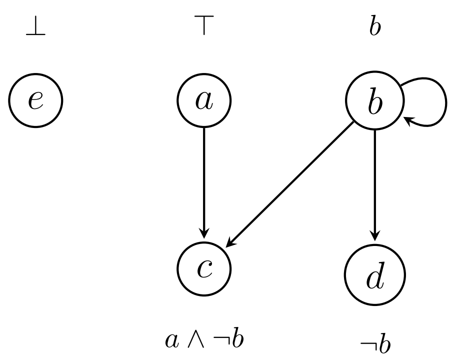

Figure 2.4: ADF used in Example 4

Definition 14. Let $D = (S, L, C)$ be an ADF. An argument $s \in S$ without any incoming and outgoing link is called *isolated argument*.

Definition 15. Let $D = (S, L, C)$ be an ADF. An argument $s \in S$ without any parents is named *initial argument*.

Acceptance condition of isolated and initial argument is either $\top$ or $\bot$. In Example 4, e is an isolated argument and a is an initial argument.

2.2.1 Semantics of ADFs

To define and compute the semantics of ADFs the concept of operation-based semantics is defined and used in [39, 17]. The mandatory definitions for computing semantics of ADFs are explained as follows:

Definition 16. A *three-valued interpretation* v is a function from a set of arguments S to $\{t, f, u\}$, $v : S \to \{t, f, u\}$. An interpretation is called *two-valued* if $v : S \to \{t, f\}$. The intuition of t, f, and u is that an argument is true, false and undecided, respectively.

Definition 17. A three-valued interpretation is named *trivial* if assigns all arguments to u. Otherwise, it is named non-trivial. The trivial interpretation is denoted by v_u.

Let $\mathcal{V}$ (resp. $\mathcal{V}_2$) be the set of all three-valued interpretations (resp. two-valued interpretations) of S, $\mathcal{V} = \{v : S \mapsto \{t, f, u\}\}$ (resp. $\mathcal{V}_2 = \{v : S \mapsto \{t, f\}\}$). Interpretations can be ordered by the information ordering $\leq_i$. This ordering assigns a greater information content to the classical truth value $\{t, f\}$ than to u, that is, $u <_i t$ and $u <_i f$, and $\leq_i$ is the reflexive, transitive closure of $<_i$. The partially ordered set $(\{t, f, u\}, \leq_i)$ forms a meet-semilattice with the meet operator $\sqcap_i$, such that $t \sqcap_i t = t$, $f \sqcap_i f = f$ and returns u otherwise. In addition $(\mathcal{V}, \leq_i)$ forms a complete meet-semilattice and its meet operator given by $(v_1 \sqcap_i v_2)(s) = v_1(s) \sqcap_i v_2(s)$, such that;

$$v_1 \leq_i v_2 \; if \; and \; only \; if \; \forall s \in S : \; v_1(s) \leq_i v_2(s).$$

The least element of this semilattice is the trivial interpretation $v_u : S \to \{u\}$ mapping all arguments to undecided. v_2 is called an extension of v_1 if and only if $v_1 \leq_i v_2$.
Let v be an arbitrary three-valued interpretation. $[v]_2 = \{w \in \mathcal{V}_2 \mid v \leq_i w\}$ is the set of all two-valued interpretations which have more information than v. That is, a single three-valued interpretation v serves to approximate a set of two-valued interpretations with more information than v with respect to $\leq_i$. In addition, v^x is used to denote the set of arguments assigned to $x \in \{t, f\}$ under v.

Definition 18. Let $\mathcal{V}$ be the set of interpretations and v_1 and v_2 be two interpretation sets of $\mathcal{V}$. v_1 and v_2 are called incomparable whenever neither $v_1 \leq_i v_2$ nor $v_2 \leq_i v_1$. This is denoted by $v_1 \not\sim v_2$.

Example 5. Consider the three-valued interpretation $v = \{a \mapsto t, b \mapsto u, c \mapsto f, d \mapsto u, e \mapsto u\}$. v has 27 different extensions: some of them are incomparable. For instance, $v_1 = \{a \mapsto t, b \mapsto t, c \mapsto f, d \mapsto u, e \mapsto u\}$ and $v_2 = \{a \mapsto t, b \mapsto f, c \mapsto f, d \mapsto u, e \mapsto u\}$ are incomparable. However, $v_3 = \{a \mapsto t, b \mapsto t, c \mapsto f, d \mapsto t, e \mapsto u\}$ is an extension of v_1 i.e. $v_1 \leq_i v_3$. $[v]_2$ contains 16 interpretations; one of them is: $v_4 = \{a \mapsto t, b \mapsto t, c \mapsto f, d \mapsto f, e \mapsto t\}$.
Let $w = \{a \mapsto f, b \mapsto t, c \mapsto f, d \mapsto u, e \mapsto f\}$ be another three-valued interpretation. The meet of v and w is: $v \sqcap_i w = \{a \mapsto u, b \mapsto u, c \mapsto f, d \mapsto u, e \mapsto u\}$.
In addition, $v^t = \{a\}$ and $v^f = \{c\}$.

The operator Γ_D which is defined in [16] transforms three-valued interpretations into others, $\Gamma_D : \mathcal{V} \to \mathcal{V}$.

Definition 19. Let $D = (S, L, C)$ be an ADF and v be a three-valued interpretation and φ_s be an acceptance condition of s. The revised interpretation $\Gamma_D(v) : S \to \{t, f, u\}$ is given by:

$$s \mapsto \sqcap_i \{w(\varphi_s) \mid w \in [v]_2\}$$

in which $w(\varphi_s)$ is obtained by evaluating φ_s with w.

The operator takes a three-valued interpretation v as an input and returns a three-valued interpretation $\Gamma_D(v)$. That is, we can compute the truth value of an argument s under interpretation v with the help of computing the meets of the evaluations of φ_s under all completions of v. If v is two-valued then $[v]_2 = \{v\}$. Therefore, in this case $\Gamma_D(v)(s) = v(\varphi_s)$. That is, when v is a two-valued interpretation $\Gamma_D(v)(s)$ is obtained by evaluating φ_s with v.
There is an alternative way of introducing the operator Γ_D described as the follows: Given an interpretation $v : S \to \{t, f, u\}$ and ADF $D = (S, L, C)$. Let φ_s be a propositional formula (acceptance condition) corresponding to s. Then the operator $\Gamma_D(v) : S \to \{t, f, u\}$ yields a new interpretation as follows:

$$s \mapsto \begin{cases} \text{t} & \text{if } \varphi_s^v \text{ is } \text{ irrefutable ,} \\ \text{f} & \text{if } \varphi_s^v \text{ is } \text{ unsatisfiable,} \\ \text{u} & \text{otherwise} \end{cases}$$

Such that the partial valuation of φ_s by v is;

$$\varphi_s^v = \varphi_s[p/\top \;:\; v(p) = t][p/\bot \;:\; v(p) = f]$$

where p is an argument occurring in φ_s.
A formula is irrefutable when it is satisfied under all two-valued interpretations, that is, the formula is a tautology. The definition of the operator is clarified via the following example:

Example 6. Consider ADF $D = (\{a, b, c, d, \}, \{\varphi_a : \top, \varphi_b : \bot, \varphi_c : \neg a \wedge b, \varphi_d : (a \wedge c) \vee b\})$ depicted in Figure 2.5 and three-valued interpretation $v = \{a \mapsto t, b \mapsto f, c \mapsto u, d \mapsto u\}$.

- First we compute revision of v with the first definition given for the operator Γ_D: Since $\varphi_a : \top$, for each $w \in [v]_2$, $w(\varphi_a) = t$. Hence, $\sqcap_i \{w(\varphi_a) \mid w \in [v]_2\} = t$. That is, $\Gamma_D(v)(a) = t$. With the same method it is easy to show that $\Gamma_D(v)(b) = f$. Since in all $w \in [v]_2$, a is assigned to t and b is assigned to f, $w(\varphi_c) = f$ for all $w \in [v]_2$. Therefore, $\Gamma_D(v)(c) = f$.
  To compute $\Gamma_D(v)(d)$ consider w_1 is a completion of v in which c is assigned to true and w_2 is a completion of v in which c is assigned to false, then $w_1(\varphi_d) = t$ and $w_2(\varphi_d) = f$. Hence, $w_1(\varphi_d) \sqcap_i w_2(\varphi_d) = t \sqcap_i f = u$. That is, $\Gamma_D(v)(d) \mapsto u$.Therefore, an operator Γ_D transforms three-valued interpretation $v = \{a \mapsto t, b \mapsto f, c \mapsto u, d \mapsto u\}$ in to $v' = \{a \mapsto t, b \mapsto f, c \mapsto f, d \mapsto u\}$.

- Then we show that applying second definition of Γ_D on v leads the same interpretation v'.
  Partial evaluation of φ_d with v takes the two-valued part of v and replaces the evaluated variables by their truth value. a, b and c are arguments that occur in φ_d and $v(a) = t$, $v(b) = f$ and $v(c) = u$. Therefore, to compute φ_d^v we replace a with $\top$, b with $\bot$ and argument c will remain in φ_d. $\varphi_d^v = (\top \wedge c) \vee \bot \equiv c$. φ_s^v in neither irrefutable nor unsatisfiable, hence, $\Gamma_D(v)(d) = u$. With the same way, $\varphi_c^v = (\neg\top) \wedge \bot \equiv \bot$, that is, φ_c^v is unsatisfiable. Then by the definition $\Gamma_D(v)(c) = f$. That is, $v' = \{a \mapsto t, b \mapsto f, c \mapsto f, d \mapsto u\}$.

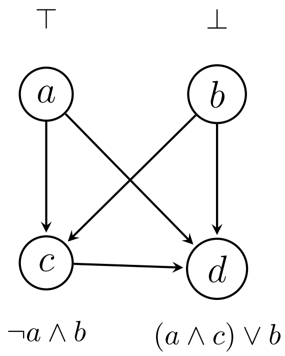

Figure 2.5: ADF used in Example 6

The fixed point of Γ_D is an interpretation that cannot be updated anymore, that is, there is no way to add a truth value to any arguments. It can be easily checked that Γ_D in Example 6 has only one fixed point $v = \{a \mapsto t, b \mapsto f, c \mapsto f, d \mapsto f\}$.
In the following, the notation $\varphi_s^v \equiv \top$ means φ_s^v is irrefutable and $\varphi_s^v \equiv \bot$ means φ_s^v is unsatisfiable.
$v|_x^s$ is a function on a three-valued interpretation v and assigns x to s, such that $x \in \{t, f\}$ and it is equal to v on all other arguments.

Example 7. Assume an ADF $D = (\{a, b, c\}, \{\varphi_a : \top, \varphi_b : a, \varphi_c : \neg a\})$ and a three-valued interpretation $v = \{a \mapsto t, b \mapsto u, c \mapsto u\}$. By the definition $v|_f^b$ is $\{a \mapsto t, b \mapsto f, c \mapsto u\}$ and $v|_t^c$ is $\{a \mapsto t, b \mapsto u, c \mapsto t\}$. However, neither $v|_f^b$ nor $v|_t^b$ is obtainable by applying the operator Γ_D on v. Since, $\Gamma_D(v) = \{a \mapsto t, b \mapsto t, c \mapsto f\}$

The concept of acceptability is defined for AFs and it is used in the Fundamental Lemma [24]. In the following we define acceptability and deniablity of an argument s w.r.t. an interpretation v to reformulate the lemma in ADFs.

Definition 20. Let $D = (S, L, C)$ be an ADF and v be an interpretation on S.

- An argument $s \in S$ is named *acceptable* w.r.t. v if φ_s^v is irrefutable,
- An argument $s \in S$ is called *deniable* w.r.t. v if φ_s^v is unsatisfiable.

In Example 8 the acceptability and deniability of arguments of the given ADF are illustrated with respect to the given interpretation.

Example 8. Let $D = (\{a, b, c, d, e\}, \{\varphi_a : \neg d, \varphi_b : \top, \varphi_c : \neg a \wedge b, \varphi_d : \neg b, \varphi_e : \neg a\})$ and $v = \{a \mapsto t, b \mapsto u, c \mapsto u, d \mapsto u, e \mapsto u\}$.

- By the definition $\varphi_a^v = \neg d$. Therefore, a is neither acceptable nor deniable w.r.t. v.
- $\varphi_b^v \equiv \top$ then b is acceptable w.r.t. v. In addition, b is acceptable under all interpretations.
- $\varphi_c^v \equiv \neg\bot \wedge \top \equiv \top$, that is, c is acceptable w.r.t. v.
- $\varphi_d^v = \neg b$ that is, d is neither acceptable nor deniable w.r.t. v.
- $\varphi_e^v \equiv \neg\top \equiv \bot$, that is, e is deniable w.r.t. v.

Since ADFs are defined basically based on arguments and their acceptance conditions, looking for the truth value of each argument is a common issue. The semantics of ADFs are the criteria by which the acceptability, deniability and even undecidability of an argument is clarified. Formally, let $\mathcal{F}_S$ denotes the set of all ADFs over a S. A semantics is a function form $\mathcal{F}_S$ to the set of interpretation sets, i.e. $\sigma : \mathcal{F}_S \mapsto 2^{\mathcal{V}}$. More specific; semantics of an ADF $D = (S, L, C)$ is $\sigma(D) \subseteq \mathcal{V}$. Clearly, the elements of $\sigma(D)$ are interpretations. As mentioned before the operator Γ_D defined by Brewka and Woltran [16], Brewka et al. [17] is used in ADFs to transform three-valued interpretations to others. Operator Γ_D can be used to define semantics of ADFs as follows.

Definition 21. [16, 17], Let $D = (S, L, D)$ be an ADF and $v : S \mapsto \{t, f, u\}$ be an interpretation;

- v is an *admissible interpretation* for D if $v \leq_i \Gamma_D(v)$; the set of all admissible interpretations for D is denoted by $adm(D)$.
- v is a *preferred interpretation* for D if v is $\leq_i$-maximal admissible; the set of all preferred interpretation for D is denoted by $prf(D)$.
- v is a *complete interpretation* for D if $v = \Gamma_D(v)$; the set of all complete interpretations for D is denoted by $com(D)$.
- v is a grounded interpretation for D if and only if v is the $\leq_i$-least fixed point of $\Gamma_D(v)$. v is called non-trivial grounded interpretation iff v does not take u to all arguments. The unique grounded interpretation for D is denoted by $grd(D)$

- v is a two-valued model for D if and only if v assigns all arguments to either true or false and $v = \Gamma_D(v)$; the set of all two-valued models (or models) for D is denoted by $mod(D)$.

- v is conflict-free for D iff for each $s \in S$ we have; $v(s) = t$ implies φ_s^v is satisfiable and $v(s) = f$ implies φ_s^v is unsatisfiable. The set of all conflict-free interpretations for D is denoted by $cf(D)$.

Intuitively, v is an admissible interpretation if the truth assignment of its arguments can be justified. For example, each member of v^t must be valid no matter how the undecided arguments of v are interpreted. That is, for an argument a in v^t(resp. v^f), φ_a must evaluate to true (resp. false) in all completions of v. In other words, v is an admissible if for all $s \in S$, $v(s) = t$ implies φ_s^v is irrefutable and $v(s) = f$ implies φ_s^v is unsatisfiable. Hence, the trivial interpretation, v_u, is an admissible interpretation in an arbitrary ADF. Clearly, conflict-freeness is a weaker version of admissibility, in which the truth assignment of an argument is justified by satisfiability instead of irrefutably. That is, in contrast to admissibility for conflict-freeness it is enough to check whether each argument of v^t is satisfiable. The difference of these two interpretations is clarified in Example 14.

Example 9. Let $D = (\{a, b, c\}, \{\varphi_a : \neg c, \varphi_b : \neg a, \varphi_c : \neg b\})$ be the ADF depicted in Figure 2.6. D contains ten conflict-free interpretations, but only one of them is admissible, v_u. That is the only admissible interpretation of D is $v_u = \{a \mapsto u, b \mapsto u, c \mapsto u\}$. To check whether $v_1 = \{a \mapsto t,\ b \mapsto f,\ c \mapsto u\}$ is a conflict-free interpretation we show that $\varphi_a^{v_1}$ is satisfiable and $\varphi_b^{v_1}$ is unsatisfiable. By evaluating $\varphi_b^{v_1} = \varphi_b[a/\top \ : \ v_1(a) = t]$, $\varphi_b^{v_1} \equiv \bot$. That is, $\varphi_b^{v_1}$ is unsatisfiable.
To show that $\varphi_a^{v_1} = \neg c$ is satisfiable it is enough to consider v_2 a completion of v_1 in which c assigns to f. Then, $\varphi_a^{v_2} \equiv \neg\bot \equiv \top$. That is, $\varphi_a^{v_1}$ is satisfiable.To show that v_1 is not an admissible interpretation, we show that $\varphi_a^{v_1}$ is not irrefutable. Then, it is enough to assume v_3 a completion of v_1 in which c assigns to t. That is, $\varphi_a^{v_3} \equiv \neg\top \equiv \bot$. That is, $\varphi_a^{v_1}$ is not irrefutable. Thus, v_1 is not an admissible interpretation.

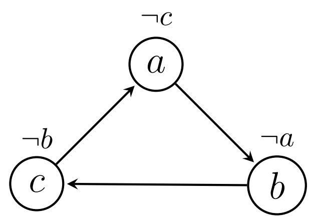

Figure 2.6: ADF used in Example 14

Next, we reformulate Dung' s Fundamental Lemma [24] for ADFs.

Lemma 4. ***Fundamental Lemma*** *Assuming v is an admissible interpretation of ADF D, and a and a' are arguments which are acceptable (resp. deniable) with respect to v. Then,*

1. $v' = v|_t^a$ *(resp.* $v' = v|_f^a$*) is admissible, and*
2. a' *is acceptable (resp. deniable) with respect to* v'*.*

Proof. Let v be an admissible interpretation of D and a be an argument acceptable (resp. deniable) w.r.t. v and $v' = v|_t^a$ (resp. $v' = v|_f^a$). First, we illustrate that $v \leq_i v'$.

> Assume that a is acceptable w.r.t. v we claim that $v(a)$ cannot be f. Suppose to the contrary that $v(a) = f$. Since v is an admissible interpretation, $v \leq_i \Gamma_D(v)$ implement that $\Gamma_D(v)(a) = f$. By the definition, $\Gamma_D(v)(a) = f$ if φ_a^v is unsatisfiable. This is a contradiction with our assumption that a is acceptable w.r.t. v. Therefore, $v(a)$ could be either t or u. By the definition $v'(a) = t$. Thus, $v \leq_i v'$ whenever a is acceptable w.r.t. v.
>
> Suppose that a is deniable w.r.t. v we show that $v(a) \neq t$. Toward a contradiction assume that $v(a) = t$. Again since v is an admissible interpretation, $v \leq_i \Gamma_D(v)$ enforce that $\Gamma_D(v)(a) = t$. By the definition of the operator Γ_D, $\Gamma_D(v)(a) = t$ if φ_a^v is irrefutable. This is a contradiction with the assumption that a is deniable w.r.t. v. Then, $v_a \in \{u, f\}$. By the definition $v'(a) = f$. Thus, $v \leq_i v'$ whenever a is deniable w.r.t. v.

Hence, $v \leq_i v'$. In the following we prove two parts of lemma separately.

1. By the definition of admissible interpretation we need to show that $v' \leq_i \Gamma_D(v')$, that is, for all $s \in S$, $v'(s) \leq_i \Gamma_D(v')(s)$. Then, it is enough to show that $\forall s \in S$: $v'(s) \in \{t, f\} \Rightarrow v'(s) = \Gamma_D(v')(s)$. Let s be an argument such that $v'(s) = t$. There are two cases either $s = a$ or $s \neq a$.

   a) Assuming $s \neq a$ by the definition of v', $v(s) = t$. Since v is an admissible interpretation $\Gamma_D(v)(s) = \sqcap_i\{w(\varphi_s) \mid w \in [v]_2\} = t$.

   b) Suppose $s = a$. By our assumption a is acceptable w.r.t. v, that is, φ_s^v is irrefutable. That is, $\sqcap_i\{w(\varphi_s) \mid w \in [v]_2\} = t$.

   By our claim $v \leq_i v'$, then $[v']_2 \subseteq [v]_2$. Therefore, $\sqcap_i\{w(\varphi_s) \mid w \in [v']_2\} = t$. That is, in both cases $\Gamma_D(v')(s) = t$. Let s be an argument such that $v'(s) = f$. With the same proof method we can show that $\Gamma_D(v')(s) = f$. Therefore, $v' \leq_i \Gamma_D(v')$, that is, v' is an admissible interpretation.

2. Let a' be acceptable w.r.t. v, we show that a' is acceptable w.r.t. v'. By our assumption a' is acceptable w.r.t. v, hence $\sqcap_i\{w(\varphi_{a'}) \mid w \in [v]_2\} = t$. Since $v \leq_i v'$, $[v']_2 \subseteq [v]_2$. Therefore, $\Gamma_D(v')(a') = \sqcap_i\{w(\varphi_{a'}) \mid w \in [v']_2\} = t$. That is, a' is acceptable w.r.t. v'. The proof method when a' is deniable w.r.t. v is exactly the same.

□

To define the stable model semantics for ADFs we define a method, first proposed in [16] similar to the one used in logic programming. Suppose $D = (S, L, C)$ is an ADF and v is a two-valued model. We are eager to know whether v is a stable model. To compute the reduction of ADFs, first, we eliminate all nodes and corresponding links from D which are false under the two-valued model v. Then, we replace eliminated nodes by $\perp$ in all acceptance conditions. We check whether nodes that are t in v coincide with those that are t in the grounded interpretation of the reduced ADF. That is, we must find a constructive proof for all arguments v takes to be true. This is the reason why we compute the grounded interpretation of the reduced ADFs. The formal method of computing the reduced ADFs D^v and checking whether v is a stable model is defined as the follows:

Definition 22. Let $D = (S, L, C)$ be an ADF and v be a two-valued model of D. Define the reduced ADF $D^v = (S^v, L^v, C^v)$ where,

- $S^v = v^t$,
- $L^v = L \cap (S^v \times S^v)$,
- $C^v = \{\varphi_s[p/\perp \ : \ v(p) = f]\}_{s \in S^v}$.

Denote by w the unique grounded interpretation of D^v. The two-valued model v of D is a stable model of D if and only if $v^t = w^t$.

We illustrate the semantics defined in this section by the following example:

Example 10. Consider the ADF explained in Example 4, $D = (\{a, b, c, d, e\}, \{\varphi_a : \top, \varphi_b : b, \varphi_c : a \wedge \neg b, \varphi_d : \neg b, \varphi_e : \perp\})$. Let $v = \{a \mapsto t, b \mapsto f, c \mapsto u, d \mapsto u, e \mapsto u\}$.

We show that v is an admissible interpretation but not a complete interpretation. D possesses 32 different admissible interpretations. To show that v is an admissible interpretation we have to show $v \leq_i \Gamma_D(v)$. It is enough to check whether $\Gamma_D(v)(a) = t$ and $\Gamma_D(v)(b) = f$.
Since $\varphi_a \equiv \top$, φ_a^v is irrefutable w.r.t. all interpretations v. That is $\Gamma_D(v)(a) = t$. Acceptance condition of b is defined based on itself, $\varphi_b = b$. That is, if b is assigned to false by an interpretation v then $\varphi_b^v = \varphi_b[b/\perp \ : \ v(b) = f] \equiv \perp$. Since

b is assigned to f by v, $\varphi_b^v \equiv \bot$, that is, φ_b is unsatisfied under v. Therefore, $\Gamma_D(v)(b) = f$. That is, $v(b) = \Gamma_D(v)(b)$. Then v is an admissible interpretation. Applying $\Gamma_D(v)$ on the remaining arguments, c, d, and e, changes their truth assignment. It can be computed that $\Gamma_D(v)(c) = t$, $\Gamma_D(v)(d) = t$ and $\Gamma_D(v)(e) = f$. Therefore, $v \neq \Gamma_D(v)$. That is, v is not a complete interpretation.

$com(D) = \{\{a \mapsto t, b \mapsto u, c \mapsto u, d \mapsto u, e \mapsto f\}, \{a \mapsto t, b \mapsto t, c \mapsto f, d \mapsto f, e \mapsto f\}, \{a \mapsto t, b \mapsto f, c \mapsto t, d \mapsto t, e \mapsto f\}\}$. The first interpretation is the $\leq_i$-least fixed point of $\gamma_D(v)$ hence it is the grounded interpretation. The two latter interpretations are two-valued and preferred interpretations.

We illustrate whether the two-valued models are stable. Let $v_1 = \{a \mapsto t, b \mapsto t, c \mapsto f, d \mapsto f, e \mapsto f\}$ and $v_2 = \{a \mapsto t, b \mapsto f, c \mapsto t, d \mapsto t, e \mapsto f\}$. The reduct of v_1 depicted in Figure 2.7. The grounded interpretation of D^{v_1} is $w_1\{a \mapsto t, b \mapsto u\}$. While, $w_1(b) = u$, b is assigned to t by v_1. Hence, v_1 is not a stable interpretation.

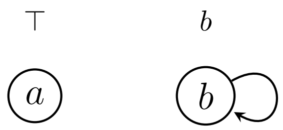

Figure 2.7: Reduct D^{v_1}

The reduct of v_2 is depicted in Figure 2.8. The grounded interpretation of D^{v_2} is $w_2 = \{a \mapsto t, c \mapsto t, d \mapsto t\}$. Since the arguments that are assigned to t in v_2 coincide with those that are t in the grounded interpretation of the reduced ADF, v_2 is a stable model. Hence, the only stable model of D is $\{a \mapsto t, b \mapsto t, c \mapsto f, d \mapsto t, e \mapsto f\}$.

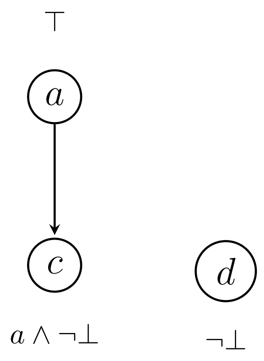

Figure 2.8: Reduct D^{v_2}

The following theorem [17] shows that the relationship among semantics of AFs defined on Dung [24] carry over to ADFs.

Theorem 2. *Let D be an ADF.*

- *Each stable model of D is a two-valued model of D;*
- *each two-valued model of D is a preferred interpretation;*
- *each preferred interpretation of D is a complete one;*
- *each complete interpretation of D is an admissible interpretation;*
- *The grounded interpretation of D is a complete interpretation.*

It is stated in [16] that any ADF possesses at least a preferred interpretation and the grounded interpretation, however, there is no guarantee of possessing a stable or two-valued model. The fact that $prf(D)$ is non-empty for each ADF D is concluded directly by the fact that the trivial interpretation is admissible and by the definition of preferred interpretation which says each preferred interpretation v is $\leq_i$-maximal admissible. Again the fact that the trivial interpretation is admissible (together with the fact that the set of all three-valued interpretations of an ADF D with the meet operator $\sqcap_i$ forms a complete meet-semilattice) implements the existence of the grounded interpretation which is a $\leq_i$-least fixed point of Γ_D. The relation of different semantics is illustrated in Figure 2.9. We show a σ-interpretation is a γ-interpretation by drawing an arrow from σ-interpretation to γ-interpretation.

To clarify that ADFs are a proper generalization of Dung's notion for AFs it is explained in [17] that each AF is associated with an ADF which is rescripted in Definition 23. In addition, it is illustrated by a theorem in [17], which says the class of all semantics of an AF F and its associated ADF D_F are essentially the same.

Definition 23. For an AF $F = (A, R)$ define the ADF associated to F as $D_F = (A, R, C)$ with $C = \{\varphi_a\}_{a \in A}$ s.t. the acceptance condition is given by:

$$\varphi_a = \bigwedge_{a \in A, (b,a) \in R} \neg b$$

In Section 2.1 semantics of AFs are defined based on extensions and in Section 2.2 semantics of ADFs are defined based on interpretations. To investigate the correspondence between semantics of an AF F and its associated ADF D_F, first we show how these notions of semantics relate to one another.

Definition 24. Let $F = (A, R)$ be an AF, σ be a semantics $\sigma \in \{adm, prf, stb, com, grd\}$, and e be an extension of $\sigma(F)$, $e \in \sigma(F)$. The associated interpretation v_e is defined as follows.

$$v_e(a) = \begin{cases} t & a \in e \\ f & \exists b \in e, (b, a) \in R \\ u & otherwise \end{cases}$$

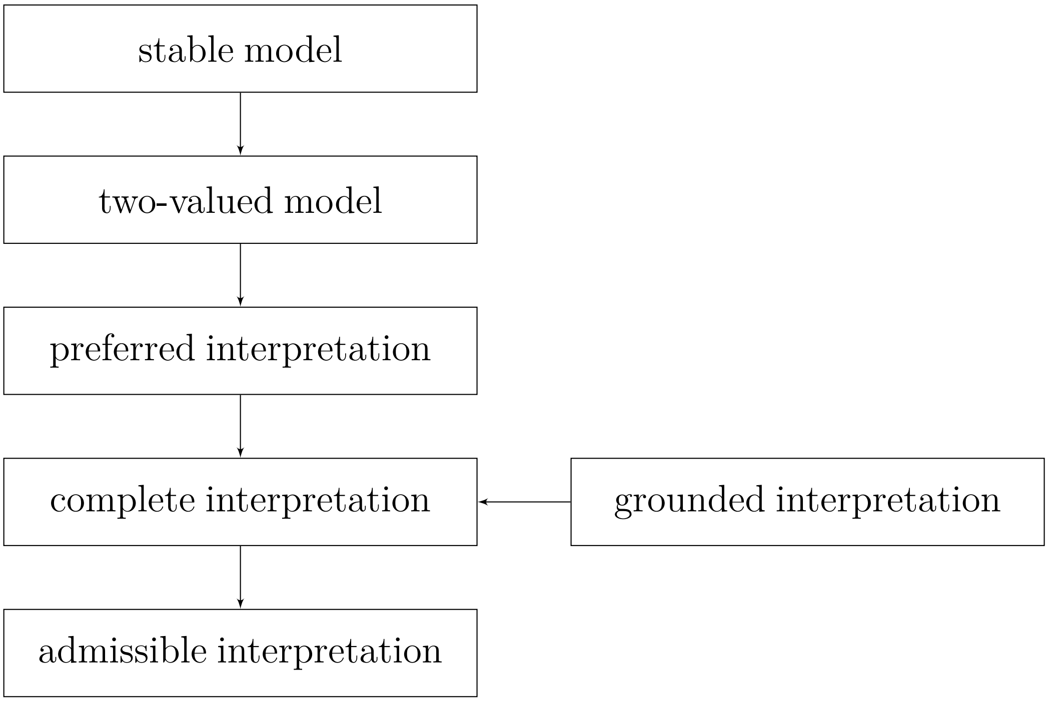

Figure 2.9: Relation between ADF semantics

Note that the associated interpretation to $e = \emptyset$ is v_ϵ which does not assign any argument to t, f or u.

Definition 25. Let $D = (A, L, C)$ be an ADF and v be a three-values interpretation of D. The associated extension e_v of v is:

$$e_v = \{a \in A \mid a \mapsto t \in v\}$$

By v_e which is defined in Definition 24 an interpretation correspondence to an extension e of an AF $F = (A, R)$ is computable by computing $v_e(a)$ for each $a \in A$. Note that the associated extension, e_v of v for $\sigma \in \{prf, com, grd\}$ is unique, however, for $\sigma = adm$ sometimes e_v is not one-to-one. In addition, v_e for $\sigma = adm$ is not always a subjective function. These facts are illustrated by Example 11.

Example 11. Let $F = (\{a, b, c, d\}, \{(a, b), (a, c), (c, a), (c, c), (b, d), (d, b), (d, d)\})$ be an AF depicted in Figure 2.10. By the Definition 23, the associated ADF to F is $D_F = (\{a, b, c, d\}, \{\varphi_a : \neg c, \varphi_b : \neg a \wedge \neg d, \varphi_c : \neg a \wedge \neg c, \varphi_d : \neg b \wedge \neg d\})$.
$adm(F) = \{\emptyset, \{a\}\}$ let $e_1 = \emptyset$ and $e_2 = \{a\}$. By the Definition 24 the associated interpretations of e_1 and e_2 are $v_{e_1} = \{a \mapsto u, b \mapsto u, c \mapsto u, d \mapsto u\}$ and $v_{e_2} = \{a \mapsto t, b \mapsto f, c \mapsto f, d \mapsto u\}$, respectively. It is clear that v_{e_1} and v_{e_2} are admissible

interpretations of D_F. In addition,

$$\begin{aligned} adm(D_F) = & \{\{a \mapsto u, b \mapsto u, c \mapsto u, d \mapsto u\}, \\ & \{a \mapsto t, b \mapsto f, c \mapsto f, d \mapsto u\}, \\ & \{a \mapsto t, b \mapsto u, c \mapsto f, d \mapsto u\}\} \end{aligned}$$

Which shows that v_e for $\sigma = adm$ is not a subjective function. By the Definition 25 the extension correspondence to the first interpretation of $adm(D_F)$ is $e_{v_1} = \emptyset$ and the extension correspondence to two others interpretations is $e_v = \{a\}$. That is, e_v for $\sigma = adm$ is not one-to-one.
Moreover, in this example $com(F) = prf(F) = \{\{a\}\}$, $grd(F) = \{a\}$. By Definition 24 the associated interpretation of e_2 is $v_{e_2} = \{a \mapsto t, b \mapsto f, c \mapsto f, d \mapsto u\}$ which is equivalent with the grounded, preferred and complete interpretation of D_F.

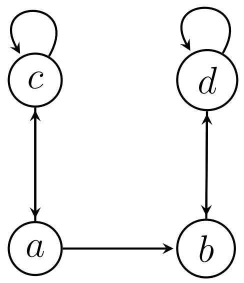

Figure 2.10: AF used in Example 11

In summary, let $F = (A, R)$ be an AF and $D_F = (A, R, C)$ be its associated ADF. If an extension e is admissible, preferred, complete or grounded for F then its associated interpretation v_e is admissible, preferred, complete or grounded for D_F, respectively. Moreover, if an interpretation v is admissible, preferred, complete or grounded for D_F its associated extension e_v is admissible, preferred, complete or grounded for F.

2.2.2 Bipolar ADFs

As mentioned before acceptance conditions characterize acceptability of arguments based on their parents' status. That is, each ADF $D = (S, L, C)$ can be represented by a tuple (S, C), set of arguments and their acceptance conditions, and implicity $(a, b) \in L$ iff a apears in φ_b. In this subsection the meaning of the link, $(a, b) \in L$ is investigated. In addition, we would like to study whether a restriction of acceptance conditions lead to a subclass of ADFs. The relations can be categorized in four groups, depending on whether they have support or attacking nature (or both or neither). This leads to the concept of bipolar ADFs (BADFs for short) defined in [16]. BADFs are a strict generalization of AFs and are a strict subclass of ADFs. BADFs rely on attacking and supporting which are defined as follows:

Definition 26. Let $D = (S, L, C)$ be an ADF. A link $(r, s) \in L$ is called

- supporting in D iff for all $v \in V_2$, $v(\varphi_s) = t$ implies $v \mid_t^r (\varphi_s) = t$,
- attacking in D iff for all $v \in V_2$, $v(\varphi_s) = f$ implies $v \mid_t^r (\varphi_s) = f$,
- dependent iff it is neither attacking nor supporting,
- redundant iff it is both attacking and supporting.

An ADF $D = (S, L, C)$ is a BADF if all links in L are supporting, attacking or both.

Definition 27. Let $D = (S, L, C)$ be an ADF and L^+ be the set of all support links of L and L^- be the set of all attack links of L. D is named a bipolar ADF (BADF) iff $L = L^+ \cup L^-$.

It is clear that the associated ADF D_F of each AF F is a bipolar ADF in which all links are attack, that is, $L = L^-$. Example 12 is an instance of an ADF which is a BADF and Example 13 is an instance of an ADF which is not a BADF.

Example 12. Let $D = (\{a, b, c\}, \{\varphi_a : c \to b, \varphi_b : \neg a \vee c, \varphi_c : a \wedge c\})$ be an ADF depicted in Figure 2.11.

We illustrate whether (c, a) is a support or attack. There are three two-valued interperetations, $v_1 = \{c \mapsto f, b \mapsto t\}, v_2 = \{c \mapsto t, b \mapsto t\}, v_3 = \{c \mapsto f, b \mapsto f\}$ by which $v_i(\varphi_a) = t$ for $i \in \{1, 2, 3\}$. However, $v_3|_t^c(\varphi_a) = f$ thus (c, a) is not a support relation.
There is a two-valued interpretation $v_4 = \{c \mapsto t, b \mapsto f\}$ by which $v_4(\varphi_a) = f$ and $v_4|_t^c(\varphi_a) = f$. Therefore, (c, a) is an attack relation.

To check whether (a, b) is a support link, there are two two-valued interpretations $v_1 = \{a \mapsto t\}$ and $v_2 = \{a \mapsto f\}$ by which $v_i(\varphi_b) = t$ for $i \in \{1, 2\}$. It is clear that $v_i|_t^a(\varphi_b) = t$ for $i \in \{1, 2\}$. Hence, (a, b) is support.
Since there is no two-valued interpretation by which $v(\varphi_b) = f$ then (a, b) is also an attack link. Since (a, b) is both an attack and support simultaneously, it is redundant.

It is easy to check that $(b, a), (a, c)$ and (c, c) are supports.

Therefore, the set of support links of D is $L^+ = \{(b, a), (a, b), (a, c), (c, c)\}$ and the set of attack links of D is $L^- = \{(a, b), (c, a)\}$. Since $L = L^+ \cup L^-$, D is a BADF.

Example 13 illustrates that the family of BADFs is a proper subset of ADFs.

Example 13. Let $D = (\{a, b, c\}, \{\varphi_a : b, \varphi_b : a \leftrightarrow c, \varphi_c : \neg b\})$ be an ADF depicted in Figure 2.12. We show that (a,b) is neither an attack nor a support.

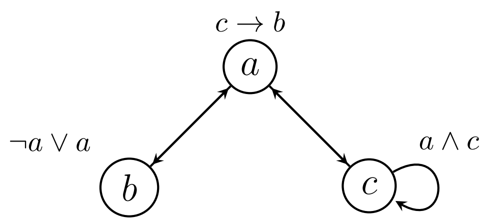

Figure 2.11: ADF used in Example 12

- There are two two-valued interpretations $v_1 = \{a \mapsto t, c \mapsto t\}$ and $v_2 = \{a \mapsto f, b \mapsto f\}$ under which $v_i(\varphi_b) = t$, for $i \in \{1, 2\}$. However, $v_2|_t^a(\varphi_b) = f$. That is, (a, b) is not a support relation.
- There are two two-valued interpretations $v_3 = \{a \mapsto t, c \mapsto f\}$ and $v_4 = \{a \mapsto f, b \mapsto t\}$ under which $v_i(\varphi_b) = f$, for $i \in \{1, 2\}$. However, $v_4|_t^a(\varphi_b) = t$. That is, (a, b) is not an attack relation.

That is $(a, b) \notin L^+$ and $(a, b) \notin L^-$. Then, $L \neq L^+ \cup L^-$. Therefore, D is not a BADF.

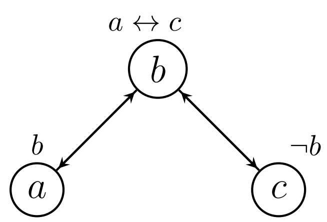

Figure 2.12: ADF used in Example 13

CHAPTER 3

Investigating Subclasses of Abstract Dialectical Framework

In this chapter we investigate whether certain results which have been shown for AFs carry over to ADFs. In [24] the properties under which certain extensions of an AF are equivalent is studied. In Section 3.1 the conditions under which main semantics of ADFs are collapsing into a unique semantics is studied. In Section 3.2 we study whether the properties of AFs, coherency and relatively groundedness, which are proven in [22] for symmetric AFs carry over to symmetric ADFs. To show that under which conditions these properties of AFs also hold for ADFs we introduce more fine-grained subclasses of ADFs: attack symmetric ADFs and acyclic support symmetric ADFs. In Section 3.3 we investigate a subclass of ADFs in which the sets of admissible and complete interpretations are the same.

3.1 Acyclic ADFs

In [24] an AF $F = (A, L)$ is named well-founded if there exists no infinite sequence of arguments $a_1, \dots, a_i, \dots$ s.t. for each i, $(a_{i+1}, a_i) \in L$. It is shown by Dung [24] that when a framework is well-founded all AF semantics are equivalent. That is, each well-founded AF $F = (A, L)$ has exactly one complete extension which is also grounded, preferred and stable. In this section we investigate under which conditions the results carry over to ADFs. That is, we illustrate whether different semantics can collapse to the same set of interpretations.

Definition 28. A finite ADF $D = (S, L, C)$ is named *acyclic* if its corresponding directed graph does not contain any directed cycle.

To show our results we need the concept of maximum level of ADFs defined as follows.

Definition 29. Let $D = (S, L, C)$ be an ADF. The *level* of an argument s is the number of links on the longest path from an initial argument to s plus one.

Example 14. Let $D = (S, L, C)$ be an ADF, depicted in Figure 3.1. The level of a is one and the level of c is 3. a is an initial argument and it is clear that the level of each initial argument is always one.

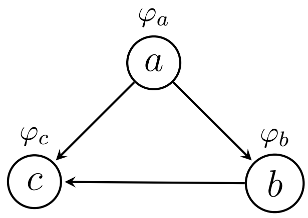

Figure 3.1: ADF used in Example 14

Definition 30. The *maximum level* of an (acyclic) ADF D is the level that is greater than every other level of an argument of D.

The maximum level of the ADF D in Example 14 is 3. Note that level and maximum level are not defined for cyclic ADFs.

Lemma 5. *When the maximum level of an acyclic ADF D is a number m then for each i, $1 \leq i \leq m$, there exists an argument of level i.*

Proof. Suppose D is an acyclic ADF with finite maximum level m. Towards a contradiction assume there exists an i between 1 and m such that there exists no argument with level i. By our assumption the maximum level of D is m, that is, there exists an argument s_m with level m. By the definition of level there is a path between an initial argument s_1 and s_m with length $m - 1$. Define s_j, for each $1 \leq j \leq m$, as the argument such that the number of the links in this path between s_1 and s_j is $j - 1$. By the contradiction assumption there is no argument with level i. Then the level of s_i is not i. It is clear that the level of s_i cannot be less than i. Suppose level of s_i is greater than i. Then, there exists an initial argument s'_1 and there exists a path between s'_1 and s_i with length k such that $k \geq i$. Therefore, there exists a path between s'_1 and s_m with length $m - i + k - 1$ which is greater than $m - 1$. This is a contradiction by the assumption that the maximum level of D is m. □

Proposition 1. *In every acyclic ADF D with maximum level m the least fixed point of Γ_D is a two-valued model.*

Proof. Assuming $D = (S, L, C)$ is an acyclic ADF. Define the following interpretations.

- $v_0 := S \mapsto \{u\}$, maps all the arguments to unknown.

- $v_i := \Gamma_D(v_{i-1})$ for $1 \leq i \leq m$.

Claim: For all i, $1 \leq i \leq m$, and an arbitrary argument s_j which is in level j, if $j \leq i$ then either $v_i(s_j) = t$ or $v_i(s_j) = f$.
Proof of claim by induction on i:

- Base case: Suppose s_1 is an arbitrary argument of level one (an acyclic ADF always includes an initial argument). Since s_1 is an initial argument, its acceptance condition φ_{s_1} is either $\top$ or $\bot$. Then, $v_1(s_1) = \Gamma_D(v_0)(s_1)$ is either true or false.
- Inductive step: Assuming this property holds for all k, $1 \leq k \leq i < m$ we show it holds for $i+1$.
  We know that: $\varphi_{s_j}^{v_i} = \varphi_{s_j}[s_k/\top : v_i(s_k) = t][s_k/\bot : v_i(s_k) = f]$ such that s_k is an argument of φ_{s_j} which occurs in a level less than j.
  For all s_k that occur in φ_{s_j} if $j \leq i+1$ then $k < j \leq i+1$. Therefore, by induction hypothesis for each s_k, $v_i(s_k)$ is either true or false. Then, $\varphi_{s_j}^{v_i}$ can be simplified to either $\top$ or $\bot$. Therefore, $v_{i+1}(s_j)$ is either true or false.

Claim: v_m is a two-valued model. We know that m is a maximum level, that is, the level of all arguments is less than or equal to m. Therefore, by the previous claim $v_m(s_j)$ is either true or false for all $s_j \in S$, i.e. it is a two-valued interpretation. It remains to show that $v_m = \Gamma(v_m)$. Let $s_j \in S$ be an arbitrary argument of level j, $1 \leq j \leq m$. Assuming $v_m(s_j)$ is true (the proof method for $v_m(s_j) = f$ is exactly the same), then $\varphi_{s_j}^{v_{m-1}}$ can be simplified to $\top$. Therefore, $\varphi_{s_j}^{v_m} = \varphi_{s_j}^{v_{m-1}} \equiv \top$. Hence, $\Gamma_D(v_m) = v_m$ that is, v_m is a two-valued model.
Claim: v_m is the least fixed point of Γ_D. In the previous claim it is shown that $\Gamma_D(v_m) = v_m$, i.e. v_m is a fixed point of Γ_D. Towards a contradiction assume v_m is not the least fixed point. Then there exists an interpretation v which is a least fixed point. That is, $v = \Gamma_D(v)$, $v \leq_i v_m$ and $v \neq v_m$. Then there exists an argument s such that $v_m(s)$ is either true or false and $v(s)$ is undecided. Assume s is in level i. Since D is an acyclic ADF all arguments s_k that occur in φ_s are in level less than i. Therefore, there exists at least an argument s_j of level $j < i$ in φ_s such that $v(s_j)$ is undecided. By iterating this method after at most $i-1$ times we reach an argument of level one which is undecided. This is a contradiction, since the acceptance condition of each initial argument is either $\top$ or $\bot$. □

Corollary 1 is an immediate consequence of Proposition 1 together with the fact that each ADF D possesses a unique grounded interpretation.

Corollary 1. *Every acyclic ADF D with maximum level m has a non-trivial grounded interpretation.*

Let v be the least fixed point of Γ_D. Thus v is a two-valued model. Intuitively, it is clear that v is a stable model. Because, there is a constructive proof for all arguments assigned to t by v. In Proposition 2 it is formally proven that in acyclic ADFs, their two-valued models and stable models coincide.

Proposition 2. *In every acyclic ADF D with maximum level m each two-valued model of D is a stable model of D.*

Proof. Let $D = (S, L, C)$ be an ADF with maximum level m. Assume v is a two-valued model of Γ_D. Further, let $D^v = (S^v, L^v, C^v)$ be a reduct of D and w be the unique grounded interpretation of D^v. Since D^v is also an acyclic ADF then for each $s_k \in S^v$, $w(s_k)$ is either true or false. Towards a contradiction suppose v is not a stable model. Therefore, there exists an argument $s_k \in S$ in level k, $1 \leq k \leq m$ such that $v(s_k) = t$ and $w(s_k) = f$. We show the contradiction in two following steps:

- If $k = 1$ then s_1 is an initial argument. By the definition, $v(s_1) = t$ if and only if $\varphi^v_{s_1} \equiv \top$ and by the reduct definition $s_1 \in S^v$. s_1 is an initial argument then, $\varphi^v_{s_1} = \varphi^{v_0}_{s_1} = \varphi_{s_1} \equiv \top$. $s_1 \in S^v$ and $\varphi_{s_1} \equiv \top$ then, by the definition of ground interpretation $w(s_1) = t$. That is, for all initial arguments s, $v(s) = t$ implies $w(s) = t$. In addition, for an initial argument s_1 if $v(s_1) = f$ then $\varphi_{s_1} \equiv \bot$ and by the reduction definition $s_1 \notin S^v$. Then $w(s_1) \neq t$. Therefore, for each initial argument s, $v(s) = t$ if and only if $w(s) = t$.

- Let s_k be an argument in level k, $1 < k \leq m$, such that $v(s_k) = t$ and $w(s_k) = f$. Then φ_{s_k} contains an argument s_i, $1 \leq i < k$ such that $v(s_i) \neq w(s_i)$. If $i = 1$ by the first step it is a contradiction. Otherwise, $v(s_i) \neq w(s_i)$ means φ_{s_i} contains an argument s_j, $1 \leq j < i$, such that $v(s_j) \neq w(s_j)$. Since D is an ADF of maximum level by iterating this process after finite steps we reach an initial argument s such that $v(s) \neq w(s)$. This has a contradiction with the fact proven in the first step which says for each initial argument s of an acyclic ADF D, $v(s) = t$ if and only if $w(s) = t$.

□

The following theorem shows that each acyclic ADF D with maximum level m has exactly one complete interpretation.

Theorem 3. *An acyclic ADF $D = (S, L, C)$ with maximum level m has exactly one complete interpretation which is grounded, two-valued model, preferred and stable.*

Proof. Let D be an acyclic ADF with maximum level m. We show that the grounded interpretation v_g is a stable model. By Proposition 1 the least fixed point of Γ_D is a two-valued model. That is, the grounded interpretation is a two-valued model. By Proposition 2 each two-valued model is a stable model. Therefore, the grounded interpretation v_g is a stable model. By Theorem 2 the grounded interpretation of D is a complete interpretation of D. Since the grounded interpretation v_g of D is a two-valued model, D does not contain any other complete interpretations. By Theorem 2 each stable model of D is a two-valued model of D, each two-valued model of D is a preferred interpretation of D and each preferred interpretation of D is a complete interpretation of D. Thus, D has exactly one complete interpretation which is grounded, two-valued model, preferred and stable as well. □

To summarize, all main semantics coincide for acyclic ADFs.

3.2 Symmetric ADFs

It is shown in [22] that the family of Dung's finite AFs for which attacks are nonempty, irreflexive and symmetric, every element of this family is coherent and relatively grounded, however, none of them is well-founded. That is, in this family of AFs which is named symmetric AFs, $prf(D) = stb(D)$ and $grd(D) = \bigcap prf(D)$, for each AF D. In Section 3.2.1 we investigate the relation between extensions of a symmetric AF which are not investigated by Coste-Marquis et al. [22]. For instance, we illustrate whether the set of admissible extensions and complete extensions (resp. complete extension and preferred extension) of symmetric AFs are equivalent. As mentioned in Section 3.1 all semantics of an acyclic ADF are equivalent. The results of [22] motivate us to answer to the natural question whether the same properties of symmetric AFs carry over to a subclass of ADFs. In Sections 3.2.2, 3.2.3 and 3.2.4 first we define subclasses of ADFs and then we illustrate whether they are coherent and relatively grounded and also study several other properties.

3.2.1 Symmetric AFs

In [22] the relation between the set of admissible extensions and the set of complete extensions (resp. the set of complete extensions and the set of preferred extensions) in symmetric AFs is not shown. To investigate that the sets of former extensions are not equivalent we draw our attention to the Example 15; Example 16 will clarify that the complete and preferred extensions are not equivalent. In the following first symmetric AFs, coherency and relatively groundedness are recalled formally by Definitions 31, 32 and 33, respectively.

Definition 31. Let $F = (A, L)$ be an AF. It is called *symmetric AF* if L is symmetric and irreflexive.

Definition 32. Let $F = (A, L)$ be an AF. It is named *coherent* whenever the set of all preferred extensions and the set of all stable extensions of F are equivalent, i.e. $prf(F) = stb(F)$.

Definition 33. Let $F = (A, L)$ be an AF. It is called *relatively grounded* if the intersection of preferred extensions and the grounded extension are the same, i.e. $grd(F) = \bigcap prf(F)$.

In the definition of symmetric AFs in [22] it is assumed that the attack relation is non-empty. However, without considering this extra condition all properties proven in [22] remain unchanged, that is, each symmetric AF is coherent and relatively grounded. We thus will not make this assumption in the current work.

Proposition 3. *For each symmetric AF $F = (A, L)$.*

1. $cf(F) = adm(F)$.
2. $nai(F) = stb(F) = prf(F)$.

Proof. 1. Whenever L is non-empty, it is proven in [22] that $cf(F) = adm(F)$. Whenever L is an empty set, $cf(F) = adm(F) = \{B \mid B \subseteq A\}$. Hence, in each symmetric AF $cf(F) = adm(F)$.

2. It is shown in [22] that whenever L is non-empty, each symmetric AF is coherent. Let L be the empty set, then, $prf(F) = stb(F) = \{A\}$. Therefore, each symmetric AF is coherent. In addition, by the definition each naive extension is a maximal conflict-free set w.r.t. set inclusion and each preferred extension is a maximal admissible set w.r.t. set inclusion. In a family of symmetric AFs, even when L is the empty set, $nai(F) = prf(F)$ since $cf(F) = adm(F)$. Therefore, $nai(F) = stb(F) = prf(F)$.

□

In the following we investigate whether in each symmetric AF F, $com(F) = adm(F)$ and $com(F) = prf(F)$.

Example 15. Let $F = (\{a, b, c\}, \{(a, b), (b, a), (a, c), (c, a)\}$ be a symmetric AF, depicted in Figure 15. The set $\{c\}$ is an admissible extension but it is not a complete extension. Therefore, $adm(F) \neq com(F)$.

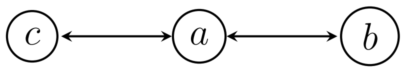

Figure 3.2: Symmetric AF used in Example 15

Example 16. Let $F = (\{a, b, c\}, \{(a, b), (b, a)\})$ be a symmetric AF, depicted in Figure 3.3. The single grounded extension of F is $\{c\}$. However, $prf(F) = \{\{a, c\}, \{b, c\}\}$. That is, the set of preferred extensions and complete extensions are not equivalent.

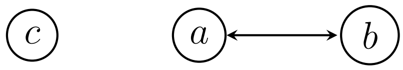

Figure 3.3: Symmetric AF used in Example 16

By Proposition 4 it is shown that symmetric AFs are relatively grounded, even when the attack relation L is the empty set.

Proposition 4. *For a symmetric AF F, it holds that $grd(F) = \bigcap prf(F)$.*

Proof. In [22] it is proven that if $F = (A, L)$ is a symmetric AF in which L is non-empty then F is relatively grounded. To show that the result is true even when L is the empty set assume that $L = \emptyset$. In this case, since $com(F) = prf(F) = \{A\}$, the unique grounded extension of F is $\{A\}$. Therefore, $grd(F) = \bigcap prf(F)$. ☐

3.2.2 General Symmetric ADFs

To clarify whether the properties of symmetric AFs which are proven in [22] carry over to symmetric ADFs, that is, whether the family of symmetric ADFs is coherent and relatively grounded, we first generalize the definition of symmetric AFs to symmetric ADFs. Then we show that these properties do not carry over to symmetric ADFs in general. In addition, we restrict ADFs to some subclasses to clarify under which specific conditions some of these properties hold.

Definition 34. (Symmetric ADF)
A symmetric ADF is a finite abstract dialectic framework $D = (S, L, C)$, in which L is irreflexive and symmetric and L does not contain any redundant links.

Definition 35. An ADF $D = (S, L, C)$ is said to be coherent if each preferred interpretation of D is a stable model.

In the following we investigate the coherency of symmetric ADFs by some examples.

Example 17. Let $D = (\{a, b, c\}, \{\varphi_a : \neg b \vee \neg c, \varphi_b : \neg a, \varphi_c : \neg a\})$ be the symmetric ADF, depicted in Figure 3.4. D has two preferred interpretations $v_1 = \{a \mapsto t, b \mapsto f, c \mapsto f\}$ and $v_2 = \{a \mapsto f, b \mapsto t, c \mapsto t\}$ and each of them is a stable model. Hence, D is coherent.

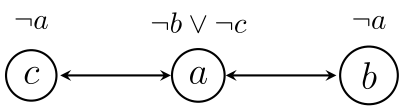

Figure 3.4: ADF used in Example 17

Example 18. Let $D = (\{a, b\}, \{\varphi_a : b, \varphi_b : a\})$ be the ADF depicted in Figure 3.5. D is a symmetric ADF and both links are supporting. D has two preferred interpretation $v_1 = \{a \mapsto t, b \mapsto t\}$ and $v_2 = \{a \mapsto f, b \mapsto f\}$. Both of them are two-valued models and v_2 is stable model. However, v_1 is not a stable model. Therefore, D is not coherent.

In the following example a symmetric ADF is investigated in which only one of the links is support, however it is not coherent.

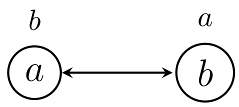

Figure 3.5: ADF used in Example 18

Example 19. Let $D = (\{a, b\}, \{\varphi_a : b, \varphi_b : \neg a\})$ be the ADF depicted in Figure 3.6 in which the link (a, b) is attacking and the link (b, a) is supporting. The trivial interpretation v_u is a unique preferred interpretation of D. That is, the preferred interpretation of D is not a two-valued model. Therefore, D is not coherent.

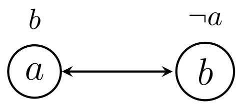

Figure 3.6: ADF used in Example 19

Due to this distinction between two-valued and stable models in ADFs, different levels of coherency can be considered which are defined in the Definitions 36 and 37.

Definition 36. A symmetric ADF $D = (S, L, C)$ is called weak-coherent if each two-valued model of D is a stable model.

Definition 37. A symmetric ADF $D = (S, L, C)$ is called semi-coherent if each preferred interpretation of D is a two-valued model.

Example 18 is an instance of a semi-coherent ADF which is not weak-coherent and Example 19 is an instance of a weak-coherent ADF which is not semi-coherent. One may suppose that whenever an ADF contains a support link it is not coherent. To investigate whether an ADF contains a symmetric link is coherent we draw your attention to Example 20.

Example 20. Let $D = (\{a, b, c\}, \{\varphi_a : b \wedge \neg c, \varphi_b : \neg a, \varphi_c : \neg a\})$ be the symmetric ADF depicted in Figure 3.7. $v = \{a \mapsto f, b \mapsto t, c \mapsto t\}$ is the unique preferred interpretation of D which is also both a two-valued model and a stable model. That is, D is a symmetric ADF including a support which is semi-coherent, weak-coherent and consequently, coherent as well.

Proposition 5. *An ADF is coherent if and only if it is semi-coherent and weak-coherent.*

Another concept which is expressed in [22] is relatively groundedness. It is demonstrated in [22] that each symmetric AF is relatively grounded. First we generalize this definition to ADFs and then we illustrate whether each symmetric ADF is relatively grounded. An ADF D is named relatively grounded if the unique grounded interpretation of D is

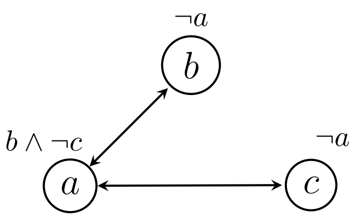

Figure 3.7: Symmetric ADF used in Example 20

equivalent to the meet of all preferred interpretations of D. It is formally defined as follows:

Definition 38. An ADF $D = (S, L, C)$ is said to be relatively grounded if $grd(D) = \bigsqcap_i prf(D)$.

Example 21. The symmetric ADF $D = (\{a, b, c\}, \{\varphi_a : \neg b \vee \neg c, \varphi_b : \neg a, \varphi_c : \neg a\})$ defined in Example 17 and depicted in Figure 3.4 is relatively grounded, since the meet of two preferred interpretations $v_1 = \{a \mapsto t, b \mapsto f, c \mapsto f\}$ and $v_2 = \{a \mapsto f, b \mapsto t, c \mapsto t\}$ is the trivial interpretation and the grounded interpretation of D is the trivial interpretation as well.

To illustrate that it is not the case that each symmetric ADF is relatively grounded we draw our attention to Example 22.

Example 22. Let $D = (\{a, b, c\}, \{\varphi_a : c \vee \neg b, \varphi_b : c \vee \neg a, \varphi_c : b \vee \neg a\})$ be a symmetric ADF, depicted in Figure 3.8. The set of preferred interpretations of D is, $prf(D) = \{\{a \mapsto t, b \mapsto t, c \mapsto t\}, \{a \mapsto t, b \mapsto f, c \mapsto f\}\}$ and the unique grounded interpretation of D is the trivial interpretation. However, the meet of preferred interpretation is $\{a \mapsto t, b \mapsto u, c \mapsto u\}$, hence D is not relatively grounded.

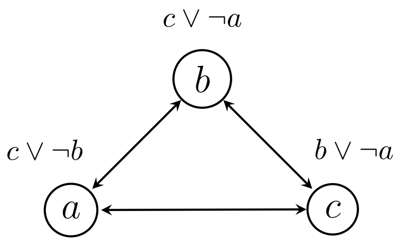

Figure 3.8: ASADF used in Example 22

Example 22 investigates that symmetric ADFs are not relatively grounded in general. In the following we investigate whether there exist subclasses of symmetric ADF which are coherent and relatively grounded.

3.2.3 Attack Symmetric ADFs

In [22] it is proven that each symmetric AF F is coherent. By the definition in [17] for each AF $F = (S, R)$ there exists an associated ADF $D_F = (S, R, C)$, (cf. Definition 23). Hence the corresponding D_F of a symmetric AF F is also symmetric. It is shown in [17] that an interpretation is admissible, complete, preferred, grounded for F iff it is admissible, complete, preferred, grounded for D_F, in the sense of Definitions 24 and 25. Therefore, the corresponding symmetric ADF D_F of symmetric AF F is also coherent. Based on the definition all relations in D_F associated to F are attacks. One may guess symmetric ADFs in which all relations are attack relations are coherent. However, by Example 23 it is illustrated that this assumption is not true.

Definition 39. A symmetric ADF $D = (S, L, C)$ in which all links are attacking is named *attack symmetric ADF* (ASADF for short).

Example 23. Let

$$\begin{aligned} D = (&\{a, b, c, d, e\},\\ &\{\varphi_a : \neg c \wedge (\neg d \vee \neg b),\\ &\ \varphi_b : \neg a \wedge (\neg d \vee \neg c),\\ &\ \varphi_c : \neg b \wedge (\neg d \vee \neg a),\\ &\ \varphi_d : \neg e \wedge (\neg a \vee \neg b \vee \neg c),\\ &\ \varphi_e : \neg d\}). \end{aligned}$$

D depicted in Figure 3.9 is an attack symmetric ADF and it has four preferred interpretations:

$$\begin{aligned} v_1 =& \{a \mapsto f, b \mapsto f,\ c \mapsto t,\ d \mapsto t,\ e \mapsto f\},\\ v_2 =& \{a \mapsto f,\ b \mapsto t,\ c \mapsto f,\ d \mapsto t,\ e \mapsto f\},\\ v_3 =& \{a \mapsto t,\ b \mapsto f,\ c \mapsto f,\ d \mapsto t,\ e \mapsto f\},\\ v_4 =& \{a \mapsto u,\ b \mapsto u,\ c \mapsto u,\ d \mapsto f,\ e \mapsto t\}. \end{aligned}$$

Each two-valued interpretation of D, v_1, v_2 and v_3, is stable model, that is, D is weak-coherent. v_4 is a preferred interpretation which is not a two-valued model. Hence, D is not semi-coherent, and therefore, D is not coherent.

Although, by Example 23 it is shown that attack symmetric ADFs are not coherent, it seems that they are weak-coherent. To prove this statement formally in Proposition 4 we first prove Lemma 6.

Lemma 6. *Let $D = (S, L, C)$ be an ADF, v be a two-valued model of D and $s \in S$ be an argument s.t. all parents of s are attackers and s does not occur in φ_s. If φ_s^v is irrefutable then $\varphi_s[s_i/\bot\ :\ v(s_i) = f]$ is irrefutable.*

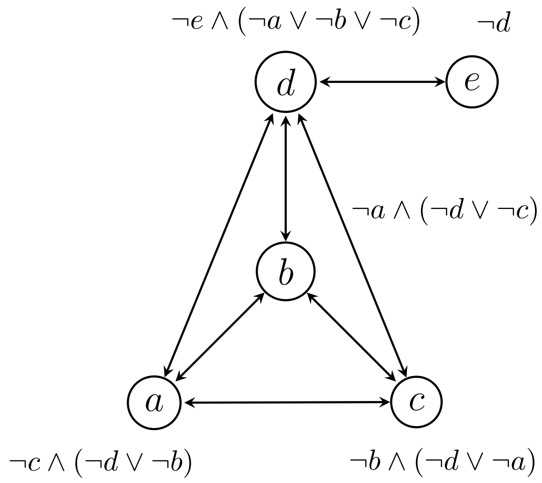

Figure 3.9: ASADF used in Example 23

Proof. Assume v is a two-valued model of D. Towards a contradiction assume φ_s^v is irrefutable and $\varphi_s[s_i/\bot \;:\; v(s_i) = f]$ is not irrefutable. Since, all parents of s are attackers, each s_i in acceptance condition of s occurs in the form of $\neg s_i$. By our assumption $\varphi_s[s_i/\bot \;:\; v(s_i) = f]$ is not irrefutable, there are two cases: either it is unsatisfiable or it is undecided. If it is unsatisfiable then φ_s^v will be unsatisfiable which is a contradiction. Suppose it is undecided, since s does not occur in φ_s, φ_s contains arguments $s_i \neq s$ such that $v(s_i) \neq f$. Since v is a two-valued model, $v(s_i) = t$. Because φ_s only contains connectives $\wedge/\vee$ by replacing the arguments which are assigned to true by v in φ_s we have φ_s^v is unsatisfiable. This is a contradiction with our assumption that φ_s^v is irrefutable. □

In specific, since in each attack symmetric ADF all relations are irreflexive and all links are attacking, Corollary 2 is an immediate consequence of Lemma 6.

Corollary 2. *Let $D = (S, L, C)$ be an attack symmetric ADF and v be a two-valued model of D. If φ_s^v is irrefutable then $\varphi_s[s_i/\bot \;:\; v(s_i) = f]$ is irrefutable.*

Proposition 6. *Let $D = (S, L, C)$ be an attack symmetric ADF. Each two-valued model of D is a stable model.*

Proof. Assume D is an attack symmetric ADF and $v : S \to \{t, f\}$ is a two-valued model of D. We show that v is a stable model of D. Suppose $D^v = (S^v, L^v, C^v)$ is the reduct of D

and w is the unique grounded interpretation of D^v. We show that for each $s \in S$, $v(s) = t$ implies $w(s) = t$. Let s be an argument such that $v(s) = t$, that is φ_s^v is irrefutable. By the definition, $\varphi_s[s_i/\top \quad : \quad v(s_i) = t][s_i/\bot \quad : \quad v(s_i) = f]$ is irrefutable, in which s_i are arguments occurring in φ_s, by Corollary2, $\varphi_s[s_i/\bot \quad : \quad v(s_i) = f]$ is irrefutable. That is, for each $s \in S^v$, φ_s^v is irrefutable, therefore, $w(s) = t$. □

Theorem 4. *Every attack symmetric ADF is weak-coherent.*

Proof. Let $D = (S, L, C)$ be an attack symmetric ADF. By Proposition 6 each two-valued model of D is a stable model of D. In general, for every ADF, every stable model is a two-valued model. Therefore, in D the sets of stable and two-valued models are equivalent. Thus, D is weak-coherent. □

In Section 3.2.2 it is shown by Example 22 that symmetric ADFs are not relatively grounded in general. It is shown in [22] that each symmetric AF F is relatively grounded. Therefore, its associated ADF D_F is also relatively grounded. Then, a natural question is that whether the family of attack symmetric ADFs is relatively grounded. In the following we investigate by Example 24 that there exists a symmetric ADF in which all relations are attacks but it is not relatively grounded.

Example 24. Let $D = (\{a, b, c\}, \{\varphi_a : \neg b \vee \neg c,\ \varphi_b : \neg a \wedge \neg c,\ \varphi_c : \neg a \vee \neg b\})$ be an attack symmetric *ADF* depicted in Figure 3.10.

- The unique preferred interpretation of D is $v_p = \{a \mapsto t,\ b \mapsto f, c \mapsto t\}$ and the grounded interpretation of D is the trivial interpretation. That is, the meet of all preferred interpretations is not the same as the grounded interpretation. Therefore, D is not relatively grounded.

- In addition, this example shows that complete interpretations and preferred interpretations are not the same in the family of attack symmetric ADFs, $prf(D) \neq com(D)$.

- In addition, v_p is a two-valued model which is a stable model. Hence, the grounded interpretation is not stable in this family, i.e. $stb(D) \neq \{grd(D)\}$.

- D contains four admissible interpretations. One of them is $\{a \mapsto u,\ b \mapsto f, c \mapsto t\}$ which is not preferred. Since, each preferred interpretation is a complete interpretation we can conclude that in this family the set of admissible interpretations and complete interpretations are not the same, i.e. $adm(D) \neq com(D)$.

The following Example 25 is an instance of attack symmetric ADFs which contains only one $\vee$ connective, however, it is not relatively grounded. Note that without the occurrence of $\vee$, this ADF would be associated to some symmetric AF, and thus be relatively grounded.

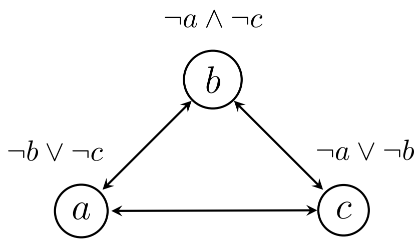

Figure 3.10: ASADF used in Example 24

Example 25. Let $D = (\{a, b, c, d\}, \{\varphi_a : \neg b \wedge \neg c,\ \varphi_b : \neg a \wedge \neg c \wedge \neg d,\ \varphi_c : \neg a \wedge \neg b \wedge \neg d,\ \varphi_d : \neg b \vee \neg c\})$. D is an attack symmetric ADF depicted in Figure 3.11 that contains only one $\vee$ connective. $v_p = \{a \mapsto t, b \mapsto f, c \mapsto f, d \mapsto t\}$ is the unique preferred interpretation of D, however, its grounded interpretation is the trivial interpretation. Therefore, D is not relatively grounded.

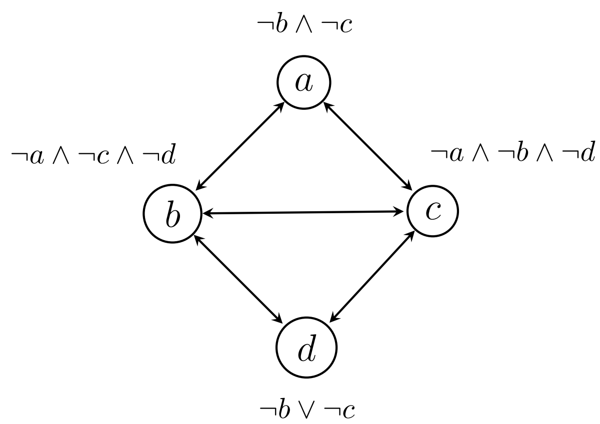

Figure 3.11: ASADF used in Example 25 which is not relatively grounded

Attack symmetric ADF of Examples 24 and 25 contain just one preferred interpretation. In Example 26 we investigate the attack symmetric ADF with more than one preferred interpretation such that their meet is not the trivial interpretation and it is not relatively grounded.

Example 26. Let $D = (\{a, b, c, d, e, f\}, \{\varphi_a : \neg b \wedge \neg c \wedge \neg e,\ \varphi_b : \neg a \wedge \neg c \wedge \neg d,\ \varphi_c : \neg a \wedge \neg b \wedge \neg d,\ \varphi_d : \neg b \vee \neg c,\ \varphi_e : \neg a \vee \neg f,\ \varphi_f : \neg e\})$ be the ADF depicted in Figure 3.12. The meet of the preferred interpretations is $\{a \mapsto u, b \mapsto f, c \mapsto f, d \mapsto t, e \mapsto u, f \mapsto u\}$

and the grounded interpretation is the trivial interpretation, v_u. Hence, D is not relatively grounded.

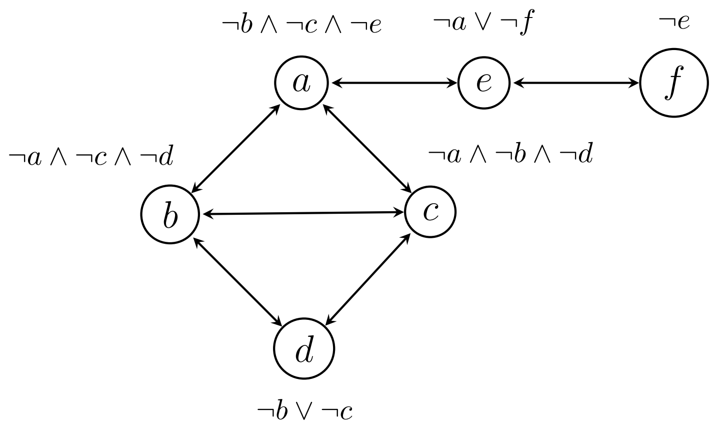

Figure 3.12: ASADF used in Example 26 which is not relatively grounded

By Examples 24, 25 and 26 we illustrated that attack symmetric ADFs are not relatively grounded in general. In the following we investigate under which condition an attack symmetric ADF is relatively grounded.

Lemma 7. *Let $D = (S, L, C)$ be an attack symmetric ADF such that it does not contain any isolated argument. The grounded interpretation of D is the trivial interpretation, v_u.*

Proof. Let $D = (A, L, C)$ be an arbitrary attack symmetric ADF that does not have any isolated argument. To show that the grounded interpretation is the trivial interpretation it is enough to show that the meet of all complete interpretations of D is the trivial interpretation, i.e. $\bigsqcap_i com(D) = v_u$. In other words, we show that $\Gamma_D(v_u) = v_u$. Suppose to the contrary that $\Gamma_D(v_u) \neq v_u$, then there exists at least an argument a s.t. $\Gamma_D(v_u)(a)$ is either assigned to t or f. W.l.o.g. assume that $\Gamma_D(v_u)(a) = t$ (the proof method for $\Gamma_D(v_u) = f$ is exactly the same). Let w_1 and w_2 be two completions of v_u s.t. in w_1 all arguments of $par(a)$ are assigned to t and in w_2 all arguments of $par(a)$ are assigned to f. Since D is an attack symmetric ADF all arguments appear in φ_a in the form of negation of arguments of $par(a)$ and the only connectives in φ_a are $\wedge/\vee$. In addition, since the set of isolated arguments of D is the empty set and D is an attack symmetric ADF, $par(a) \neq \emptyset$ for each $a \in A$. Therefore, it is obvious that $w_1(\varphi_a) = f$ and $w_2(\varphi_a) = t$. Then, $w_1(\varphi_a) \sqcap_i w_2(\varphi_a) = u$. That is, $\Gamma_D(v_u)(a) = u$ which is a contradiction with our assumption. Hence, $\Gamma_D(v_u) = v_u$. That is, the grounded interpretation of D is the trivial interpretation.

□

Lemma 8. *ADF D is relatively grounded when the meet of all preferred interpretation is the trivial interpretation, v_u.*

Proof. Assume D is an ADF in which the meet of all preferred interpretations is the trivial interpretation, i.e. $\bigsqcap_i prf(D) = v_u$. Since each preferred interpretation is a complete interpretation, $\bigsqcap_i com(D) \leq_i \bigsqcap_i prf(D)$. Therefore, $\bigsqcap_i com(D) = v_u$ which is the grounded interpretation of D. Then, D is relatively grounded. □

Theorem 5. *An attack symmetric ADF D without any isolated argument is relatively grounded iff the meet of all preferred interpretations is the trivial interpretation, v_u.*

Proof. ($\rightarrow$) Since D is an attack symmetric ADF by Lemma 7 the grounded interpretation of D is the trivial interpretation. By theorem assumption D is relatively grounded, that is, the meet of all preferred interpretations and grounded interpretation are equivalent. Hence, the meet of all preferred interpretations is the trivial interpretation.

($\leftarrow$) by Lemma 8.

□

It is also proven in [22] that in the family of symmetric AFs conflict-freeness and admissibility are equivalent. In the following we investigate whether this property carries over to ADFs. Example 27 illustrates that in attack symmetric ADFs the set of conflict-free interpretations and the set of admissible interpretations can be different.

Example 27. Assume

$$\begin{aligned} D = (&\{a, b, c, d, e\}, \\ &\{\varphi_a : \neg b \vee (\neg c \wedge \neg d), \\ &\ \varphi_b : \neg c \vee (\neg a \wedge \neg d), \\ &\ \varphi_c : \neg a \vee (\neg b \wedge \neg d), \\ &\ \varphi_d : (\neg a \wedge \neg b \wedge \neg c) \vee \neg e, \\ &\ \varphi_e : \neg d\}) \end{aligned}$$

as an attack symmetric ADF, depicted in Figure 3.13. We claim that the set of conflict-free interpretations of D and the set of admissible interpretations of D are not equivalent. In the following we show that $v = \{a \mapsto t,\ b \mapsto u,\ c \mapsto f, d \mapsto t,\ e \mapsto f\}$ is a conflict-free interpretation of D which is not an admissible interpretation. To check whether v is conflict-free we show that φ_a^v and φ_d^v are satisfiable and φ_c^v and φ_e^v are unsatisfiable. It is easy to check that $\varphi_d^v \equiv \top$ and $\varphi_c^v = \varphi_e^v \equiv \bot$. Therefore, the truth value of c, d and e are unchanged in all completions of v. That is, φ_c^v and φ_d^v are unsatisfiable and φ_d^v is not only satisfiable but also irrefutable. To illustrate that v is conflict-free, we show that $\varphi_a' = \varphi_a^v = \varphi_a[c/\bot\ :\ v(c) = f][d/\top\ :\ v(d) = t] = \neg b$ is satisfiable. Let w_1 be a completion of v in which b is assigned to f then $\varphi_a'^{w_1} \equiv \top$. That is, φ_a^v is satisfiable. To illustrate that v is not an admissible interpretation, we show that φ_a^v is not irrefutable. Let w_2 be a completion of v in which b is assigned to t then, $\varphi_a'^{w_2} \equiv \bot$. Therefore, φ_a^v is not irrefutable. That is, v is a conflict-free but not an admissible interpretation of D. That is, $adm(D) \neq cf(D)$.

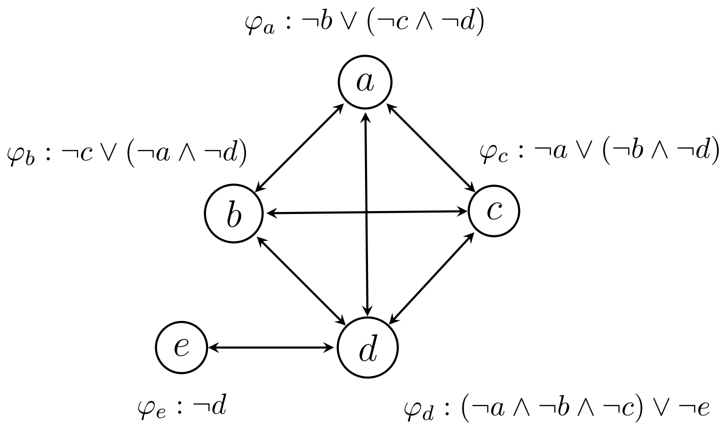

Figure 3.13: ADF used in Example 27

The following Example 28 shows an attack symmetric ADF which contains only one connective $\vee$ in which conflict-freeness and admissibility are not the same.

Example 28. Let $D = (\{a, b, c\}, \{\varphi_a : \neg b \wedge \neg c,\ \varphi_b : \neg a \wedge \neg c,\ \varphi_c : \neg a \vee \neg b\})$ be an attack symmetric ADF, depicted in Figure 3.14. We claim that $v = \{a \mapsto t, b \mapsto f, c \mapsto u\}$ is a conflict-free interpretation of D which is not admissible. To verify that v is a conflict-free interpretation we check whether φ_b^v is unsatisfiable and φ_a^v is satisfiable. Replacing truth value of a in φ_b results in $\varphi_b^v \equiv \bot$. Therefore, φ_b^v is unsatisfiable. Let w_1 and w_2 be two completions of v s.t. c is assigned to t by w_1 and c is assigned to f by w_2. Hence, φ_a^v is satisfiable under w_2 and it is not satisfiable under w_1. That is, φ_a^v is not irrefutable. Therefore, v is a conflict-free interpretation which is not admissible.

3.2.4 Acyclic Support Symmetric ADFs

In this subsection we investigate the coherency and relatively groundedness of another subclass of ADFs called acyclic support symmetric ADFs which is defined in the following.

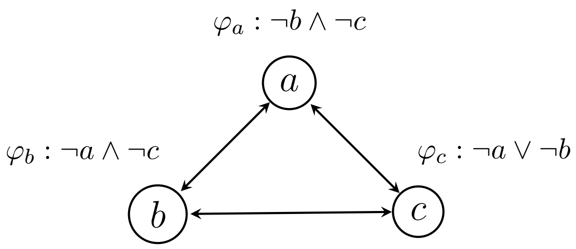

Figure 3.14: ADF used in Example 28

Definition 40. Let $D = (S, L, C)$ be an ADF. D contains a directed cycle if its corresponding graph contains a directed cycle.

Definition 41. Given $D = (S, L, C)$, let T be the set of all support links. The corresponding graph (S, T) is named the reduct of D to supports.

Definition 42. Let $D = (S, L, C)$ be an ADF. D contains a support cycle whenever the reduct of D to supports contains a directed cycle.

Example 29. Let $D = (\{a, b, c, e\}, \{\varphi_a : c \wedge \neg b, \varphi_b : a \wedge \neg c, \varphi_c : b \wedge \neg a \wedge e, \varphi_e : \neg c\})$ be the ADF depicted in Figure 3.15. The reduct of D to supports is depicted in Figure 3.16. It is clear that the graph in Figure 3.16 contains a directed cycle. That is, D contains a support cycle in which a, b and c are supported by c, a and b, respectively.

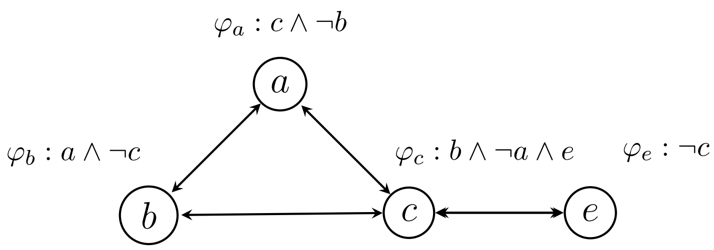

Figure 3.15: ADF used in Example 29

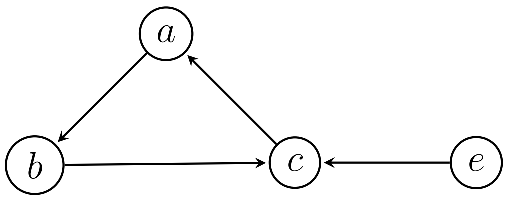

Figure 3.16: Reduct to support of the ADF of Example 29

However, the ADF of Example 30 does not contain any support cycle.

Example 30. Let $D = (\{a, b, c, e\}, \{\varphi_a : c \wedge b,\ \varphi_b : \neg a \wedge \neg c,\ \varphi_c : b \wedge \neg a \wedge e, \varphi_e : \neg c\})$. The reduct of D to supports depicted in Figure 3.17 . It does not contain any directed cycle, therefore, D does not contain any support cycle.

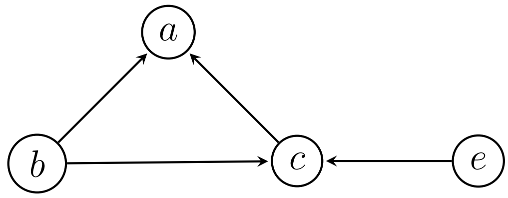

Figure 3.17: Reduct of the ADF of Example 30 to supports

Definition 43. Let $D = (S, L, C)$ be a symmetric BADF. It is named *acyclic support symmetric ADF* (ASSADF for short) if it does not contain any support cycle.

Note that both ASADFs and ASSADFs are subclasses of BADFs. In the definition of ASSADFs it is mentioned explicitly that D is a symmetric BADF, although, in the definition of ASADFs D assumed as a symmetric ADF. Since in ASADFs all links are attacking, there is no dependent link. That is, it is assumed in the definition of ASADFs implicitly that D is a BADF.
By the definition, the ADF D of Example 29 is not an ASSADF, however, the ADF D of Example 30 is an instance of ASSADFs. The natural question is that whether the family of ASSADFs is coherent and relatively grounded. Example 31 illustrates that the family of ASSADFs is not coherent.

Example 31. Let $D = (\{a, b\}, \{\varphi_a : \neg b,\ \varphi_b : a\})$ be an ASSADFs depicted in Figure 3.18. D is an instance of ASSADF with the unique preferred interpretation $v_p = \{a \mapsto u, b \mapsto u\}$ which is not two-valued. That is, D is not semi-coherent. Therefore, D is not coherent.

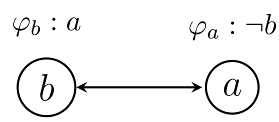

Figure 3.18: ASSADF which is not coherent

In [22] the equivalence of admissibility and conflict-freeness of symmetric AFs is studied. In Example 32 we show that this property, relatively groundedness and some other properties do not carry over from symmetric AFs to ASSADFs.

Example 32. Let $D = (\{a, b, c\}, \{\varphi_a : \neg b \wedge \neg c,\ \varphi_b : \neg a \wedge \neg c,\ \varphi_c : a \vee \neg b\})$ be an ASSADF, depicted in Figure 3.19.

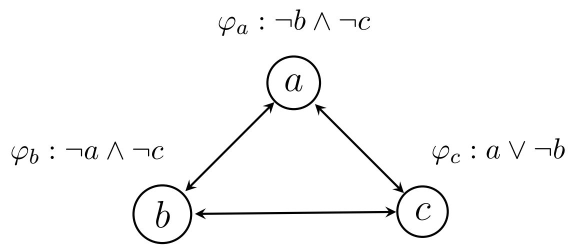

Figure 3.19: ASSADF of Example32

- We show that $v = \{a \mapsto t, b \mapsto f, c \mapsto u\}$ is a conflict-free interpretation for D. It is easy to check that φ_a^v is satisfiable but not irrefutable and φ_b^v is unsatisfiable. For instance, $\varphi_a^v \equiv \neg c$ is satisfiable by $v_1 = \{a \mapsto t, b \mapsto f, c \mapsto f\}$ and it is unsatisfiable by $v_2 = \{a \mapsto t, b \mapsto f, c \mapsto t\}$. In addition, $\varphi_b^v \equiv \bot$ is unsatisfiable by any interpretation. That is, v is conflict-free but not admissible. Therefore, D is an example of an acyclic support symmetric ADF in which conflict-freeness and admissibility are not the same.

- In addition, $v = \{a \mapsto u, b \mapsto f, c \mapsto t\}$ is an admissible interpretation of D which is not complete. Since, $\Gamma_D(v) = \{a \mapsto f, b \mapsto f, c \mapsto t\}$, $v \leq_i \Gamma_D(v)$ and $v \neq \Gamma_D(v)$. That is, admissibility and completeness of ASSADF D are not the same.

- D contains two preferred interpretations $v_1 = \{a \mapsto f, b \mapsto f, c \mapsto t\}$ and $v_2 = \{a \mapsto f, b \mapsto t, c \mapsto f\}$. The meet of these two interpretations is $\{a \mapsto f, b \mapsto u, c \mapsto u\}$. However, the grounded interpretation of D is the trivial interpretation v_u. That is, D is not relatively grounded.

- Moreover, $v_g = \{a \mapsto u, b \mapsto u, c \mapsto u\}$ is a complete interpretation of D which is not preferred. Therefore, the sets of complete and preferred interpretations of D are not equivalent.

- Both preferred interpretations of D are stable models. Although, D is an instance of coherent ASSADF, the set of stable models of D and the grounded interpretation of D are not equivalent.

We investigated via Examples 31 and 32 that the following properties hold in ASSADFs: *prf* $\neq$ *mod*, *prf* $\neq$ *stb*, *cf* $\neq$ *adm*, *adm* $\neq$ *com*, *com* $\neq$ *prf*, *grd* $\neq \bigsqcap$ *prf* and *stb* $\neq \{$*grd*$\}$.

In Theorem 6 it is shown that in every ASSADF each two-valued model is a stable model. That is, each ASSADF is weak-coherent.

Theorem 6. *Every acyclic support symmetric ADF D is weak-coherent.*

Proof. Assume that $D = (S, L, C)$ is an acyclic support symmetric ADF. Further let $v : S \to \{t, f\}$ be a two-valued model of D, and $D^v = (S^v, L^v, C^v)$ be the reduct of D, w be the unique grounded interpretation of D^v, and $\varphi'_s = \varphi_s[s_i/\bot \ : \ v(s_i) = f]$. We show that $v^t = w^t$. Suppose to the contrary that there exists an argument s, s.t. $v(s) = t$ and $w(s) \neq t$. That is, $\varphi'_s \not\equiv \top$. Then φ'_s contains an argument s_1, s.t. s_1 supports s, otherwise by Lemma 6, φ'_s is irrefutable. $v(s_1) = t$ otherwise it is replaced by $\bot$ in φ'_s. In addition, $\varphi^v_{s_1}$ is neither $\top$ nor $\bot$, otherwise it is replaced in φ'_s. Therefore, since supports are acyclic, by the same reason $\varphi'_{s_1} = \varphi_{s_1}[s_i/\bot \ : \ v(s_i) = f]$ contains an argument s_2 which is different from s and s_1 and which is support of s_1. Thus there exists an infinite sequence of $s_1, s_2, \cdots$ s.t. s_{i+1} supports s_i. This is a contradiction with our assumption that symmetric ADFs are finite. □

3.3 Complete ADFs

It is shown is Examples 24 and 32 that the sets of admissible interpretations and complete interpretations are not equivalent in ASADFs and ASSADFs, respectively. In the current section we investigate whether there exists conditions in ADFs under which *adm* = *com*. In general whenever the grounded interpretation of an ADF D is not the trivial interpretation, the set of admissible interpretations of D and the set of complete interpretations of D are not equivalent. Therefore, being the trivial interpretation for grounded interpretation is a necessary condition of an ADFD for $adm(D) = com(D)$. Example 33 is an instance of ADF in which the set of admissible interpretations and the set of complete interpretations are equivalent.

Example 33. Let $D = (\{a, b, c\}, \{\varphi_a : a \leftrightarrow (\neg b \vee c), \varphi_b : (c \wedge b) \vee (\neg c \wedge \neg b), \varphi_c : (\neg a \wedge c) \vee (a \wedge \neg c)\})$ be an ADF, depicted in Figure 3.20. We have

$$\begin{aligned} adm(D) = com(D) = \{ & \{a \mapsto u, b \mapsto u, c \mapsto u\}, \\ & \{a \mapsto f, b \mapsto u, c \mapsto t\}, \\ & \{a \mapsto f, b \mapsto f, c \mapsto t\}, \\ & \{a \mapsto f, b \mapsto t, c \mapsto t\}\}. \end{aligned}$$

Definition 44. An ADF $D = (S, L, C)$ is named a *complete ADF* (CADF for short) whenever for each $a \in S$, $\varphi_a : \psi_a \leftrightarrow a$ and ψ_a does not contain a.

It is shown by Theorem 7 that the set of admissible interpretations and the set of complete interpretations are equivalent for each CADF.

Theorem 7. *$adm(D) = com(D)$ for any CADF $D = (S, L, C)$.*

Proof. We know that each complete interpretation is admissible, hence we need to show under the theorem conditions that each admissible interpretation is complete. Assume

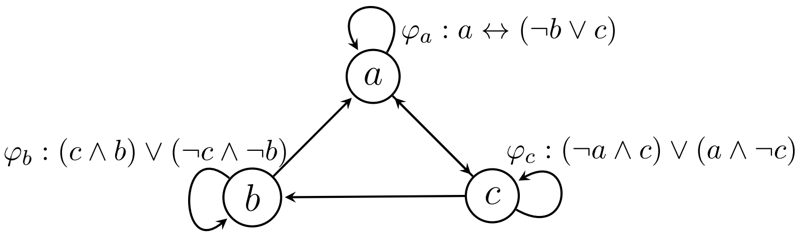

Figure 3.20: ADF used in Example 33

D is a CADF, that is, the acceptance condition of each argument a is in the form of $\varphi_a : \psi_a \leftrightarrow a$ such that ψ_a does not contain a. Towards a contradiction suppose v is an admissible interpretation which is not complete. Therefore, $v \leq_i \Gamma_D(v)$ and $v \neq \Gamma_D(v)$. That is, there exists an $a \in S$ such that $v(a) \neq \Gamma_D(v)(a)$. The only case that we have to investigate is when $v(a)$ is undecided and $\Gamma_D(v)(a)$ is either true or false. Suppose $v(a)$ is undecided and $\Gamma_D(v)(a)$ is true, that is $\varphi_a^v \equiv \top$. Let $w_1 = v|_t^a$ and $w_2 = v|_f^a$. It is clear that $v \leq_i w_1$ and $v \leq_i w_2$. Since, $\varphi_a^v \equiv \top$ and $v \leq_i w_1$ it holds that $\varphi_a^{w_1} \equiv \top$. As $w_1(a) = t$ and $\varphi_a = \psi_a \leftrightarrow a$ it follows that $\psi_a^{w_1} \equiv \top$. Since, $\varphi_a^v \equiv \top$ and $v \leq_i w_2$ it holds that $\varphi_a^{w_2} \equiv \top$. As $w_2(a) = f$ and $\varphi_a = \psi_a \leftrightarrow a$ it follows that $\psi_a^{w_2} \equiv \bot$. That is, $\psi_a^{w_1}$ and $\psi_a^{w_2}$ are not equivalent. This is a contradiction by our assumption that w_1 and w_2 are equivalent in all arguments except on a. Since ψ_a does not contain a, $\psi_a^{w_1}$ and $\psi_a^{w_2}$ have to be equivalent. □

Example 33 is an instance of an ADF which is a CADF, that is, *adm*$(D) =$ *com*(D) for the given ADF D. Now by Examples 34 and 35 we investigate further properties of CADFs, for instance, whether *cf* = *adm* in CADFs.

Example 34. We use again the ADF of Example 33, namely $D = (\{a, b, c\}, \{\varphi_a : a \leftrightarrow (\neg b \vee c), \varphi_b : (c \wedge b) \vee (\neg c \wedge \neg b), \varphi_c : (\neg a \wedge c) \vee (a \wedge \neg c)\}\})$.

- We illustrate that the interpretation, $v = \{a \mapsto u, b \mapsto t, c \mapsto u\}$ is an instance of conflict-free interpretation which is not admissible. To show that v is conflict-free it is enough to show that φ_b^v is satisfiable. It is easy to see that $\varphi_b^v \equiv c$ is indeed satisfiable, for instance by the interpretation $v_1 = \{a \mapsto t, b \mapsto t, c \mapsto t\}$. Recalling the admissible interpretations of D from Example 33, we conclude that *adm*$(D) \neq$ *cf*(D).

- The given D contains two two-valued models, *mod*$(D) = \{\{a \mapsto f, b \mapsto f, c \mapsto t\}, \{a \mapsto f, b \mapsto t, c \mapsto t\}\}$, none of them is a stable model. That is, *mod*$(D) \neq$

stb(D) for a CADF D. Thus D is not weak-coherent and in consequence it is not coherent.

- Moreover, since none of the two-valued models of D is stable, D does not have any stable model. Since the unique grounded interpretation of D is the trivial interpretation, $stb(D) \neq \{grd(D)\}$.
- In addition, in this example the unique grounded interpretation is the trivial interpretation. However, meet of preferred interpretations is $\{a \mapsto f, b \mapsto u, c \mapsto t\}$. Thus, D is not relatively grounded.
- Since the trivial interpretation is a complete interpretation which is not preferred, $prf(D) \neq com(D)$.

Some properties of CADFs are explored by Example 34. For instance, since D of Example 34 is not weak-coherent and coherent, we conclude that the family of CADFs is not weak-coherent and coherent. However, in Example 33, each preferred interpretation is a two-valued interpretation, that is, D is semi-coherent. Example 35 illustrates that CADFs are not semi-coherent in general.

Example 35. Let $D = (\{a, b, c, d\}, \{\varphi_a : a \leftrightarrow \neg c, \varphi_b : b \leftrightarrow a, \varphi_c : c \leftrightarrow d \vee \neg b, \varphi_d : d \leftrightarrow c\})$ be a CADF depicted in Figure 3.21. D contains two preferred interpretations, $v_1 = \{a \mapsto u, b \mapsto u, c \mapsto t, d \mapsto t\}$ and $v_2 = \{a \mapsto t, b \mapsto f, c \mapsto f, d \mapsto u\}$, none of them is a two-valued model. Thus, D is not semi-coherent.

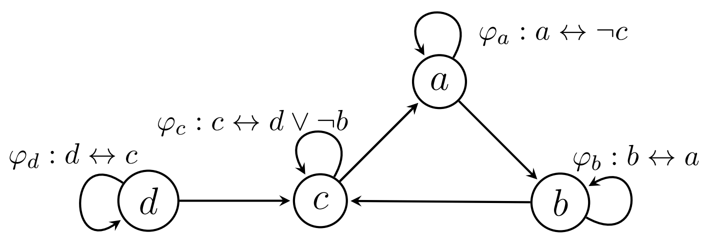

Figure 3.21: ADF used in Example 35

3.4 Summary

This chapter first studied whether the properties which are explained and proven in [22] for symmetric AFs carry over to symmetric ADFs. Then, we have tried to investigate whether some weaker properties hold in symmetric ADFs, its subclasses and CADFs. The results of this study are summarized in Table 3.1 and 3.2. Table 3.1 shows that none of the properties which hold in symmetric AFs carries over to symmetric ADFs and Table 3.2 illustrates the relation between some subclasses of ADFs and some properties.

In these tables in the first horizontal line, different families of AFs and ADFs are written and in the first vertical line different properties are listed, for example, $cf = adm$. As mentioned before, this equivalence means that the set of conflict-free interpretations and the set of admissible interpretations are the same. Whenever there exists a proof that shows that a family of argumentation formalisms has a property is indicated by $\surd$ and when a property does not hold for a family of AFs or ADFs it is marked by – in the table. For instance, each member of the family of symmetric AFs is relatively grounded while by Example 25 we showed that it is not the case for the family of attack symmetric ADFs even for an ASADF with only one $\vee$ connective.

	Symmetric AFs	Symmetric ADFs
$cf = adm$	$\surd$	–
$adm = com$	–	–
$prf = mod$ (semi-coherent)	$\surd$	–
$mod = stb$ (weak-coherent)	$\surd$	–
$com = prf$	–	–
$stb = \{grd\}$	–	–
relatively grounded	$\surd$	–
coherent	$\surd$	–

Table 3.1: Comparison of symmetric AFs and ADFs

	Acyclic ADFs	Symmetric ADFs	ASADFs	ASSADFs	CADFs
$cf = adm$	–	–	–	–	–
$adm = com$	–	–	–	–	$\surd$
$prf = mod$ (semi-coherent)	$\surd$	–	–	–	–
$mod = stb$ (weak-coherent)	$\surd$	–	$\surd$	$\surd$	–
$com = prf$	$\surd$	–	–	–	–
$stb = \{grd\}$	$\surd$	–	–	–	–
relatively grounded	$\surd$	–	–	–	–
coherent	$\surd$	–	–	–	–

Table 3.2: Properties of subclasses of ADFs

CHAPTER 4

Expressiveness and Realizability

There is no shadow of doubt that ADFs are more powerful than AFs. It is shown formally in [17] that the family of ADFs is strictly more expressiveness than the family of AFs. In Chapter 3 we studied some subclasses of ADFs. In this chapter we approach ADFs, AFs, and all of their subclasses as a knowledge representation formalisms. A formalism $\mathcal{F}$ is the set of structures available in a formalism and each element of $\mathcal{F}$ is a knowledge base (kb for short) of that formalism. Recently, the study of expressiveness of different argumentation formalisms has gotten increased attention. Intuitively, a set of sets $\mathbb{S}$ is realizable under a semantics σ in a formalism $\mathcal{F}$ when there exists a framework F in $\mathcal{F}$ s.t. $\mathbb{S} = \sigma(F)$ in which $\sigma(F)$ is the set of all σ extensions (resp. σ interpretations) of F. If such a framework does not exist in a formalism $\mathcal{F}$ then $\mathbb{S}$ is not realizable by the formalism $\mathcal{F}$. Formalism $\mathcal{F}_1$ is called more expressive than formalism $\mathcal{F}_2$ for a semantics σ if each set of sets which is realizable under σ by $\mathcal{F}_2$ is also realizable by $\mathcal{F}_1$. For instance, we would like to study whether there is a relation between the set of all sets of preferred interpretations which are realizable by ADFs and the set of all sets of preferred interpretations which are realizable by attack symmetric ADFs. Expressiveness of different formalisms has been studied in [40, 41, 29]. In addition, some efforts have been done to design and compare different semantics of a formalism. For example, it is shown in [25] that the set of all sets of preferred extensions realizable by AFs is a superset of the set of all sets of naive extensions realizable by AFs. In the following of this chapter in Section 4.1 first we define some preliminary definitions which are used in this work. In Section 4.2 we illustrate expressiveness of symmetric AFs in comparison to AFs and in Section 4.3 we study expressiveness of subclasses of ADFs which are studied in Chapter 3 and compare them with existing results.

4.1 Preliminary definitions

To compare different formalisms the concepts of realizability, signature of a formalism w.r.t. a semantics and expressiveness are defied as follows.

Definition 45. Let $\mathcal{F}$ be a formalism and $\mathbb{S}$ be a set of sets and σ be a semantics. $\mathbb{S}$ is called *realizable* by $\mathcal{F}$ under σ if there exists $kb \in \mathcal{F}$ s.t. $\sigma(kb) = \mathbb{S}$.

Signature of a formalism $\mathcal{F}$ collects all sets which are realizable by $\mathcal{F}$ under a semantics σ, which is defined formally as follows.

Definition 46. The *signature* $\Sigma^{\sigma}_{\mathcal{F}}$ of a formalism $\mathcal{F}$ w.r.t. semantics σ is defined as:

$$\Sigma^{\sigma}_{\mathcal{F}} = \{\sigma(kb) \mid kb \in \mathcal{F}\}$$

in which kb is a knowledge base of that formalism.

Definition 47. Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two formalisms. We say that $\mathcal{F}_1$ is *strictly more expressive* than $\mathcal{F}_2$ for σ, whenever $\Sigma^{\sigma}_{\mathcal{F}_2} \subset \Sigma^{\sigma}_{\mathcal{F}_1}$ on a fixed semantics σ.

Definition 48. Let $\mathcal{F}$ be a formalism and σ_1 and σ_2 be two semantics. Signature of a formalism $\mathcal{F}$ w.r.t. σ_1 and signature of $\mathcal{F}$ w.r.t. σ_2 are called incomparable whenever $\Sigma^{\sigma_1}_{\mathcal{F}} \nsubseteq \Sigma^{\sigma_2}_{\mathcal{F}}$ and $\Sigma^{\sigma_2}_{\mathcal{F}} \nsubseteq \Sigma^{\sigma_1}_{\mathcal{F}}$. It is denoted by $\Sigma^{\sigma_1}_{\mathcal{F}} \not\sim \Sigma^{\sigma_2}_{\mathcal{F}}$.

In this section whenever we are working with AFs we restrict ourselves to extension notion and whenever we are working with ADFs we use three-valued interpretations. When we want to compare expressiveness between AFs and ADFs we work with three-valued interpretations. That is, whenever we want to compare the signature of an AF with the signature of an ADF we work with the associated interpretations of extensions of such an AF, according to Definition 24.

4.2 Expressiveness of AFs

In this section we restrict ourselves to AFs. In [25] different semantics of AFs are compared comprehensively. The formal definition of realizability and the signature in AFs is as follows.

Definition 49. Let A be a set of arguments of an AF F and $\mathbb{S} \subseteq 2^A$. $\mathbb{S}$ is named an *extension-set* if $|A|$ is finite.

Definition 50. The extension-set $\mathbb{S}$ is named realizable under σ (σ-realizable) if there is an AF F s.t. $\sigma(F) = \mathbb{S}$.
The signature Σ^{σ}_{AF} w.r.t. a semantics σ is,

$$\Sigma^{\sigma}_{AF} = \{\sigma(F) \mid F \ is \ an \ AF\}.$$

In Definition 50 σ is a function which maps each F to a set of σ-extensions. More precisely, Σ^{σ}_{AF} contains all sets of σ-extensions realizable by AFs. As mentioned before in realizability we face the question whether a set of extensions, $\mathbb{S}$ is contained in Σ^{σ}_{AF}.

For all semantics σ each element $\mathbb{S} \in \Sigma^{\sigma}_{AF}$ is an extension-set whenever we are dealing with finite AFs. The extension-set $\mathbb{S}$ given in Example 36 is not an element of Σ^{cf}_{AF}.

Example 36. Let $\mathbb{S} = \{\{a, b\}, \{a, c\}, \{b, c\}\}$ and $\sigma = cf$. Since for each AF F, $cf(F)$ always contains the empty set, there is no AF F s.t. $\mathbb{S} = cf(F)$. That is, $\mathbb{S} \notin \Sigma^{cf}_{AF}$.

Intuitively, the set $\mathbb{S}$ in Example 36 is not realizable by any argumentation framework under conflict-freeness. Example 37 is an instance of extension-set which is is realizable in AFs under $\sigma = prf$.

Example 37. Let $\mathbb{S} = \{\{a, b, e\}, \{c, d, f\}\}$ and $\sigma = prf$. We claim that the extension-set $\mathbb{S}$ is an element of Σ^{prf}_{AF}. That is, we show that there exists an AF F such that $prf(F) = \mathbb{S}$. One witness of our claim is: $F = (\{a, b, c, d, e, f\}, \{(a, b), (b, a), (c, d), (d, c), (c, e), (e, c), (d, e), (e, d), (e, f), (f, e), (a, d), (d, c), (a, f), (b, f)\})$, depicted in Figure 4.3.

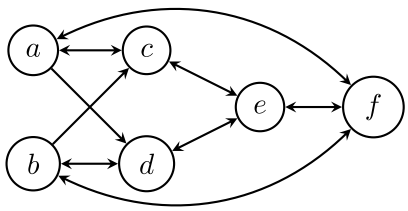

Figure 4.1: AF used in Example 37

Definition 51. Let $\mathbb{S}$ be an extension set and S_1 and S_2 be two elements of $\mathbb{S}$.

- S_1 and S_2 are called incomparable whenever neither $S_1 \subseteq S_2$ nor $S_2 \subseteq S_1$. It is denoted by $S_1 \not\sim S_2$.
- $\mathbb{S}$ is named incomparable if all elements of $\mathbb{S}$ are pairwise incomparable.

The extension-set $\mathbb{S} = \{\{a, b, e\}, \{c, d, f\}\}$ which is defined in the Example 37 is incomparable. However, the extension-set $\mathbb{S} = \{\{a, b, e\}, \{c, d, f\}, \{c, d, f, g\}\}$ is not incomparable, the second element of $\mathbb{S}$ is a subset of the third element. [25] illustrates relations among signatures of different semantics, some of them are in Theorem 8.

Theorem 8. *[25] The following relations hold:*

- $\Sigma^{nai}_{AF} \subsetneq \Sigma^{stb}_{AF} \setminus \{\emptyset\} \subsetneq \Sigma^{prf}_{AF}$,
- $\Sigma^{cf}_{AF} \subsetneq \Sigma^{adm}_{AF} \subsetneq \Sigma^{com}_{AF}$.

Theorem 8 says that whatever an extension-set is *nai*-realizable then it is *prf*-realizable but not vice versa. That is there exists a set $\mathbb{S}$ which is *prf*-realizable but not *nai*-realizable. For example, the extension-set $\mathbb{S} = \{\{a_1, b_2, b_3\}, \{a_2, b_1, b_3\}, \{a_3, b_1, b_2\}\}$ explained in [25] and depicted in Figure 4.2 is *prf*-realizable in AFs. Since b_2 and b_3 appear in the first element of $\mathbb{S}$ there is no conflict between them. With the same reason, because of the second element of $\mathbb{S}$ there is no conflict between b_1 and b_3 and because of the third element of $\mathbb{S}$ there is not any conflict between b_1 and b_2. Thus, $\{b_1, b_2, b_3\}$ is a conflict-free set. Since it is a maximal conflict-free set, $\{b_1, b_2, b_3\}$ is a naive extension. Since $\mathbb{S}$ does not contain $\{b_1, b_2, b_3\}$, it is not *nai*-realizable in AFs.

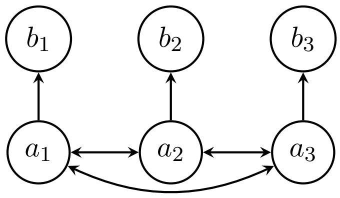

Figure 4.2: AF used in a proof pf part 1 of Theorem 8

The expressiveness among AFs, BADFs and ADFs under different semantics is illustrated in [41, 29]. It expresses in the following theorem:

Theorem 9. *[17] For $\sigma \in \{adm, com, prf, mod\}$, we have*

$$\Sigma^{\sigma}_{AF} \subsetneq \Sigma^{\sigma}_{BADF} \subsetneq \Sigma^{\sigma}_{ADF}$$

for the stable model, stb,

$$\Sigma^{stb}_{AF} \subsetneq \Sigma^{stb}_{BADF} = \Sigma^{stb}_{ADF}$$

4.2.1 Expressiveness of Symmetric AFs

In [22] symmetric AFs (SYMAFs for short) are studied as a proper subclass of AFs. In the following we study relations among various signatures of SYMAFs and expressiveness of SYMAFs in comparison to AFs.

Theorem 10. *Following relations hold among semantics of SYMAFs:*

- $\Sigma^{adm}_{SYMAF} = \Sigma^{cf}_{SYMAF}$,
- $\Sigma^{prf}_{SYMAF} = \Sigma^{stb}_{SYMAF} = \Sigma^{nai}_{SYMAF}$,
- $\Sigma^{prf}_{SYMAF} \not\sim \Sigma^{com}_{SYMAF}$,
- $\Sigma^{adm}_{SYMAF} \not\sim \Sigma^{com}_{SYMAF}$,
- $(\Sigma^{com}_{SYMAF} \cap \Sigma^{prf}_{SYMAF}) \setminus \Sigma^{adm}_{SYMAF} \neq \emptyset$,

- $(\Sigma^{com}_{SYMAF} \cap \Sigma^{adm}_{SYMAF}) \setminus \Sigma^{prf}_{SYMAF} \neq \emptyset$.

Proof.
- Let F be an arbitrary SYMAF. By Proposition 3, $cf(F) = adm(F)$ then $\Sigma^{adm}_{\text{SYMAF}} = \Sigma^{cf}_{\text{SYMAF}}$.
- By Proposition 3 $prf(F) = stb(F) = nai(F)$ for each SYMAF F. Then, $\Sigma^{prf}_{\text{SYMAF}} = \Sigma^{stb}_{\text{SYMAF}} = \Sigma^{nai}_{\text{SYMAF}}$.
- To show $\Sigma^{prf}_{\text{SYMAF}} \not\sim \Sigma^{com}_{\text{SYMAF}}$ we check whether $\Sigma^{prf}_{\text{SYMAF}} \not\subseteq \Sigma^{com}_{\text{SYMAF}}$ and $\Sigma^{com}_{\text{SYMAF}} \not\subseteq \Sigma^{prf}_{\text{SYMAF}}$.
  - To show $\Sigma^{prf}_{\text{SYMAF}} \not\subseteq \Sigma^{com}_{\text{SYMAF}}$ it is enough to show that there exists $\mathbb{S}$ s.t. $\mathbb{S} \in \Sigma^{prf}_{\text{SYMAF}}$ and $\mathbb{S} \notin \Sigma^{com}_{\text{SYMAF}}$. Let $\mathbb{S} = \{\{a\}, \{b\}\}$ be an extension-set. Let $F = (\{a, b\}, \{(a, b), (b, a)\})$. F is a SYMAF s.t. $\mathbb{S} = prf(F)$. That is, $\mathbb{S} \in \Sigma^{prf}_{\text{SYMAF}}$. Suppose to the contrary that $\mathbb{S} \in \Sigma^{com}_{\text{SYMAF}}$. That is, there exists a SYMAF F' s.t. $\mathbb{S} = com(F')$. It is proven in [25] that for each $\mathbb{S} \in \Sigma^{com}_{AF}$ the intersection of all elements of $\mathbb{S}$ is an element of $\mathbb{S}$. However, the intersection of elements of $\mathbb{S}$ is: $\{a\} \cap \{b\} = \emptyset$ and $\emptyset \notin \mathbb{S}$. Therefore, $\mathbb{S}$ is not *com*-realizable in AF. Then it is not *com*-realizable in SYMAF, too. Hence, $\Sigma^{prf}_{\text{SYMAF}} \not\subseteq \Sigma^{com}_{\text{SYMAF}}$.
  - For $\Sigma^{com}_{\text{SYMAF}} \not\subseteq \Sigma^{prf}_{\text{SYMAF}}$ we show that there exists an extension-set $\mathbb{S}$ s.t. $\mathbb{S} \in \Sigma^{com}_{\text{SYMAF}}$ and $\mathbb{S} \notin \Sigma^{prf}_{\text{SYMAF}}$. Let $\mathbb{S} = \{\{a\}, \{b\}, \emptyset\}$ be an extension-set. Let $F = (\{a, b\}, \{(a, b), (b, a)\})$. F is a SYMAFs s.t. $\mathbb{S} = com(F)$. That is, $\mathbb{S} \in \Sigma^{com}_{\text{SYMAF}}$. Suppose to the contrary that $\mathbb{S} \in \Sigma^{prf}_{\text{SYMAF}}$. That is, there exists SYMAFs F' s.t. $\mathbb{S} = prf(F')$. It is proven in [25] if $\mathbb{S} \in \Sigma^{prf}_{AF}$ then $\mathbb{S}$ is incomparable. Since $\emptyset \subseteq \{a\}$ and $\emptyset \subseteq \{b\}$, $\mathbb{S}$ is not incomparable. Then, $\Sigma^{com}_{\text{SYMAF}} \not\subseteq \Sigma^{prf}_{\text{SYMAF}}$.
- To prove $\Sigma^{adm}_{\text{SYMAF}} \not\sim \Sigma^{com}_{\text{SYMAF}}$, first we show that $\Sigma^{adm}_{\text{SYMAF}} \not\subseteq \Sigma^{com}_{\text{SYMAF}}$ and then $\Sigma^{com}_{\text{SYMAF}} \not\subseteq \Sigma^{adm}_{\text{SYMAF}}$.
  - To show $\Sigma^{adm}_{\text{SYMAF}} \not\subseteq \Sigma^{com}_{\text{SYMAF}}$ let $\mathbb{S} = \{\emptyset, \{a\}, \{b\}, \{c\}, \{b, c\}\}$. A witness of *adm*-realizability of $\mathbb{S}$ in SYMAFs is $F = (\{a, b, c\}, \{(a, c), (c, a), (a, b), (b, a)\})$. Then $\mathbb{S} \in \Sigma^{adm}_{\text{SYMAF}}$. We show that $\mathbb{S}$ is not *com*-realizable in SYMAFs. Suppose to the contrary that there exists a SYMAF $F' = (A', L')$ s.t. $com(F') = \mathbb{S}$. It is clear that A' is a superset of A. Since F' is a SYMAF each argument of A' appears in at least a preferred extension and in a complete extension then $A' \subseteq A$. Therefore, $A = A'$. In the following it is shown that under all different possibilities of choosing links of L', $com(F') \neq \mathbb{S}$:
    * any of these arguments cannot appear as an isolated argument in F', otherwise, the grounded extension which is also a complete extension of F' is not the empty set.
    * Because of $\{b, c\} \in \mathbb{S}$ there is no link between b and c in L'. Then, the only possible way of defining relations in F' is $L' = \{(a, b), (b, a), (a, c), (c, a)\}$. Hence, $com(F') = \{\emptyset, \{a\}, \{b, c\}\}$ and $\mathbb{S} \neq com(F')$.

Then, $\mathbb{S} \notin \Sigma^{com}_{\text{SYMAF}}$. That is, $\mathbb{S}$ is not *com*-realizable in SYMAFs.

– To show $\Sigma^{com}_{\text{SYMAF}} \not\subseteq \Sigma^{adm}_{\text{SYMAF}}$ let $\mathbb{S} = \{\{a\}, \{a, b\}, \{a, c\}\}$. Then, $\mathbb{S}$ is *com*-realizable in SYMAFs and the witness of $\mathbb{S}$ *com*-realizability in SYMAFs is $F = (\{a, b, c\}, \{(b, c), (c, b)\})$. We know the empty set is an admissible extension for each AF F. That is, $\emptyset \in adm(F)$ for each AF F. However, $\mathbb{S}$ does not contain the empty set. That is, $\mathbb{S}$ is not *adm*-realizable in SYMAFs.

- To show that $(\Sigma^{com}_{\text{SYMAF}} \cap \Sigma^{prf}_{\text{SYMAF}}) \setminus \Sigma^{adm}_{\text{SYMAF}} \neq \emptyset$, let $\mathbb{S} = \{\{a, b\}\}$. A witness of complete and preferred realizability of $\mathbb{S}$ in SYMAFs is $F = (\{a, b\}, \emptyset)$. Since $\emptyset$ is an admissible extension for any SYMAF F and $\mathbb{S}$ does not contain it, $\mathbb{S}$ is not *adm*-realizable in SYMAFs. This is a desired result.

- To investigate that whether $(\Sigma^{com}_{\text{SYMAF}} \cap \Sigma^{adm}_{\text{SYMAF}}) \setminus \Sigma^{prf}_{\text{SYMAF}} \neq \emptyset$, let $\mathbb{S} = \{\emptyset, \{a\}, \{b\}\}$. A witness of complete and admissible realizability of $\mathbb{S}$ in SYMAFs is $F = (\{a, b\}, \{(a, b), (b, a)\})$. Since $\mathbb{S}$ contains comparable elements, it is not *prf*-realizable in SYMAFs, which is the desired result.

□

Theorem 11 shows that AFs are strictly more expressive than SYMAFs in some of the semantics and are equivalent for $\sigma \in \{cf, nai, grd\}$:

Theorem 11. *For $\sigma \in \{cf, nai, grd\}$ the following relation holds:*

$$\Sigma^{\sigma}_{SYMAF} = \Sigma^{\sigma}_{AF}$$

and for $\sigma = \{adm, prf, stb, com\}$:

$$\Sigma^{\sigma}_{SYMAF} \subsetneq \Sigma^{\sigma}_{AF}$$

Proof. • Since each SYMAF is an AF, it is clear that $\Sigma^{\sigma}_{\text{SYMAF}} \subseteq \Sigma^{\sigma}_{AF}$ for $\sigma = \{adm, prf, cf, stb, nai, com, grd\}$. To show that $\Sigma^{\sigma}_{\text{SYMAF}} = \Sigma^{\sigma}_{AF}$ for $\sigma \in \{cf, nai, grd\}$ we illustrate that $\Sigma^{\sigma}_{AF} \subseteq \Sigma^{\sigma}_{\text{SYMAF}}$.

– To show that $\Sigma^{cf}_{AF} \subseteq \Sigma^{cf}_{\text{SYMAF}}$ let $\mathbb{A}$ be an arbitrary extension-set s.t. $\mathbb{A} \in \Sigma^{cf}_{AF}$. We show that $\mathbb{A} \in \Sigma^{cf}_{\text{SYMAF}}$. Suppose that $F = (A, L)$ is an AF s.t. $cf(F) = \mathbb{A}$. Let B be the set of all arguments of A with self-attack. Construct $F' = (A', L')$ as follows:

* $A' = A \setminus B$,
* $L' = K \cup \{(b, a) \mid (a, b) \in K\}$ in which $K = \{(a, b) \mid a, b \in A', (a, b) \in L\}$.

It is easy to see that $cf(F) = cf(F')$. The idea of F' is to remove all self-attack arguments of F and to make attacks among all other arguments of F symmetric. Therefore, there is a conflict between arguments in F' if and only if there is a conflict between arguments in F. That is, $\Sigma^{cf}_{\text{SYMAF}} = \Sigma^{cf}_{AF}$.

- To show that $\Sigma^{nai}_{AF} \subseteq \Sigma^{nai}_{\text{SYMAF}}$ let $\mathbb{A}$ be an arbitrary extension-set s.t. $\mathbb{A} \in \Sigma^{nai}_{AF}$. Then, there exists an AF $F = (A, L)$ s.t. $nai(F) = \mathbb{A}$. By the definition, the conflict-free subsets of A which are maximal w.r.t. $\subsetneq$ are naive extensions of A. By the previous part $\Sigma^{cf}_{\text{SYMAF}} = \Sigma^{cf}_{AF}$ hence, $\Sigma^{nai}_{\text{SYMAF}} = \Sigma^{nai}_{AF}$.
- To show that $\Sigma^{grd}_{AF} \subseteq \Sigma^{grd}_{\text{SYMAF}}$ let $\mathbb{A}$ be an arbitrary extension-set s.t. $\mathbb{A} \in \Sigma^{grd}_{AF}$. We investigate that $\mathbb{A} \in \Sigma^{grd}_{\text{SYMAF}}$ by dividing the problem in the following two cases:
  - Suppose the unique grounded extension $\mathbb{A}$ is the empty set. Any SYMAFs without any isolated argument is a witness of *grd*-realizability of $\mathbb{A}$ in SYMAFs.
  - Suppose the unique grounded extension $\mathbb{A}$ is not empty, i.e. $\mathbb{A} = \{S\}$. Any SYMAF in which the set of isolated arguments is equivalent with $\mathbb{A}$ is a witness of *grd*-realizability of $\mathbb{A}$ in SYMAFs. For instance $\mathbb{A}$ is *grd*-realizable in SYMAFs by $F = (S, \emptyset)$.

- To show that $\Sigma^{\sigma}_{\text{SYMAF}}$ is a proper subset of Σ^{σ}_{AF} for $\sigma = \{adm, com\}$ we show that there exists an extension-set $\mathbb{S}$ which is σ-realizable in AFs but not σ-realizable in SYMAFs. With the help of other theorems we show that $\Sigma^{\sigma}_{\text{SYMAF}} \subsetneq \Sigma^{\sigma}_{AF}$ for $\sigma \in \{prf, stb\}$.

  - To investigate that whether $\Sigma^{adm}_{\text{SYMAF}} \subsetneq \Sigma^{adm}_{AF}$ let $\mathbb{S} = \{\emptyset, \{a\}, \{a, c\}\}$. A witness of *adm*-realizability of $\mathbb{S}$ in AFs is $F = (\{a, b, c\}, \{(a, b), (b, c)\})$. Toward a contradiction assume that there exists a SYMAF $F' = (A', L')$ s.t. $adm(F') = \mathbb{S}$. Since F' is a SYMAF then each argument of A' appears in at least one admissible extension. That is, A' has to be $\{a, c\}$. In the following we compute $adm(F')$ for different possibilities of defining L':
    - If $L' = \emptyset$ then a and c are isolated arguments. Therefore, $adm(F') = \{\emptyset, \{a\}, \{c\}, \{a, c\}\}$.
    - If $L' = \{(a, c), (c, a)\}$ then $adm(F') = \{\emptyset, \{a\}, \{c\}\}$.

    In both cases $adm(F') \neq \mathbb{S}$. Hence, $\mathbb{S}$ is not *adm*-realizable in $\Sigma^{adm}_{\text{SYMAF}}$. That is, Σ^{adm}_{AF} is a strict supperset of $\Sigma^{adm}_{\text{SYMAF}}$.
  - To show $\Sigma^{prf}_{\text{SYMAF}} \subsetneq \Sigma^{prf}_{AF}$. It is proven in the first part of this theorem that $\Sigma^{nai}_{\text{SYMAF}} = \Sigma^{nai}_{AF}$. It is also shown in Theorem 10 that $\Sigma^{prf}_{\text{SYMAF}} = \Sigma^{nai}_{\text{SYMAF}}$. Hence, $\Sigma^{prf}_{\text{SYMAF}} = \Sigma^{nai}_{\text{SYMAF}} = \Sigma^{nai}_{AF}$. In addition, by Theorem 8 we have $\Sigma^{nai}_{AF} \subsetneq \Sigma^{prf}_{AF}$. Thus, $\Sigma^{prf}_{\text{SYMAF}} = \Sigma^{nai}_{\text{SYMAF}} = \Sigma^{nai}_{AF} \subsetneq \Sigma^{prf}_{AF}$. That is, $\Sigma^{prf}_{\text{SYMAF}} \subsetneq \Sigma^{prf}_{AF}$.
  - We prove $\Sigma^{stb}_{\text{SYMAF}} \subsetneq \Sigma^{stb}_{AF}$ with the same proof method as we used in previous part. It is proven in Theorem 10 that $\Sigma^{stb}_{\text{SYMAF}} = \Sigma^{nai}_{\text{SYMAF}}$. By the first part of the current theorem we have $\Sigma^{nai}_{\text{SYMAF}} = \Sigma^{nai}_{AF}$. Moreover, by Theorem 8 we have $\Sigma^{nai}_{AF} \subsetneq \Sigma^{stb}_{AF}$. Thus, $\Sigma^{stb}_{\text{SYMAF}} = \Sigma^{nai}_{\text{SYMAF}} = \Sigma^{nai}_{AF} \subsetneq \Sigma^{stb}_{AF}$. Hence, $\Sigma^{stb}_{\text{SYMAF}}$ is a strict superset of Σ^{stb}_{AF}.

- To show that $\Sigma^{com}_{\text{SYMAF}} \subsetneq \Sigma^{com}_{AF}$ let $\mathbb{S} = \{\emptyset, \{b, e\}\}$. We show that $\mathbb{S}$ is *com*-realizable in AFs. A witness of this claim could be an AF $F = (\{a, b, c, e\}, \{(a, b), (b, c), (c, a), (a, e), (e, a)\})$. Suppose to the contrary that there exists a SYMAF $F' = (S', L')$ which is a witness of *com*-realizability of $\mathbb{S}$. Since $\mathbb{S}$ contains the empty set then it has to be the grounded interpretation of F'. Therefore, F' does not contain any isolated argument. That is b and e are not isolated arguments in F'. Because of $\{b, e\} \in \mathbb{S}$ there is no link between b and e. Therefore, there exists at least an additional argument $s \in S'$. Since F' is a SYMAF then s appears in a preferred extension, P. Each preferred extension is a complete extension. That is, $P \in com(F')$, however, P is not in $\mathbb{S}$. Thus, $\mathbb{S}$ is not *com*-realizable in SYMAF.

Therefore, $\Sigma^{\sigma}_{\text{SYMAF}} \subsetneq \Sigma^{\sigma}_{AF}$, for $\sigma = \{adm, prf, stb, com\}$.

□

In the beginning of Section 4.2.1 we investigated by Theorem 10 the relations among different semantics of signatures of SYMAFs and by Theorem 11 it is shown that the signature of AFs is more expressive than the signature of SYMAFs, in semantics σ, $\sigma \in \{adm, prf, stb, com\}$. In the following of this section we show some general results about σ-realizability in SYMAFs based on the cardinality of an extension-set. For instance, it is proven by Theorem 11 that Σ^{σ}_{AF} is a strict superset of Σ^{σ}_{SYMAF} for $\sigma \in \{prf, stb\}$. However, it will be shown in Propositions 7 and 8 that whenever the cardinality of an extension-set is less than or equal to two and it is σ-realizable in AFs for $\sigma \in \{prf, stb\}$, it is σ-realizable in SYMAFs.

Proposition 7. *Let $\mathbb{S}$ be an extension-set s.t. $|\mathbb{S}| = 1$. $\mathbb{S}$ is σ-realizable in SYMAFs for $\sigma \in \{prf, stb, com\}$.*

Proof. Suppose that $\mathbb{S}$ is an extension-set s.t. $|\mathbb{S}| = 1$, $\mathbb{S} = \{K\}$. We show that $\mathbb{S}$ is σ-realizable in SYMAFs for $\sigma \in \{prf, stb, com\}$. We claim that $F = (A, L)$ in which $A = K$ and $L = \emptyset$ is a witness of σ-realizability of $\mathbb{S}$ in SYMAFs for $\sigma \in \{prf, stb, com\}$. F is a SYMAF s.t. $stb(F) = prf(F) = com(F) = \{A\}$. That is, $\sigma(F) = \mathbb{S}$ for $\sigma \in \{prf, stb, com\}$. That is, $\mathbb{S}$ is σ-realizable in SYMAF for $\sigma \in \{prf, stb, com\}$. □

Corollary 3. *If $|\mathbb{S}| = 1$ and $\mathbb{S}$ is σ-realizable in AFs for $\sigma \in \{adm, cf, nai, prf, stb, com, grd\}$ then $\mathbb{S}$ is σ-realizable in SYMAFs as well.*

Proof. Let $\mathbb{S}$ be σ-realizable in AFs and $|\mathbb{S}| = 1$:

- It is proven in Proposition 7 that each extension-set s.t. $|\mathbb{S}| = 1$ is σ-realizable in SYMAFs for $\sigma \in \{prf, stb, com\}$. Therefore, if $\mathbb{S}$ is an extension-set which is σ-realizable in AFs and $|\mathbb{S}| = 1$ then it is σ-realizable in SYMAFs for $\sigma \in \{prf, stb, com\}$.

- If $|\mathbb{S}| = 1$ and $\mathbb{S}$ is *adm*-realizable in AFs then $\mathbb{S} = \{\emptyset\}$. Let $F = (A, L)$ in which $A = \emptyset$ and $L = \emptyset$. Then F is a SYMAF and $adm(F) = \{\emptyset\}$. That is, $\mathbb{S}$ is *adm*-realizable in SYMAFs.

- It is proven in Theorem 11 that an extension is σ-realizable in AFs for $\sigma \in \{cf, nai, grd\}$ if and only if it is σ-realizable in SYMAFs for $\sigma \in \{cf, nai, grd\}$ which includes this case, too.

□

Proposition 8. *Let $\mathbb{S}$ be an extension-set s.t. $|\mathbb{S}| = 2$ and $\mathbb{S}$ be σ-realizable in AFs for $\sigma = \{prf, stb\}$. $\mathbb{S}$ is σ-realizable in SYMAFs for $\sigma = \{prf, stb\}$.*

Proof. Suppose $\mathbb{S}$ is σ-realizable in AFs for $\sigma = \{prf, stb\}$ and $|\mathbb{S}| = 2$. That is, there exists an AF $F = (S, L)$ s.t. $\sigma(F) = \mathbb{S}$. We illustrate that there exists a SYMAF $F' = (S', L')$ s.t. $\sigma(F') = \mathbb{S}$. Since $|\mathbb{S}| = 2$ and $\mathbb{S}$ is σ-realizable in AFs for $\sigma = \{prf, stb\}$ there are two non-empty incomparable sets S_1 and S_2 s.t. $S_1, S_2 \in \mathbb{S}$. Construct $F' = (S', L')$ as follows:

- $S' = S_1 \cup S_2$,
- $L' = \{(a, b), (b, a) \mid a \in (S_1 \setminus S_2), b \in (S_2 \setminus S_1)\}$.

The idea of F' is to consider each argument in the intersection of S_1 and S_2 as an isolated argument and to define a symmetric relation between other arguments of S_1 and S_2 s.t. there is no relation between arguments of S_1 (resp. S_2). It is easy to see that $stb(F') = \mathbb{S}$. Then, $\mathbb{S}$ is *stb*-realizable in SYMAFs. By Proposition 3 in each SYMAF F, $prf(F) = stb(F)$. Then, $\mathbb{S}$ is σ-realizable in SYMAF for $\sigma \in \{prf, stb\}$. □

We know by Theorem 11 that Σ^{σ}_{AF} is a strict superset of $\Sigma^{\sigma}_{\text{SYMAF}}$ for $\sigma \in \{prf, stb\}$. That is, there exists $\mathbb{S}$ s.t. $\mathbb{S} \in \Sigma^{\sigma}_{AF}$ and $\mathbb{S} \notin \Sigma^{\sigma}_{\text{SYMAF}}$. By Propositions 7 and 8 we have shown that whenever $|\mathbb{S}| \leq 2$, $\mathbb{S}$ is σ-realizable in AFs if and only if it is σ-realizable in SYMAFs for $\sigma \in \{prf, stb\}$. The natural question is that whether there exist conditions under which $\mathbb{S} \in \Sigma^{\sigma}_{\text{SYMAF}}$ whenever $\mathbb{S} \in \Sigma^{\sigma}_{AF}$ and $|\mathbb{S}| > 2$ for $\sigma \in \{prf, stb\}$. In Proposition 9 we investigate that whenever the intersection of each two elements of $\mathbb{S}$ is the same, $\mathbb{S} \in \Sigma^{\sigma}_{\text{SYMAF}}$ for $\sigma \in \{prf, stb\}$.

Proposition 9. *Let $\mathbb{S}$ be an extension-set s.t. $|\mathbb{S}| > 2$ and $\mathbb{S}$ be σ-realizable in AFs for $\sigma = \{prf, stb\}$ and there is some set K s.t. for every $S_i, S_j \in \mathbb{S}$, $S_i \cap S_j = K$ for $i \neq j$. $\mathbb{S}$ is σ-realizable in SYMAFs for $\sigma = \{prf, stb\}$.*

Proof. Let $\mathbb{S}$ be σ-realizable in AFs s.t. $|\mathbb{S}| > 2$ for $\sigma = \{prf, stb\}$. Then, there exists a finite AF F s.t. $\sigma(F) = \mathbb{S}$. Then there exists a finite number $n > 2$ s.t. $|\mathbb{S}| = n$. That is, there exists n different non-empty incomparable sets, $S_1, \ldots, S_n$ s.t. $S_i \in \mathbb{S}$ for $1 \leq i \leq n$.

Let s^i be an element of S_i. By proposition assumption the intersection of each two elements of $\mathbb{S}$ is a fixed set K. We investigate that $F' = (S', L')$ constructed as follows is a witness of *prf*-realizability of $\mathbb{S}$ in SYMAFs. That is, $\mathit{prf}(F') = \mathbb{S}$.

- $S' = \bigcup_{1 \leq i \leq n} S_i$,
- $L' = \bigcup_{1 \leq l \leq n} \bigcup_{s^l \in S_l \setminus K} \bigcup_{m \neq l} \bigcup_{s^m \in S_m \setminus K} (s^l, s^m)$.

The idea of F' is to keep each element of K as an isolated argument and to classify other arguments of S' in to n different subsets s.t. there is no relation among elements of each set and each element of each set is in conflict with all elements of the other sets. Then, it is clear that $\mathit{stb}(F') = \mathbb{S}$. Again by Proposition 3 we conclude that $\mathit{stb}(F') = \mathit{prf}(F') = \mathbb{S}$. Therefore, $\mathbb{S}$ is σ-realizable in SYMAFs for $\sigma \in \{\mathit{prf}, \mathit{stb}\}$. □

By Propositions 7, 8 and 9 the conditions under which an extension-set which is σ-realizable in AFs is also σ-realizable in SYMAFs for $\sigma \in \{\mathit{prf}, \mathit{stb}\}$ are studied. In the following we investigate whether the result of Proposition 8 carries over to $\sigma \in \{\mathit{adm}, \mathit{com}\}$. That is, whether for each extension-set s.t. $|\mathbb{S}| = 2$ and $\mathbb{S} \in \Sigma^{\sigma}_{AF}$, we can conclude that $\mathbb{S} \in \Sigma^{\sigma}_{\text{SYMAF}}$. The results are illustrated in Proposition 10 and in Example 38. Proposition 10 shows that in general whenever $|\mathbb{S}| = 2$, $\mathbb{S} \notin \Sigma^{\mathit{com}}_{\text{SYMAF}}$. Since Lemma 9 is used in the proof of Proposition 10, first we show Lemma 9.

Lemma 9. *Let $F = (A, L)$ be a SYMAF. If F contains arguments which are not isolated then $\mathit{prf}(F)$ contains at least two incomparable preferred extensions.*

Proof. Suppose that $F = (A, L)$ is a SYMAF s.t. F contains arguments which are not isolated. Then, at least there are two arguments $a_1, a_2 \in A$ s.t. $(a_1, a_2) \in L$ and $(a_2, a_1) \in L$. Since F is a SYMAF each of its arguments occurs at least in a preferred extension. Since there is a symmetric link between a_1 and a_2 then, there exists at least two preferred extensions A_1 and A_2 s.t. $a_1 \in A_1$ and $a_2 \in A_2$. It is clear that $A_1 \not\sim A_2$. Then $\mathit{prf}(F)$ contains at least two incomparable preferred extension. □

Proposition 10. *Let $\mathbb{S}$ be an extension-set s.t. $|\mathbb{S}| = 2$. Then $\mathbb{S}$ is not com-realizable in SYMAFs.*

Proof. Towards a contradiction assume that $\mathbb{S}$ is *com*-realizable in SYMAFs. That is, there exists a SYMAF $F = (A, L)$ s.t. $\mathit{com}(F) = \mathbb{S}$. Let S_1 and S_2 be two elements of $\mathbb{S}$. Since $|\mathbb{S}| = 2$, $A \neq \emptyset$. Let $B = S_1 \cap S_2$. That is, B is the grounded extension of F. There two different possibilities for B we investigate in both cases $\mathbb{S}$ cannot be *com*-realizable in SYMAFs.

- If $B \neq S_1$ and $B \neq S_2$ then B, S_1 and S_2 are in $\mathit{com}(F)$. That is, $|\mathit{com}(F)| > 2$.

- If B is equivalent with either S_1 or S_2 then F contains isolated arguments B and arguments which are not isolated. Since F contains arguments which are not isolated, by Lemma 9 F contains at least two incomparable preferred extensions. Each preferred extension is a complete extension. Hence, $|com(F)| > 2$.

Then, $com(F) \neq \mathbb{S}$. That is, $\mathbb{S}$ is not *com*-realizable in this case, too. □

It is proven by Theorem 11 that $\mathbb{S}$ is σ-realizable in AFs, for $\sigma \in \{cf, nai, grd\}$, if and only if it is σ-realizable in SYMAFs. Corollary 3 together with the fact that $\Sigma^{\alpha}_{\text{SYMAF}} \subseteq \Sigma^{\alpha}_{AF}$ show that if $|\mathbb{S}| = 1$ then $\mathbb{S}$ is σ-realizable in AFs if and only if σ-realizable in SYMAFs, for $\sigma = \{adm, cf, nai, prf, stb, com, grd\}$. In addition, it is shown by Proposition 8 that whenever $|\mathbb{S}| = 2$ and $\mathbb{S}$ is σ-realizable in AFs, for $\sigma \in \{prf, stb\}$, then it is σ-realizable in SYMAFs . However, we investigate in Proposition 10 that each extension-set $\mathbb{S}$ s.t. $|\mathbb{S}| = 2$ is not *com*-realizable in SYMAFs. In the following we try to answer this natural question that whenever $\mathbb{S}$ is an extension-set which is *adm*-realizable in AFs and $|\mathbb{S}| = 2$ whether $\mathbb{S}$ is *adm*-realizable in SYMAFs. There are a large number of *adm*-realizable extension-sets in AFs which are also *adm*-realizable by SYMAFs, for instance, $\mathbb{S} = \{\emptyset, \{a\}\}$. In general, whenever $\mathbb{S}$ is *adm*-realizable in AFs and $|\mathbb{S}| = 2$ s.t. the cardinality of one of the elements of $\mathbb{S}$ is one then $\mathbb{S}$ is *adm*-realizable in SYMAFs. In Example 38 we show that there exists an extension-set with cardinality two which is *adm*-realizable in AFs but not in SYMAFs.

Example 38. Let $\mathbb{S} = \{\emptyset, \{a, b\}\}$ be an extension-set. A witness of *adm*-realizability of $\mathbb{S}$ in AFs is an AF F, depicted in Figure 4.3. We claim that $\mathbb{S}$ is not *adm*-realizable in SYMAFs. Towards a contradiction assume that there is a SYMAF $F' = (A', L')$ s.t. $adm(F') = \mathbb{S}$. Since F' is a SYMAF, $A' = \{a, b\}$. Otherwise, an additional argument has to appear in at least one of admissible interpretations of F'. Since F' is a SYMAF, it does not contain any self-attack. There are two possibilities of defining relations of F' as follows:

- a and b are isolated arguments, ($L' = \emptyset$): $adm(F') = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$.
- There is a symmetric link between a and b, ($L' = \{(a, b), (b, a)\}$): $adm(F') = \{\emptyset, \{a\}, \{b\}\}$.

In both cases $adm(F') \neq \mathbb{S}$. Therefore, $\mathbb{S}$ is not *adm*-realizable in SYMAFs.

4.3 Expressiveness of ADFs

In Chapter 3 we have studied some subclasses of ADFs: acyclic ADFs (ACADFs for short), attack symmetric ADFs (ASADFs for short) and acyclic support symmetric ADFs (ASSADFs for short). It is reasonable to study the expressiveness of these subclasses compared to other formalisms, namely, AFs, BADFs and ADFs. As mentioned in the beginning of this chapter, from now on we are working with three-valued interpretations.

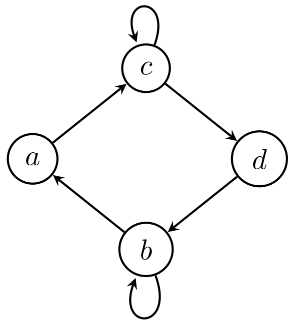

Figure 4.3: AF used in Example 38

Definition 52. Let $F = (A, L, C)$ be an ADF and $\mathcal{V}$ be the set of all three-valued interpretations of A. A set $\mathbb{S} \subseteq 2^{\mathcal{V}}$ is named an *interpretation-set* if $|A|$ is finite.

Whenever we are dealing with finite ADFs, for each semantics σ each $\mathbb{S} \in \Sigma^{\sigma}_{ADF}$ is named an interpretation-set.

4.3.1 Expressiveness of Subclasses of ADFs

In Proposition 11 it is shown that BADFs are more expressive than ASSADFs for $\sigma \in \{adm, prf, com, mod\}$. Expressivity of ASSADFs compared to ASADFs is investigated by Proposition 12.

Proposition 11. *For $\sigma \in \{adm, prf, com, mod\}$ the following relation holds:*

$$\Sigma^{\sigma}_{ASSADF} \subsetneq \Sigma^{\sigma}_{BADF}$$

Proof. Since the family of ASSADFs is, by definition, a strict subset of the family of BADFs, $\Sigma^{\sigma}_{\text{ASSADF}} \subseteq \Sigma^{\sigma}_{BADF}$ for $\sigma \in \{adm, prf, com, mod\}$. To show that Σ^{σ}_{BADF} is a strict superset of $\Sigma^{\sigma}_{\text{ASSADF}}$ it is enough to find an interpretation-set $\mathbb{S}$ which is σ-realizable in BADFs, for $\sigma \in \{adm, prf, com, mod\}$, but not σ-realizable in ASSADFs.

- To investigate that $\Sigma^{\sigma}_{\text{ASSADF}} \subsetneq \Sigma^{\sigma}_{BADF}$ for $\sigma \in \{prf, mod\}$, we show that there exists a witness of σ-realizability in BADFs which is not σ-realizable in ASSADF. Suppose $\mathbb{S} = \{\{a \mapsto t\}, \{a \mapsto f\}\}$. A witness of σ-realizability of $\mathbb{S}$ in BADFs is $F = (\{a\}, \{\varphi_a : a\})$, that is, $mod(F) = prf(F) = \mathbb{S}$. Suppose to the contrary that $\mathbb{S}$ is σ-realizable in ASSADFs. That is, there exists an ASSADF F' s.t. $mod(F') = prf(F') = \mathbb{S}$. Since $\mathbb{S}$ contains only one argument, F' has to have just one argument. Otherwise, an additional argument has to appear in preferred interpretations and two-valued models of F'. Since F' is an ASSADF and it contains only one argument, it does not have any link. Therefore, there are only two ways to define an acceptance condition of a:

  – If $\varphi_a \equiv \top$ then $mod(F') = prf(F') = \{\{a \mapsto t\}\}$,
  – if $\varphi_a \equiv \bot$ then $mod(F') = prf(F') = \{\{a \mapsto f\}\}$.

  In both cases $\sigma(F') \neq \mathbb{S}$ for $\sigma \in \{mod, prf\}$. That is, $\mathbb{S}$ is not σ-realizable in ASSADFs.

- To show that $\Sigma^{\sigma}_{\text{ASSADF}} \subsetneq \Sigma^{\sigma}_{BADF}$, for $\sigma \in \{com, adm\}$, let $\mathbb{X} = \{\{a \mapsto u\}, \{a \mapsto t\}, \{a \mapsto f\}\}$. The witness of σ-realizability of $\mathbb{X}$ in BADFs for $\sigma \in \{com, adm\}$ is the BADF $F = (\{a\}, \{\varphi_a : a\})$. The proof method to show that $\mathbb{X}$ is not σ-realizable in ASSADFs for $\sigma \in \{com, adm\}$ is exactly the same as previous part of the proof.

□

Proposition 12. *Let $\sigma \in \{adm, prf, com\}$. It holds that*

$$\Sigma^{\sigma}_{ASADF} \subsetneq \Sigma^{\sigma}_{ASSADF}$$

Proof. Since the family of ASADFs is a subset of the family of ASSADFs it is clear that $\Sigma^{\sigma}_{\text{ASADF}} \subseteq \Sigma^{\sigma}_{\text{ASSADF}}$, for an arbitrary interpretation σ. In the following we show that $\Sigma^{\sigma}_{\text{ASSADF}}$ is a strict superset of $\Sigma^{\sigma}_{\text{ASADF}}$, for $\sigma \in \{adm, prf, com\}$.

- To investigate that $\Sigma^{prf}_{\text{ASADF}} \subsetneq \Sigma^{prf}_{\text{ASSADF}}$, let $\mathbb{S} = \{\{a \mapsto u, b \mapsto u\}\}$. A witness of *prf*-realizability of $\mathbb{S}$ in ASSADFs is $F = (\{a, b\}, \{\varphi_a : b, \varphi_b : \neg a\})$. Suppose to the contrary that $\mathbb{S}$ is *prf*-realizable in ASADFs. Then, there exists an ASADF F' s.t. $prf(F') = \mathbb{S}$. The set of arguments of F' is $\{a, b\}$. Otherwise, an additional argument has to appear in preferred interpretation of F'. None of the arguments could be an isolated argument. Otherwise, their acceptance conditions are either equivalent with $\top$ or $\bot$. Then there is a symmetric attack link between a and b. It is clear that in this case, $prf(F') = \{\{a \mapsto t, b \mapsto f\}, \{a \mapsto f, b \mapsto t\}\}$. Hence, $\mathbb{S}$ is not *prf*-realizable in ASADFs.

- To show $\Sigma^{com}_{\text{ASADF}} \subsetneq \Sigma^{com}_{\text{ASSADF}}$ we use an interpretation-set $\mathbb{S} = \{\{a \mapsto u, b \mapsto u\}\}$. It is clear that $\mathbb{S}$ is *com*-realizable in ASSADFs by $F = (\{a, b\}, \{\varphi_a : b, \varphi_b : \neg a\})$. Suppose to the contrary that $\mathbb{S}$ is *com*-realizable in ASADFs. That is, there exists an ASADF F' s.t. $com(F') = \mathbb{S}$. The same as before, F' contains only two arguments a and b s.t. none of them is an isolated argument. Then there is a symmetric attack link between a and b. Therefore, $com(F') = \{\{a \mapsto u, b \mapsto u\}, \{a \mapsto t, b \mapsto f\}, \{a \mapsto f, b \mapsto t\}\}$. That is, $com(F') \neq \mathbb{S}$. Hence, $\mathbb{S}$ is not *com*-realizable in ASADFs.

- To show that $\Sigma^{adm}_{\text{ASADF}} \subsetneq \Sigma^{adm}_{\text{ASSADF}}$ we claim that the given $\mathbb{S}$ and F of previous parts also work here. It is easy to see that $F = (\{a, b\}, \{\varphi_a : b, \varphi_b : \neg a\})$ is a witness of *adm*-realizability of $\mathbb{S} = \{\{a \mapsto u, b \mapsto u\}\}$ in ASSADFs. However, $\mathbb{S}$ is not *adm*-realizable in ASADFs. It is easy to proof this claim in exactly the same way as we did for $\sigma = com$.

□

In the following we try to investigate that neither ASADFs are more expressive than AFs nor AFs are more expressive than ASADFs, for $\sigma \in \{adm, prf, com\}$. The former one is proven in Proposition 13 and the latter one is shown in Proposition 14.

Proposition 13. *$\Sigma^{\sigma}_{AF} \not\subseteq \Sigma^{\sigma}_{ASADF}$, for $\sigma \in \{adm, prf, com\}$.*

Proof. To investigate that $\Sigma^{\sigma}_{AF} \not\subseteq \Sigma^{\sigma}_{\text{ASADF}}$, for $\sigma \in \{adm, prf, com\}$ we show that there exists an interpretation-set which is σ-realizable in AFs but not in ASADFs, for $\sigma \in \{adm, prf, com\}$.

- To show $\Sigma^{\sigma}_{AF} \not\subseteq \Sigma^{\sigma}_{\text{ASADF}}$, for $\sigma \in \{prf, com\}$, suppose $\mathbb{S} = \{\{d \mapsto u\}\}$. The witness of σ-realizability of $\mathbb{S}$ in AFs for $\sigma \in \{prf, com\}$ is $F = (\{d\}, \{(d, d)\})$. The ADF associated to F is $D_F = (\{d\}, \{(\varphi_d : \neg d\}))$. However $\mathbb{S}$ is not σ-realizable in ASADFs for $\sigma \in \{prf, com\}$. Suppose to the contrary that there exists an ASADF F' s.t. $\sigma(F') = \mathbb{S}$ for $\sigma \in \{prf, com\}$. The set of arguments of F' has to be $\{d\}$. Otherwise, an additional argument has to appear in preferred (resp. complete) interpretations of F'. Since F' is assumed as an ASADF, its relations are irreflexive. Therefore, there are two possibilities to define acceptance condition of d as follows:
  - If $\varphi_d \equiv \top$ then $prf(F') = com(F') = \{\{d \mapsto t\}\}$.
  - If $\varphi_d \equiv \bot$ then $prf(F') = com(F') = \{\{d \mapsto f\}\}$.

  In both cases $\sigma(F') \neq \mathbb{S}$ for $\sigma \in \{prf, com\}$. Hence, $\mathbb{S}$ is not σ-realizable in ASADFs for $\sigma \in \{prf, com\}$.
- To show that $\Sigma^{adm}_{AF} \not\subseteq \Sigma^{adm}_{\text{ASADF}}$, again let $\mathbb{S} = \{\{d \mapsto u\}\}$. Given an AF $F = (\{d\}, \{(d, d)\})$ and its correspondence ADF $D_F = (\{d\}, \{\varphi_d : \neg d\}))$ are witness of adm-realizability of $\mathbb{S}$ in AFs. Towards a contradiction assume that $\mathbb{S}$ is adm-realizable in ASADFs by F'. By the same arguments that we have in the previous part there are two posibilities of defining acceptance condition of D as follows:
  - If $\varphi_d \equiv \top$ then $adm(F') = \{\{d \mapsto u\}, \{d \mapsto t\}\}$.
  - If $\varphi_d \equiv \bot$ then $adm(F') = \{\{d \mapsto u\}, \{d \mapsto f\}\}$.

  In both cases $adm(F') \neq \mathbb{S}$. Therefore, $\mathbb{S}$ is not adm-realizable in ASADFs.

□

Proposition 14. *$\Sigma^{\sigma}_{ASADF} \not\subseteq \Sigma^{\sigma}_{AF}$, for $\sigma \in \{adm, prf, com\}$.*

Proof.

- To show that $\Sigma^{prf}_{\text{ASADF}} \not\subseteq \Sigma^{prf}_{AF}$, let

$$\begin{aligned}\mathbb{S} = \{&\{a \mapsto f, b \mapsto t, c \mapsto t, e \mapsto t\},\\ &\{a \mapsto t, b \mapsto f, c \mapsto t, e \mapsto f\},\\ &\{a \mapsto t, b \mapsto t, c \mapsto f, e \mapsto f\}\}.\end{aligned}$$

  A witness of *prf*-realizability of $\mathbb{S}$ in ASADFs is $F = (\{a, b, c, e\}, \{\varphi_a : \neg e \wedge (\neg b \vee \neg c), \varphi_b : \neg a \vee \neg c, \varphi_c : \neg a \vee \neg b, \varphi_e : \neg a\})$. Note that all arguments in F only occur as attackers and all links are symmetric, thus F is clearly an ASADF. Suppose to the contrary that $\mathbb{S}$ is *prf*-realizable in AFs. Then, there exists an F' in AFs s.t. $prf(F') = \mathbb{S}$. The set of arguments of F' is $\{a, b, c, e\}$. Otherwise, an additional argument has to appear in $\mathbb{S}$. Because of the first set in $\mathbb{S}$ there is no attack between b and c in F', because of the second set in $\mathbb{S}$ there is no attack between a and c in F' and because of the third set there is no attack between a and b in F'. That is, there is no link between a, b and c in F'.

  - If there is no attack from any of a, b and c to e then e has to be assigned to t in all preferred interpretations of F'.
  - If there is an attack from any of a, b and c to e then $a \mapsto t, b \mapsto t$ and $c \mapsto t$ appear in a preferred interpretation of F'.

  In both cases, $prf(F') \neq \mathbb{S}$. Hence, $\mathbb{S}$ is not *prf*-realizable in AFs.

- To show that $\Sigma^{com}_{\text{ASADF}} \not\subseteq \Sigma^{com}_{AF}$, let $\mathbb{S}' = \mathbb{S} \cup \{\{a \mapsto u, b \mapsto u, c \mapsto u, e \mapsto u\}\}$, for $\mathbb{S}$ defined in the previous part. It is easy to check that $\mathbb{S}'$ is *com*-realizable in ASADFs. A witness of this realizability is the ASADF $F = (\{a, b, c, e\}, \{\varphi_a : \neg e \wedge (\neg b \vee \neg c), \varphi_b : \neg a \vee \neg c, \varphi_c : \neg a \vee \neg b, \varphi_e : \neg a\})$ defined above. Towards a contradiction assume that $\mathbb{S}'$ is *com*-realizable in AFs. That is, there exists an AF F' s.t. $com(F') = \mathbb{S}'$. Since all elements of $\mathbb{S}$ are incomparable and are two-valued, each of them is a preferred interpretation of F'. Therefore, $prf(F') = \mathbb{S}$. That is, $\mathbb{S}$ defined in the previous part of the proof is *prf*-realizable in AFs, which is a contradiction.

- To show that $\Sigma^{adm}_{\text{ASADF}} \not\subseteq \Sigma^{adm}_{AF}$, again let $F = (\{a, b, c, e\}, \{\varphi_a : \neg e \wedge (\neg b \vee \neg c), \varphi_b : \neg a \vee \neg c, \varphi_c : \neg a \vee \neg b, \varphi_e : \neg a\})$ and $\mathbb{X} = adm(F)$. Then, obviously $\mathbb{X}$ is *adm*-realizable in ASADFs and $prf(F) = \mathbb{S}$, for $\mathbb{S}$ defined in first part. Towards a contradiction assume that $\mathbb{X}$ is *adm*-realizable in AFs. That is, there exists an AF F' s.t. $adm(F') = \mathbb{X}$. Therefore, the set of maximal admissible interpretations of F' is equivalent with the set of maximal admissible interpretations of F. That is, $prf(F') = \mathbb{S}$. Hence, the interpretation-set $\mathbb{S}$ defined in the first part is *prf*-realizable in AFs, which is a contradiction.

□

Theorem 12 is a direct consequence of Proposition 13 and Proposition 14 s.t. the former one shows that $\Sigma^{\sigma}_{AF} \not\subseteq \Sigma^{\sigma}_{\text{ASADF}}$ and the latter one shows $\Sigma^{\sigma}_{\text{ASADF}} \not\subseteq \Sigma^{\sigma}_{AF}$ for $\sigma \in \{adm, prf, com\}$.

Theorem 12. *$\Sigma^{\sigma}_{AF} \not\sim \Sigma^{\sigma}_{ASADF}$, for $\sigma \in \{adm, prf, com\}$.*

It is shown by Theorem 12 that Σ^{σ}_{AF} and $\Sigma^{\sigma}_{\text{ASADF}}$ are incomparable under $\sigma \in \{adm, prf, com\}$. In the following we investigate whether they are incomparable under $\sigma = stb$ as well.

Proposition 15. *$\Sigma^{stb}_{ASADF} \not\subseteq \Sigma^{stb}_{AF}$.*

Proof. Let $\mathbb{S} = \{\{a \mapsto t, b \mapsto t, c \mapsto f\}, \{a \mapsto f, b \mapsto t, c \mapsto t\}, \{a \mapsto t, b \mapsto f, c \mapsto t\}\}$ be an interpretation-set. A witness of *stb*-realizability of $\mathbb{S}$ in ASADFs is $F = (\{a, b, c\}, \{\varphi_a : \neg b \vee \neg c, \varphi_b : \neg a \vee \neg c, \varphi_c : \neg a \vee \neg b\})$. That is, $\mathbb{S} \in \Sigma^{stb}_{\text{ASADF}}$. We investigate that $\mathbb{S}$ is not *stb*-realizable in AFs. Towards a contradiction assume that there exists an AF $F' = (A', L')$ s.t. $stb(F') = \mathbb{S}$. The set of arguments of F' is equivalent with $\{a, b, c\}$ otherwise it has to appear in $stb(F')$. Because of the first set in $\mathbb{S}$ there is no link between a an b in F', because of the second set in $\mathbb{S}$ there is no link between b and c and because of the third set in $\mathbb{S}$ there is no link between a an c in F'. Then there is no link among a, b and c in F'. Then, $\{a \mapsto t, b, \mapsto t, c \mapsto t\}$ is an element of $stb(F')$. That is, $stb(F') \neq \mathbb{S}$. Hence, $\mathbb{S} \notin \Sigma^{stb}_{AF}$. □

Corollary 4. *$\Sigma^{stb}_{\alpha} \not\subseteq \Sigma^{stb}_{AF}$ for $\alpha \in \{ASADF, ASSADF, BADF, ADF\}$.*

Proof. Each ASADF is an ASSADF, is a BADF and is an ADF. Therefore, interpretation-set $\mathbb{S} = \{\{a \mapsto t, b \mapsto t, c \mapsto f\}, \{a \mapsto f, b \mapsto t, c \mapsto t\}, \{a \mapsto t, b \mapsto f, c \mapsto t\}\}$ defined in Proposition 15 is *stb*-realizable in α, for $\alpha \in \{\text{ASADF}, \text{ASSADF}, BADF, ADF\}$. By Proposition 15 $\mathbb{S}$ is not *stb*-realizable in AFs. That is, for each $\alpha \in \{\text{ASADF}, \text{ASSADF}, BADF, ADF\}$, $\Sigma^{stb}_{\alpha} \not\subseteq \Sigma^{stb}_{AF}$. □

Corollary 5. *$\Sigma^{stb}_{ASADF} \not\subseteq \Sigma^{nai}_{AF}$.*

Proof. By Proposition 15 $\Sigma^{stb}_{\text{ASADF}} \not\subseteq \Sigma^{stb}_{AF}$ and we know that the family of SYMAFs is a subset of the family of AFs, therefore, $\Sigma^{stb}_{\text{ASADF}} \not\subseteq \Sigma^{stb}_{\text{SYMAF}}$. By Theorem 11 $\Sigma^{nai}_{AF} = \Sigma^{nai}_{\text{SYMAF}}$ and by Theorem 10 $\Sigma^{stb}_{\text{SYMAF}} = \Sigma^{nai}_{\text{SYMAF}}$. Therefore, $\Sigma^{stb}_{\text{SYMAF}} = \Sigma^{nai}_{AF}$. Hence, $\Sigma^{stb}_{\text{ASADF}} \not\subseteq \Sigma^{nai}_{AF}$. □

In the following we would like to investigate whether ASADFs are strictly more expressive than AFs under stable interpretation, that is, $\Sigma^{stb}_{AF} \subsetneq \Sigma^{stb}_{\text{ASADF}}$. It is mentioned in [17] that each AF F is associated to an ADF D_F. Therefore, each symmetric AF associates to a symmetric ADF. That is, whenever an interpretation-set $\mathbb{S}$ is *stb*-realizable in SYMAFs it is also *stb*-realizable in ASADFs, $\Sigma^{stb}_{\text{SYMAF}} \subseteq \Sigma^{stb}_{\text{ASADF}}$. In Theorem 11 it is proven that $\Sigma^{stb}_{\text{SYMAF}} \subsetneq \Sigma^{stb}_{AF}$, that is, there exists an extension-set which is *stb*-realizable in AFs and it is not *stb*-realizable in SYMAFs. One may guess that this set could be a good candidate to show Σ_{AF}^{stb} and $\Sigma_{\text{ASADF}}^{stb}$ are incomparable. Surprisingly, as is illustrated with the following interpretation-set

$$\begin{aligned}\mathbb{S} = \{&\{a \mapsto t, b \mapsto f, c \mapsto f, d \mapsto f, e \mapsto t, f \mapsto t\},\\ &\{a \mapsto f, b \mapsto t, c \mapsto f, d \mapsto t, e \mapsto f, f \mapsto t\},\\ &\{a \mapsto f, b \mapsto f, c \mapsto t, d \mapsto t, e \mapsto t, f \mapsto f\}\},\end{aligned}$$

$\mathbb{S}$ is *stb*-realizable in ASADFs, however, it is not *stb*-realizable in SYMAFs. A witness of *stb*-realizablity of $\mathbb{S}$ in ASADFs is as follows:

$$\begin{aligned}F = (&\{a, b, c, d, e, f\},\\ &\{\varphi_a : \neg d \wedge \neg b \wedge \neg c,\\ &\varphi_b : \neg a \wedge \neg c \wedge \neg e,\\ &\varphi_c : \neg a \wedge \neg b \wedge \neg f,\\ &\varphi_d : \neg a \wedge (\neg e \vee \neg f),\\ &\varphi_e : \neg b \wedge (\neg d \vee \neg f),\\ &\varphi_f : \neg c \wedge (\neg d \vee \neg e)\})\end{aligned}$$

Proposition 16 bellow shows that the signature of ASADFs under $\sigma = stb$ is a strict super set of the signature of AFs under $\sigma = stb$. That is, if a framework is *stb*-realizable in AFs it is also *stb*-realizable in ASADFs.
It is proven in [25] that for each AF $F = (A, R)$, $stb(F)$ is incomparable. To investigate whether this property holds in ADFs first we define the notion of $\leq_t$ order. This ordering assigns a greater value to t rather than f, that is, $f \leq_t t$. Thus an interpretation v_i is named $\leq_t$-less than v_j if and only if for each argument s, $v_i(s) \leq_t v_j(s)$.

Definition 53. Let $\mathcal{V}$ be an interpretation-set and v_i and v_j be two elements of $\mathcal{V}$.

- v_i and v_j are called $\leq_t$-incomparable whenever neither $v_i \leq_t v_j$ nor $v_j \leq_t v_i$. This is denoted by $v_i \not\sim_t v_j$.
- $\mathcal{V}$ is named $\leq_t$-incomparable if all elements of $\mathcal{V}$ are pairwise incomparable.

It is remarked in [41] and rescripted in Lemma 10 that $stb(F)$ of each ADF F is $\leq_t$-incomparable.

Lemma 10. *Let $F = (A, L, C)$ be an ADF. $stb(F)$ is $\leq_t$-incomparable.*

Proposition 16. $\Sigma_{AF}^{stb} \setminus \{v_\epsilon\} \subsetneq \Sigma_{ASADF}^{stb}$

Proof. By Proposition 15 we know that $\Sigma_{\text{ASADF}}^{stb} \not\subseteq \Sigma_{AF}^{stb}$. To show that $\Sigma_{\text{ASADF}}^{stb}$ is a strict superset of Σ_{AF}^{stb} we show that $\Sigma_{AF}^{stb} \subseteq \Sigma_{\text{ASADF}}^{stb}$. Let $\mathbb{S} = \{v_1, \ldots, v_n\}$ be an interpretation-set which is *stb*-realizable in AFs. Therefore, there exists an AF $F = (A, L)$ s.t. $stb(D_F) = \mathbb{S}$ in which, D_F is the ADF associated to F. We claim that an ADF $F' = (A', L', C')$ constructed as follows is a desired ASADF:

- $A' = A$
- If $a \mapsto t \in \bigsqcap v_i$ then $\varphi_a = \top$.
- If $a \mapsto f \in \bigsqcap v_i$ then $\varphi_a = \bot$.
- Otherwise,

$$\varphi_a : \bigvee_{v_i \in \mathbb{S}, v_i(a)=t} \quad \bigwedge_{\substack{v_i(b)=f \wedge \\ \exists v_j \in \mathbb{S}:(v_j(a)=f \wedge v_j(b)=t)}} \neg b \tag{4.1}$$

In the following first we illustrate that F' is an ASADF then we show that $stb(F') = \mathbb{S}$.

- To show that F' is an ASADF we show that F' is a symmetric ADF in which all links are attacking.
  - Assume a is an argument which is not assigned to t (resp. f) in all v_i, for $1 \leq i \leq n$, then, φ_a is defined as above. Assume that b is an argument which appears in φ_a. To show that F' is symmetric we investigate that a also appears in φ_b. b appears in φ_a if and only if, by the definition there exists $v_i \in \mathbb{S}$ s.t. $v_i(a) = t$ and $v_i(b) = f$, and there exists $v_j \in \mathbb{S}$ s.t. $v_j(b) = t$ and $v_j(a) = f$. That is, there exists $v_j \in \mathbb{S}$ s.t. $v_j(b) = t$ and $v_j(a) = f$, and there exists $v_i \in \mathbb{S}$ s.t. $v_i(a) = t$ and $v_i(b) = f$, therefore, by definition a appears in φ_b. That is, all links are symmetric. Since, each argument a cannot assign to t and f in a v_i, the defined ADF F' is irreflexive.
  - Assume that $v_i(a) = t$ (resp. $v_i(a) = f$) appears in all v_i, for $1 \leq i \leq n$ then $\varphi_a \equiv \top$ (resp. $\varphi_a \equiv \bot$). By the previous part all links of F' are symmetric. That is, such an argument a is an isolated argument.

  That is, F' is a symmetric ADF.
  It remains to show that F' is an attack symmetric ADF which is clear by the syntax of the acceptance condition of each argument.

To investigate that $stb(F') = \mathbb{S}$ we show that $\mathbb{S} \subseteq stb(F')$ and $stb(F') \subseteq \mathbb{S}$.

- To prove that $\mathbb{S} \subseteq stb(F')$ let v_i be an arbitrary element of $\mathbb{S}$. We show that $v_i \in stb(F')$. In order to show that $v_i \in stb(F')$ first we investigate whether v_i is a two-valued model of F'.

  Let a be an argument s.t. $v_i(a) = t$. There are two cases either $a \mapsto t$ in all elements of $\mathbb{S}$ or not. If $a \mapsto t$ in all elements of $\mathbb{S}$ then by the definition $\varphi_a = \top$. Hence, $\varphi_a^{v_i} \equiv \top$.
  If it is not the case that $a \mapsto t$ in all elements of $\mathbb{S}$ then there exists $v_j \in \mathbb{S}$ s.t. $v_j(a) = f$. It is proven in [25] that for each AF F, $stb(F)$ is incomparable.

Hence, $\mathbb{S}$ is incomparable. Therefore, there exists b_i s.t. $v_i(b_i) = f$ and $v_j(b_i) = t$. Hence, b_i appears in the acceptance condition of a. The set of all arguments like b_i which are assigned to f by v_i and are assigned to t by a v_j s.t. $a \mapsto f \in v_j$ make a conjunctive clause of φ_a which guarantees that $\varphi_a^{v_i} \equiv \top$.

Let a be an argument s.t. $v_i(a) = f$. If a is assigned to f by all elements of $\mathbb{S}$ then by the definition $\varphi_a = \bot$ thus $\varphi_a^{v_i} \equiv \bot$.
If it is not the case that a is assigned to f by all elements of $\mathbb{S}$ then there exists $v_j \in \mathbb{S}$ s.t. $v_j(a) = t$. Let N be the set of all elements of $\mathbb{S}$ in which a is assigned to t. Since $\mathbb{S}$ is incomparable, for each $v_j \in N$ there exists an argument b_j s.t. $v_j(b_j) = f$ and $v_i(b_j) = t$. Therefore, each conjunctive clause of the acceptance condition of a contains such a b_j. This guarantees that $\varphi_a^{v_i} \equiv \bot$.

Hence, each element of $\mathbb{S}$ is a two-valued model of F'. By Proposition 4 each ASADF is weak-coherent, that is, each two-valued model of an ASADF is stable. Since F' is an ASADF, each element of $\mathbb{S}$ is a stable model of F'.

- To show that $stb(F') \subseteq \mathbb{S}$ if $|\mathbb{S}| = 1$ then by the construction of F' it is clear that $stb(F') = \mathbb{S}$. Suppose that $|\mathbb{S}| > 1$ we assume that there exists $v \in stb(F')$ s.t. $v \notin \mathbb{S}$. Then we show that v is not a model of F' and in consequence it cannot be a stable model of F'. Lemma 10 together with the assumption that $|\mathbb{S}| > 1$ and the fact that F' is an ASADF implies that there exists at least an $a \in A$ s.t. $\varphi_a \not\equiv \top$ and $\varphi_a^v \equiv \top$, otherwise, v is $\leq_t$-comparable with other elements of $stb(F)$. Fix this a and let K be the set of all $v_i \in \mathbb{S}$ in which $v_i(a) = t$. Since $v \notin \mathbb{S}$ and $stb(F')$ is $\leq_t$-incomparable, in each $v_i \in K$ there exists at least an argument b_i s.t. $v(b_i) = t$ and $v_i(b_i) = f$. Let B be the set of all such b_i's. We claim that either each conjunctive clause of φ_a contains a b_i, $1 \leq i \leq m$ or there exists at least a $b_i \in B$ s.t. each conjunctive clause of φ_{b_i} contains an element of B. By proof of this claim we show that either $\varphi_a^v \equiv \bot$ or there exists a $b_i \in B$ s.t. $\varphi_{b_i}^v \equiv \bot$. Thus, v cannot be a model of F' and consequently, it cannot be a stable model of F'.

  We proof the claim by induction on the cardinality of K.

  Base case: $|K| = 1$, that is, there exists only one v_i in which a is assigned to t and b_i is assigned to f. Since b_i is assigned to t by v and is assigned to f by v_i, $\varphi_{b_i} \not\equiv \bot$. Then, there exists v_j in which $v_j(a) = f$ and $v_j(b_i) = t$, otherwise, $\varphi_{b_i} \equiv \bot$. Therefore, b_i appears in a conjunctive clause of φ_a. Since a is assigned to t by only one of v_i, its acceptance condition contains only one conjunctive clause. Therefore, if $\varphi_{b_i}^v = t$ then $\varphi_a^v = f$. That is, v is not a model of F' and in consequence it is not a stable model of F'.
  To clarify the idea of the proof we show also base case for $|K| = 2$. That is, when a is assigned to t by exactly two elements of $\mathbb{S}$. Assume that a is assigned to t by v_i and v_j. By the assumption there exist b_i and b_j s.t. $v(b_i) = v(b_j) = t$ and $v_i(b_i) = f$ and $v_j(b_j) = f$. By the same reason that we

describe for $|K| = 1$, there exist elements of $\mathbb{S}$ in which b_i and b_j are assigned to t. If they are assigned to true in elements by which a is assigned to f there is noting to do. Since it guarantees that b_i and b_j occur in conjunctive clauses of φ_a. Assume that none of b_i and b_j is assigned to t by any of the $\mathbb{S} \setminus K$. Therefore, b_i has to be assigned to t by v_j and b_j has to be assigned to f by v_i. This means that b_i appears in a conjunctive clause of φ_{b_j} and b_j appears in a conjunctive clause of φ_{b_i}. That is, b_i and b_j cannot be assigned to t by a model of F', simultaneously. Thus, v cannot be a stable model.

Induction step: Let $|K| = m$, that is, a is assigned to t by exactly m elements of $\mathbb{S}$. W.l.o.g assume $v_1, \ldots, v_m$ are elements of $\mathbb{S}$ by which a is assigned to t and $b_1, \ldots, b_m$ are arguments which are assigned to t by v and are assigned to f by $v_1, \ldots, v_m$, respectively. If each conjunctive clause of φ_a contains a b_i there is nothing to prove. Assume that there exists at least a conjunctive clause of φ_a which does not contain any b_i we show that there exists a b_i s.t. each of the conjunctive clauses of its acceptance condition contains a b_j. W.l.o.g. assume that b_1 is an argument which does not appear in any conjunctive clause of φ_a. If each conjunctive clause in φ_{b_1} contains a b_j there is nothing to do, otherwise, eliminate v_1 from K. That is, $|K|$ become $m - 1$ which means the acceptance condition of a contains exactly $m - 1$ conjunctive clauses. By induction hypothesis, either all of these conjunctive clauses contain b_i's or there exists b_i s.t. each of the conjunctive clauses of φ_{b_i} contains a b_j. In the former case all b_i's which appear in φ_a have to be assigned to t by v_1, otherwise, all of m conjunctive clauses of φ_a contain b_i's, which is a contradiction with our assumption. Since $\varphi_{b_1} \not\equiv \bot$ and b_1 does not appear in φ_a, b_1 has to be assigned to t by a $v_i \in K$. Since by our assumption all of these $m - 1$ conjunctive clauses of the acceptance condition of a contains a b_i, there exists a b_j which is assigned to t by v_1 and assigned to f by v_i. Therefore, all conjunctive clauses of b_1 contain a b_i, which is a contradiction! That is, it is not the case that all of these $m - 1$ conjunctive clauses in the acceptance condition of a contains a b_i. Therefore, by the latter part of the induction hypothesis there exists a b_i s.t. each of the conjunctive clause of its acceptance condition contains a b_j. Fix this b_i.

If $v_1(b_i) = f$ the acceptance condition of b_i does not change by adding v_1, which is removed from K. That is, b_i is an argument s.t. all the conjunctive clauses of φ_{b_i} contains one of the elements of B. Thus, b_i is a desired argument. Assume that $v_1(b_i) = t$ if at least one of the b_j's which appears in φ_{b_i} is assigned to false by v_1 again b_i is a desired argument. Otherwise, since b_1 does not appear in the acceptance condition of a, it has to be assigned to t by at least one of $v_i \in K$. Let v_l be an element of K s.t. $v_l(b_1) = t$, either b_i is assigned to t or to f by v_l. If $v_l(b_i) = f$ then b_1 appears in a conjunctive clause of b_i, which means again b_i is a desired argument. If $v_l(b_i) = t$ by our assumption there exists an argument b_k which appears in acceptance condition of b_i and $v_l(b_k) = f$. Therefore, b_k appears in φ_{b_1}. Hence, each conjunctive clause of b_1

contains an element of B. That is, b_1 is a desired argument. That is, if φ_a contains a conjunctive clause which does not contain any b_i then there exists at least a b_i s.t. each conjunctive clause of its acceptance condition contains an element of B. That is, all elements of B's and a cannot be assigned to t by a model of F'. Thus, v cannot be a model of F' and in consequence, v is not a stable model of F'.

□

Now we strength Proposition 13 in the following proposition. The construction of the proof is exactly the same.

Proposition 17. $\Sigma^{\sigma}_{AF} \nsubseteq \Sigma^{\sigma}_{ASSADF}$, *for* $\sigma \in \{adm, prf, com\}$.

Proof. To prove that $\Sigma^{\sigma}_{AF} \nsubseteq \Sigma^{\sigma}_{\text{ASSADF}}$, for $\sigma \in \{adm, prf, com\}$ again let $\mathbb{S} = \{\{d \mapsto u\}\}$ defined in Proposition 13. As it is mentioned, a witness of σ-realizability in AF, for $\sigma \in \{adm, prf, com\}$ is $F = (\{d\}, \{(d, d)\})$ and its ADF correspondence to F is $D_F = (\{d\}, \{\varphi_d : \neg d\})$. Suppose to the contrary that there exists an ASSADF F' s.t. $\sigma(F') = \mathbb{S}$, for $\sigma \in \{adm, prf, com\}$. The set of arguments of F' is $\{d\}$. Otherwise, an additional argument has to appear in interpretations of F'. Since F' is assumed as an ASSADF, by the definition, all relations are irreflexive. Therefore, There are two possibilities of defining acceptance condition of F' as follows:

- If $\varphi_d \equiv \top$ then $prf(F') = com(F') = \{\{d \mapsto t\}\}$ and $adm(F') = \{\{d \mapsto u\}, \{d \mapsto t\}\}$.
- If $\varphi_d \equiv \bot$ then $prf(F') = com(F') = \{\{d \mapsto f\}\}$ and $adm(F') = \{\{d \mapsto u\}, \{d \mapsto f\}\}$.

In both cases, $\sigma(F') \neq \mathbb{S}$, for $\sigma \in \{adm, prf, com\}$. Hence, $\mathbb{S} \notin \Sigma^{\sigma}_{\text{ASSADF}}$, for $\sigma \in \{adm, prf, com\}$. □

Proposition 18. $\Sigma^{\sigma}_{ASSADF} \nsubseteq \Sigma^{\sigma}_{AF}$, *for* $\sigma \in \{adm, prf, com\}$.

Proof. By Proposition 14 $\Sigma^{\sigma}_{\text{ASADF}} \nsubseteq \Sigma^{\sigma}_{AF}$, for $\sigma \in \{adm, prf, com\}$. Then there exists an interpretation-set $\mathbb{S}$ s.t. $\mathbb{S} \in \Sigma^{\sigma}_{\text{ASADF}}$ and $\mathbb{S} \notin \Sigma^{\sigma}_{AF}$, for $\sigma \in \{adm, prf, com\}$. By Proposition 12 $\Sigma^{\sigma}_{\text{ASADF}} \subsetneq \Sigma^{\sigma}_{\text{ASSADF}}$, for $\sigma \in \{adm, prf, com\}$. Therefore, for $\sigma \in \{prf, adm, com\}$ there exists an interpretation-set $\mathbb{S}$ s.t. $\mathbb{S}$ is σ-realizable in ASADFs and ASSADFs but it is not σ-realizable in AFs. Hence, $\Sigma^{\sigma}_{\text{ASSADF}} \nsubseteq \Sigma^{\sigma}_{AF}$, for $\sigma \in \{adm, prf, com\}$. □

Theorem 13 is an immediate consequence of Propositions 17 and 18.

Theorem 13. $\Sigma^{\sigma}_{ASSADF} \not\sim \Sigma^{\sigma}_{AF}$ *for* $\sigma \in \{adm, prf, com\}$.

Proposition 19 indicates that there exists an interpretation-set which is realizable in AFs and ASSADFs but it is not realizable in ASADFs, for $\sigma \in \{adm, prf, com\}$.

Proposition 19. $(\Sigma^{\sigma}_{ASSADF} \cap \Sigma^{\sigma}_{AF}) \setminus \Sigma^{\sigma}_{ASADF} \neq \emptyset$*, for* $\sigma \in \{adm, prf, com\}$*.*

Proof. Let $\mathbb{S} = \{\{a \mapsto u, b \mapsto u\}\}$. A witness of σ-realizability of $\mathbb{S}$ in AFs (resp. in ASSADFs) is $F = (\{a, b\}, \{(a, a), (b, b)\})$ (resp. $F' = (\{a, b\}, \{\varphi_a : b, \varphi_b : \neg a\})$), for $\sigma \in \{adm, prf, com\}$. We show that $\mathbb{S}$ is not σ-realizable in ASADFs. Suppose to a contrary that there exists a ASADF G s.t. $\sigma(G) = \mathbb{S}$ for $\sigma \in \{adm, prf, com\}$. The set of arguments of G is $\{a, b\}$. Otherwise an additional argument has to appear in interpretations of G. Since by our assumption G is an ASADF, either a and b are isolated arguments or there is an attack symmetric link between a and b. In both cases $\sigma(G)$ contains at least an interpretation in which one of these arguments is not assigned to u. Thus, $\mathbb{S}$ is not σ-realizable in ASADFs for $\sigma \in \{adm, prf, com\}$ which is a desired result. $\square$

The following theorem concludes some main results of this section and a theorem in [17].

Theorem 14. *For* $\sigma \in \{adm, com, prf\}$*, we find that*

$$\Sigma^{\sigma}_{ASADF} \subsetneq \Sigma^{\sigma}_{ASSADF} \subsetneq \Sigma^{\sigma}_{BADF} \subsetneq \Sigma^{\sigma}_{ADF}$$

Proof. By Proposition 12 it is shown that $\Sigma^{\sigma}_{\text{ASADF}} \subsetneq \Sigma^{\sigma}_{\text{ASSADF}}$, for $\sigma \in \{adm, com, prf\}$. By Proposition 11 it follows that $\Sigma^{\sigma}_{\text{ASSADF}} \subsetneq \Sigma^{\sigma}_{BADF}$, for $\sigma \in \{adm, prf, com, mod\}$. By Theorem 9 mentioned in [17], for $\sigma \in \{adm, prf, com, mod\}$ we find that $\Sigma^{\sigma}_{BADF} \subsetneq \Sigma^{\sigma}_{ADF}$. Hence, $\Sigma^{\sigma}_{\text{ASADF}} \subsetneq \Sigma^{\sigma}_{\text{ASSADF}} \subsetneq \Sigma^{\sigma}_{BADF} \subsetneq \Sigma^{\sigma}_{ADF}$ for $\sigma \in \{adm, com, prf\}$. $\square$

In the following of this subsection we study expressiveness of acyclic ADFs (ACADFs for short) in comparison to other formalisms. Since each acyclic ADF is an ADF, the immediate result is that $\Sigma^{\sigma}_{\text{ACADF}} \subseteq \Sigma^{\sigma}_{ADF}$ for $\sigma \in \{adm, prf, stb, mod, grd, com\}$. In Proposition 21 it is investigated that Σ^{σ}_{ADF} is a strict supper set of $\Sigma^{\sigma}_{\text{ACADF}}$, for $\sigma \in \{adm, prf, stb, mod, grd, com\}$.

Proposition 20. *Let* $\mathbb{S}$ *be an interpretation-set s.t.* $|\mathbb{S}| > 1$*.* $\mathbb{S}$ *is not* σ*-realizable in ACADFs, for* $\sigma \in \{prf, stb, mod, com\}$*.*

Proof. By Theorem 3 an ACADF D with maximal level m has exactly one complete interpretation which is preferred, stable and a two-valued model. That is, for each ACADF D, $|prf(D)| = |com(D)| = |mod(D)| = |stb(D)| = 1$. Therefore, if $|\mathbb{S}| > 1$ then $\mathbb{S}$ is not σ-realizable in ACADFs, for $\sigma \in \{prf, stb, mod, com\}$. $\square$

By definition the grounded interpretation is always unique. One may conclude that whatever is *grd*-realizable in ADFs is also *grd*-realizable in ACADFs. However, it is shown in Proposition 21 that Σ^{grd}_{ADF} is a strict superset of $\Sigma^{grd}_{\text{ACADF}}$.

Proposition 21. $\Sigma^{\sigma}_{ADF} \nsubseteq \Sigma^{\sigma}_{ACADF}$, *for* $\sigma \in \{adm, prf, stb, mod, com, grd\}$.

Proof. We show that there exists an interpretation-set $\mathbb{S}$ which is σ-realizable in ADFs but not in ACADFs, for $\sigma \in \{adm, prf, stb, mod, com\}$.

- Let $\mathbb{S} = \{\{a \mapsto t, b \mapsto f\}, \{a \mapsto f, b \mapsto t\}\}$. A witness of σ-realizability of $\mathbb{S}$ in ADFs, for $\sigma \in \{prf, stb, mod, com\}$, is $F = (\{a, b\}, \{\varphi_a : \neg b, \varphi_b : \neg a\})$. By Proposition 20 $\mathbb{S}$ is not σ-realizable in ACADFs, for $\sigma \in \{prf, stb, mod, com\}$, since $|\mathbb{S}| > 1$.
- Let $\mathbb{S}' = \{a \mapsto u, b \mapsto u\} \cup \mathbb{S}$. Again $F = (\{a, b\}, \{\varphi_a : \neg b, \varphi_b : \neg a\})$ is a witness of *adm*-realizability of $\mathbb{S}'$ in ADFs. Suppose to a contrary that $\mathbb{S}'$ is *adm*-realizable in ACADFs by F'. By the definition of preferred interpretation $\{a \mapsto t, b \mapsto f\}$ and $\{a \mapsto f, b \mapsto t\}$ are preferred interpretations of F'. That is, $\mathbb{S}$ is *prf*-realizable in ACADFs. This is a contradiction.
- Again by Theorem 3, the unique grounded interpretation of an ACADF D is a two-valued model. Therefore, whenever an interpretation-set $\mathbb{S}$ contains an argument assigned to u it is not *grd*-realizable in ACADFs. For instance, $\mathbb{S} = \{a \mapsto u, b \mapsto u\}$ is *grd*-realizable in ADFs by $F = (\{a, b\}, \{\varphi_a : \neg b, \varphi_b : \neg a\})$ but it is not *grd*-realizable in ACADFs.

□

Corollary 6 is an immediate consequence of Proposition 21 and that fact that each ACADF is an ADF, that is, $\Sigma^{\sigma}_{\text{ACADF}} \subseteq \Sigma^{\sigma}_{ADF}$ for each semantics σ.

Corollary 6. *For* $\sigma \in \{adm, prf, stb, mod, com, grd\}$ *it holds that,*

$$\Sigma^{\sigma}_{ACADF} \subsetneq \Sigma^{\sigma}_{ADF}$$

Corollary 7. $\Sigma^{\sigma}_{\beta} \nsubseteq \Sigma^{\sigma}_{ACADF}$, *for* $\sigma \in \{adm, prf, stb, mod, grd, com\}$ *and* $\beta \in \{ASADF, ASSADF, BADF\}$.

Proof. The ADF $F = (\{a, b\}, \{\varphi_a : \neg b, \varphi_b : \neg a\})$ which is used in the proof of Proposition 21 to show that $\mathbb{S}$ (resp. $\mathbb{S}'$) is σ-realizable in ADF, for $\sigma \in \{adm, prf, stb, mod, grd, com\}$, and is not σ-realizable in ACADFs is also an ASADF, an ASSADF and a BADF. Therefore, $\Sigma^{\sigma}_{\beta} \nsubseteq \Sigma^{\sigma}_{\text{ACADF}}$, for $\sigma \in \{adm, prf, stb, mod, grd, com\}$ and for $\beta \in \{\text{ASADF}, \text{ASSADF}, BADF\}$. □

Since each ACADF does not contain any dependent link, each ACADF is a BADF. Therefore, we immediately conclude that $\Sigma^{\sigma}_{\text{ACADF}} \subseteq \Sigma^{\sigma}_{BADF}$ for $\sigma \in \{adm, prf, com, grd, stb, mod\}$. All of these properties except admissibility hold for ASADF and ASSADF which are shown in Proposition 22. In Proposition 23 it is illustrated that there exists an interpretation-set which is *adm*-realizable in ACADF but not in ASADF and ASSADF.

Proposition 22. *$\Sigma^{\sigma}_{ACADF} \subsetneq \Sigma^{\sigma}_{\beta}$, for $\sigma \in \{prf, com, grd, stb, mod\}$ and $\beta \in \{ASADF, ASSADF\}$.*

Proof. Let $\mathbb{S} = \{S\}$ be an interpretation-set which is σ-realizable in ACADF, for $\sigma \in \{prf, com, grd, stb, mod\}$. Since each ASADF is an ASSADF, we only show that $\mathbb{S}$ is σ-realizable in ASADF, for $\sigma \in \{prf, com, grd, stb, mod\}$. Construct $F = (A, L, C)$ as follow:

- A contains all arguments which appear in S.
- If $a \mapsto t \in S$ then $\varphi_a : \top \in C$.
- If $a \mapsto f \in S$ then $\varphi_a : \bot \in C$.

It is obvious that F is an ASADF and $\sigma(F) = \mathbb{S}$. That is, $\mathbb{S} \in \Sigma^{\sigma}_{\text{ASADF}}$, for $\sigma \in \{prf, com, grd, stb, mod\}$. □

Proposition 23. *$\Sigma^{adm}_{ACADF} \not\subseteq \Sigma^{adm}_{\beta}$, for $\beta \in \{ASADF, ASSADF\}$.*

Proof. Let $\mathbb{S} = \{\{a \mapsto u, b \mapsto u\}, \{a \mapsto t, b \mapsto u\}, \{a \mapsto t, b \mapsto t\}\}$ be an interpretation-set. A witness of *adm*-realizability of $\mathbb{S}$ in ACADF is $F = (\{a, b\}, \{\varphi_a : \top, \varphi_b : a\}$. We claim that $\mathbb{S}$ is neither *adm*-realizable in ASADF nor in ASSADF. Since each ASADF is an ASSADF we show the result only for ASSADF. Suppose to the contrary that $\mathbb{S}$ is *adm*-realizable in ASSADF by $F = (A, L, C)$. Therefore, $A = \{a, b\}$ otherwise, the additional argument has to appear in $adm(F)$. If a and b are isolated arguments in all the ways that one can define their acceptance conditions, $adm(F) \neq \mathbb{S}$. Assume that a and b are not isolated arguments then there are two possibilities to define their acceptance conditions as follows:

- If $\varphi_a : \neg b$ and $\varphi_b : \neg a$ then $adm(F) = \{\{a \mapsto u, b \mapsto u\}, \{a \mapsto t, b \mapsto f\}, \{a \mapsto f, b \mapsto t\}\}$.
- If $\varphi_a : b$ and $\varphi_b : \neg a$ (resp. $\varphi_a : \neg b$ and $\varphi_b : a$) then $adm(F) = \{\{a \mapsto u, b \mapsto u\}\}$.

In both cases $adm(F) \neq \mathbb{S}$. Hence, $\mathbb{S}$ is not *adm*-realizable in ASADF and ASSADF. □

In Section 4.3.1 we studied expressiveness of subclasses of ADFs namely, ACADFs, ASADFs, ASSADFs. For instance, it showed in Theorem 14 that $\Sigma^{\sigma}_{\text{ASADF}} \subsetneq \Sigma^{\sigma}_{\text{ASSADF}} \subsetneq \Sigma^{\sigma}_{BADF} \subsetneq \Sigma^{\sigma}_{ADF}$ for $\sigma \in \{adm, prf, com\}$. In the following we focus on some exceptions like the ones we had in the previous section. For instance, it is proved in Proposition 7 and 8 that whenever $\mathbb{S}$ is an extension-set which is *stb*-realizable in AFs and $|\mathbb{S}| \leq 2$, it is *stb*-realizable in SYMAFs. In the following it is investigated whether this property carries over to ADFs and its subclasses.

Proposition 24. *Suppose that* $|\mathbb{S}| = 1$ *and* $\mathbb{S}$ *is stb-realizable in ADFs. Then,* $\mathbb{S}$ *is stb-realizable in ACADFs.*

Proof. Since $|\mathbb{S}| = 1$, let S be the unique element of $\mathbb{S}$. We claim that $F = (A, L, C)$ in which A and C are defined as follows is a witness of *stb*-realizabilty of $\mathbb{S}$ in ACADFs.

- Let A be the set of all arguments of S.
- For each $a \in A$:
  - If $a \mapsto t \in S$ then $\varphi_a \equiv \top$,
  - If $a \mapsto f \in S$ then $\varphi_a \equiv \bot$.

By our assumption, since $\mathbb{S}$ is *stb*-realizable in ADFs then each argument is either assigned to t or f by S. Therefore, if $a \mapsto t \in S$ then $\varphi_a \equiv \top \in C$ and if $a \mapsto f \in S$ then $\varphi_a \equiv \bot \in C$. Then, $stb(F) = \mathbb{S}$. □

With the same proof method of Proposition 24, Proposition 25 is also provable.

Proposition 25. *Let* $\mathbb{S}$ *be an interpretation-set s.t.* $|\mathbb{S}| = 1$ *and* $\mathbb{S}$ *be mod-realizable in ADFs.* $\mathbb{S}$ *is mod-realizable in ASADFs.*

Corollary 8. *Let* $\mathbb{S}$ *be an interpretation-set s.t.* $|\mathbb{S}| = 1$ *and* $\mathbb{S}$ *be stb-realizable in ADFs. Then,* $\mathbb{S}$ *is stb-realizable in ASADFs, ASSADFs and BADFs.*

Corollary 9. *Let* $\mathbb{S}$ *be a two-valued interpretation-set s.t.* $|\mathbb{S}| = 1$. *Then,* $\mathbb{S} \in \Sigma_{\alpha}^{stb}$, *for* $\alpha \in \{ADF, BADF, ASADF, ASSADF, ACADF\}$.

We know by Theorem 9 that in general Σ_{ADF}^{stb} is a strict superset of Σ_{AF}^{stb}. By Corollary 9 we know that whenever the cardinality of an interpretation-set $\mathbb{S}$ is one and it is *stb*-realizable in α, for $\alpha \in \{ADF, BADF, \text{ASADF}, \text{ASSADF}, \text{ACADF}\}$ then it is *stb*-realizable in β, for $\beta \in \{ADF, BADF, \text{ASADF}, \text{ASSADF}, \text{ACADF}\}$. In the following we investigate whether we can generalize Corollary 9 for AFs. That is, whether $|\mathbb{S}| = 1$ and it is *stb*-realizable in ADFs then it is *stb*-realizable in AFs. We investigate in Example 39 that there exists an interpretation-set with cardinality one which is *stb*-realizable in ADFs but not in AFs. That is, even for the class of interpretation-sets with cardinality one Σ_{ADF}^{stb} is a strict superset of Σ_{AF}^{stb}.

Example 39. Let $\mathbb{S} = \{\{a \mapsto f, b \mapsto f, c \mapsto f\}\}$. A witness of *stb*-realizability of $\mathbb{S}$ in ADFs is $F = (\{a, b, c\}, \{\varphi_a : c, \varphi_b : a, \varphi_c : b\})$. We claim that $\mathbb{S}$ is not *stb*-realizable in AFs. Towards a contradiction, assume that there exists an AF $F = (A, L)$ s.t. $stb(F) = \mathbb{S}$. $A = \{a, b, c\}$, otherwise an argument has to appear in a stable model of F. None of these arguments can be an isolated argument, otherwise, it is assigned to t in a stable model. It is easy to check for all way of defining links among a, b and c, $stb(F) \neq \mathbb{S}$. Hence, $\mathbb{S}$ is not σ-realizable in AFs.

In the following we investigate whether an interpretation-set with cardinality two which is *stb*-realizable in ADFs is *stb*-realizable in subclasses of ADFs. However, it is proven by Proposition 20 that whenever $|\mathbb{S}| > 1$, $\mathbb{S}$ is not *stb*-realizable in ACADF. It is shown bellow in Proposition 26 that if $|\mathbb{S}| = 2$ and it is *stb*-realizable in ADFs it is *stb*-realizable in other subclasses of ADFs explained in this study.

Proposition 26. *Suppose that* $|\mathbb{S}| = 2$ *and* $\mathbb{S}$ *is stb-realizable in ADFs. Then* $\mathbb{S}$ *is stb-realizable in ASADFs.*

Proof. Let $\mathbb{S} = \{v_1, v_2\}$. By the assumption $\mathbb{S}$ is *stb*-realizable in ADFs then there exists an ADF $D = (A, L, C)$ s.t. $stb(D) = \mathbb{S}$. Construct an ADF $F = (A', L', C')$ as follows:

- Let $A' = A$.
- For each $a \in A'$:
  - If $a \mapsto t \in v_1 \cap v_2$ (resp. $a \mapsto f \in v_1 \cap v_2$) then let $\varphi_a : \top \in C'$ (resp. $\varphi_a : \bot \in C'$).
  - otherwise, If $a \mapsto t \in v_1$ (resp. $a \mapsto f \in v_1$) then let $\varphi_a : \bigwedge_{a_i \mapsto f \in v_1} \neg a_i$ (resp. $\varphi_a : \bigwedge_{a_i \mapsto t \in v_1} \neg a_i$).

Since $\mathbb{S}$ is *stb*-realizable in ADFs it is not the case that all arguments are assigned to t by either v_1 or v_2. Hence, by the definition of C', F is an ASADF. In addition, $stb(F) = \mathbb{S}$. That is, $\mathbb{S}$ is *stb*-realizable in ASADFs. □

By the construction of acceptance conditions in the proof of Proposition 26 each constructed ADF F is not only a ASADF but also correspondence to a SYMAF. Therefore, whenever, an interpretation-set $\mathbb{S}$ with cardinality two is *stb*-realizable in ADFs it is also *stb*-realizable in SYMAFs and in AFs as a consequence. However, Example 39 shows that there exists an interpretation-set $\mathbb{S}$ with cardinality one which is *stb*-realizable in ADFs and it is not *stb*-realizable in AFs.

Corollary 9, Proposition 26 and the fact that each ASADF is an ASSADF and a BADF yield the following corollary.

Corollary 10. *Suppose that* $|\mathbb{S}| \leq 2$ *and* $\mathbb{S}$ *is stb-realizable in ADFs. Then* $\mathbb{S}$ *is stb-realizable in* α*, for* $\alpha \in \{ASADF, ASSADF, BADF\}$.

In the following we investigate whether there is a corresponding result for two-valued models as stable models proven in Proposition 26. That is, we illustrate in Example 40 that it is not the case that whenever the cardinality of interpretation-set is two and it is *mod*-realizable in ADFs then it is *mod*-realizable in ASADFs.

Example 40. Let $\mathbb{S} = \{\{a \mapsto t\}, \{a \mapsto f\}\}$. It is clear that $\mathbb{S}$ is *mod*-realizable in ADFs by $F = (\{a\}, \{\varphi_a : a\})$ and it is not realizable by any ASADFs.

The same interpretation-set $\mathbb{S} = \{\{a \mapsto t\}, \{a \mapsto f\}\}$ of Example 40 is used to show that there exists an interpretation-set with cardinality two which is *prf*-realizable in ADFs but not in ASADFs.

4.4 Summary

In the beginning of this chapter first we focused on AFs and its subclass SYMAFs. The relations between signatures of SYMAFs for semantics, $\sigma \in \{adm, cf, prf, stb, nai\}$ were studied in Theorem 10, depicted in Figure 4.4. For instance, it is shown that while Σ_{AF}^{adm} is a strict superset of Σ_{AF}^{cf} by [25] , the signature of SYMAFs under admissible and conflict-free semantics are equivalent.

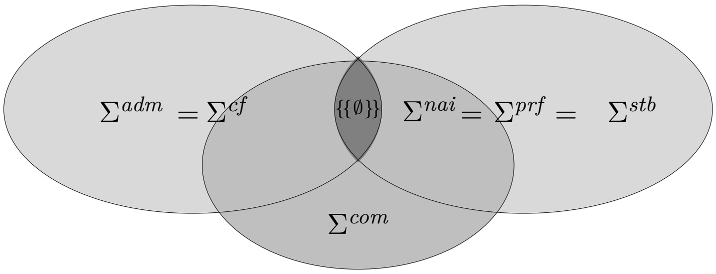

Figure 4.4: Relations between signatures of SYMAFs shown in Theorem 10

Next we investigated expressiveness of AFs in comparison with SYMAFs. In Theorem 11 depicted in Figure 4.5 we have shown that for $\sigma \in \{adm, prf, stb, com\}$ the signatures of AFs are strict supersets of the signatures of SYMAFs while for $\sigma \in \{cf, nai, grd\}$ both of them are equivalent.

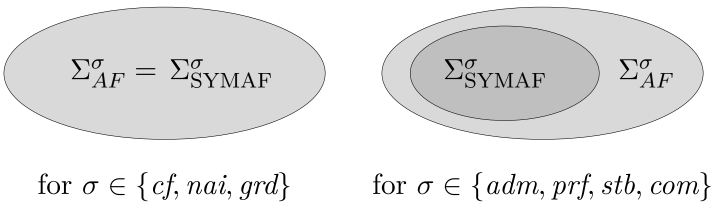

Figure 4.5: Relations between signatures of SYMAFs and AFs shown in Theorem 11

In the following of this section we concentrated on the cardinality of an extension-set and we investigated some results of σ-realizability in SYMAFs based on cardinality of extension-sets which do not hold in general. For instance, it is proven in Theorem 11 that Σ_{AF}^{stb} is a strict superset of $\Sigma_{\text{SYMAF}}^{stb}$, while whenever the cardinality of an extension-set is two, it is *stb*-realizable in AFs if and only if it is *stb*-realizable in SYMAFs.

Via Section 4.3.1 we studied the expressiveness of subclasses of ADFs. The main results which are proven in Theorems 12, 13, 14 and Proposition 19 are depicted in Figure 4.6, for $\sigma \in \{adm, com, prf\}$.

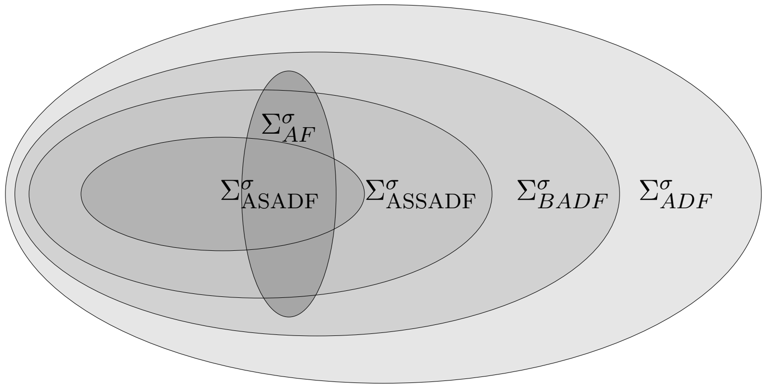

Figure 4.6: Expressivity of subclasses of ADFs shown in Theorem 14 for $\sigma \in \{adm, prf, com\}$

In addition, it is proven that, while signatures of ASADFs and ASSADFs are incomparable with signatures of AFs in most of the semantics, namely for $\sigma \in \{adm, com, prf\}$, Σ^{stb}_{AF} is a strict subset of $\Sigma^{stb}_{\text{ASADF}}$. It is shown by [39] that Σ^{stb}_{AF} is a strict subset of Σ^{stb}_{BADF}. However, in Proposition 16 it is proven that $\Sigma^{stb}_{AF} \setminus \{v_\epsilon\}$ is a strict subset of $\Sigma^{stb}_{\text{ASADF}}$. Thus, consequently, Σ^{stb}_{AF} is a strict subset of $\Sigma^{stb}_{\text{ASSADF}}$ and Σ^{stb}_{BADF}.
Moreover, the expressiveness of ACADFs in comparison to ADFs, ASADFs and ASSADFs is also studied in this work. We showed that signatures of ACADFs are strict subsets of signatures of ASSADFs, and in consequence they are strict subsets of signature of ASSADFs and signature of ADFs for $\sigma \in \{prf, com, grd, stb, mod\}$. While the signature of ACADFs are also a strict subset of the signature of ADFs for admissible semantics as well, $\Sigma^{adm}_{\text{ACADF}} \nsubseteq \Sigma^{adm}_{\beta}$, for $\beta \in \{\text{ASADF}, \text{ASSADF}\}$. At the end of this section we again focused on the cardinality of an interpretation-set and for instance, we have shown that an interpretation-set with cardinality 2 is *stb*-realizable in ADFs if and only if it is *stb*-realizable in ASADFs.

CHAPTER 5

Experiments on Subclasses of ADFs

In this chapter we change focus, and report on a preliminary investigation on the extent to which current solvers for ADFs are affected by their inputs being restricted to certain subclasses of ADFs. For this, we first adapted an existing generator for ADFs so that it generates acyclic, attack symmetric and acyclic support symmetric ADFs inhereting the structure of any arbitrary undirected graph. Secondly, we carried out experiments to determine the performance of existing solvers for ADFs on acyclic vs non-acylic ADFs generated via our generator.

The first section of this chapter describes the functioning (Section 5.1) as well as the use (Section 5.1.2) of the generator we implemented. In Section 5.2 we report on the experimental setup (Section 5.2.2) and the results of our experiments (Section 5.2.3), after having given a very brief survey on current existing solvers for ADFs (Section 5.2.1).

5.1 A generator for subclasses of ADFs

5.1.1 Description of the generator

This section focuses on the description of the generator by which acyclic ADFs, attack symmetric ADFs (ASADFs) and acyclic support symmetric ADFs (ASSADFs) are generated for the input undirected graph s.t. the generated ADF inheriting the structure of the input graph except that it is irreflexive. As indicated previously, the generator on which we report here builds on an existing generator for ADFs used in the experiments reported in [19]. We have modified the generator reported in [19] so that it expects undirected graphs as input. The generated ADFs then inherit the structure of the input graphs, i.e. arguments correspond to nodes and attacks to edges in the input graph. We also build on the generator in [19] in constructing the acceptance conditions associated

to the arguments. Arguments without parents (in the input graph) have as acceptance conditions either $\top$ or $\bot$ with equal probability. Otherwise, parents of each argument s in the input graph are first assigned to one of five different groups: attackers, group-attackers, supporters, group-supporters and the XOR group denoted by A, B, S, T and X, respectively. This is done with some probability that can be given by the user; the default is equal probability. The group to which a parent of s is assigned, determines how it will appear in the acceptance condition of s. Attackers appear negated and connected via conjunction. Group-attackers appear negated and connected via disjunction. Supporters appear as positive (not negated) literals connected via disjunction, while group-supporters appear positively and connected via conjunction. Finally, parents in the XOR group are connected via XOR; they appear positively or negatively with equal probability.
In the following we detail how we adapt the generator reported in [19] for our purposes.

To compute an acyclic ADF:

1. A total order on the vertices of the input graph is chosen randomly.
2. An acyclic directed graph is produced based on the total order.
3. Each parent of an argument is assigned to one of the 5 groups explained above with equal probability.
4. The acceptance condition of each argument is constructed following [19].

To compute an ASADF:

1. All self-loops of the given graph are removed.
2. Let a and b be two arbitrary nodes of an edge of the input graph. By the assumption the input graph is undirected. Hence, $a \in par(\varphi_b)$ and $b \in par(\varphi_a)$.
3. Since each parent in an ASADF is an attacker, it could either belong to attacks or group-attacks.
4. The acceptance condition of each argument is constructed following [19].

To compute the ASSADF:

1. A total order is picked for vertices of the input graph.
2. A directed graph is produced based on the total order. Note that, this directed graph is used to choose supporter nodes.
3. Those parents of an argument appearing as parents also in the support-group are assigned to any of the five groups: attacks, group-attacks, support, group-support, XOR.

4. Those parents not being parents in the support group (the one which are not chosen in previous part) can only be attacks or group-attacks.
5. The acceptance condition of each argument is constructed following [19].

Our generator has been implemented using the programming language Scala [31]; Scala programs can be compiled into Java executables.

5.1.2 Use of the generator

We make the generator, adfgen (version 0.2) available, as a java executable [1]. To see all the options with which the generator can be executed (some of these are not relevant for our purposes), adfgen can be called as follows:

java -jar adfgen-0.2.0.jar -h

The input graph is specified via the option -I inputFile where inputFile is a file containing the description of the graph. We refer to the representation of the graph later on.
There are two main types of options for calling adfgen that are relevant for our work. The first type of options control the generation of the acceptance conditions while the others are used to transform the input graph. In the case of the acceptance conditions, probabilities with which parents are assigned to one of the different groups detailed in Subsection 5.1 are set via the handle -A pA -S pS -B pB -T pT -X pX where A, S, B, T, X stand for attackers, supporters, group-attackers, group-supporters, XOR-group and pA, pS, pB, pT, pX refer to the probabilities.
Regarding the construction of the graph, it can be controlled via the following options:

-acyc make (undirected) graph generated via input file acyclic.

-nslf remove self-loops from (undirected) graph generated from input file.

-supacyc make ASSADFs which may not irreflexive.

Our design of adfgen enables a user to generate not only acyclic ADFs, ASADFs and ASSADFs but also other desired ADFs by changing the input parameters.
There are some other options like -G used to generate a random ADF based on the input graph which we do not go into details, since they are not related to our implementation.
The options -c and -d are used to determine the format of the output (these are optional).

-c generate integer identifiers for arguments in the input graph.

-d print debug information.

[1] https://www.dbai.tuwien.ac.at/proj/grappa/subadfgen

In the following we give examples of the use of adfgen to generate arbitrary ADFs as well as acyclic ADFs, ASADFs and ASSADFs.

- An arbitrary ADF can be constructed as follows:

  java -jar adfgen-0.2.0.jar -I [given graph] -A [pA] -S[pS] -B [pB] -T [pT] -X [pX] [options]

  The parameter [given graph] is the file name where the input graph is specified. It is mandatory to specify the file name. [p–] could be any arbitrary number between 0 and 1. It is used to determine the probability of assigning parents to groups. Note that the sum of the probability of all groups has to be one. Although, if they are not denoted in command line the probability of all groups are assumed equal by default. [options] can be either -c, -d or none of them.

- An acyclic ADF can be generated as follows:

  java -jar adfgen-0.2.0.jar -I [given graph] -acyc -A [pA] -S [pS] -B [pB] -T [pT] -X [pX] [options]

  To generate an acyclic ADF [given graph] and -acyc are mandatory. Just as for generating an arbitrary graph assigning probabilities parents to groups is optional.

- An ASADF can be generating using:

  java -jar adfgen-0.2.0.jar -I [given graph] -nslf -A [p] -S 0.0 -B [p] -T 0.0 -X 0.0 [options]

  Since we desire to generate ASADFs, the probabilities of -S, -T and -X are 0.0. In addition, choosing the probability of A and B must be specified by the user. Moreover, using -nslf in the command eliminates all self-loops in the input graph. Hence, to generate an ASADF using -nslf is necessary.

- An ASSADF can be generated by using:

  java -jar adfgen-0.2.0.jar -I [given graph] -nslf -supacyc [options]

  Having [given graph], -nslf and -supacyc are necessary to generate an ASSADF. This command assigns parents also appearing as parents in the support or support-group to one of the five groups with equal probability (if one wants, it is also possible to control the probabilities). Parents which do not appear as parents in supports or support-groups are considered attackers or group-attackers with equal probability.

Input

To represent the input graph, the input file should contain an expression (relation schema) $"edge("s", "t")."$ and an expression $"vertex("s")."$ in which s and t are strings representing vertices of the graph. An expression $"edge("s", "t")."$ in the input file means that there is an edge between the two vertices s and t in the given graph. It is not necessary to list all vertices appearing in edges in the graph, that is only necessary for isolated nodes.
As mentioned previously the input graph is assumed to be undirected. If a user writes both $"edge("a", "b")."$ and $"edge("b", "a")."$ in the input file the program ignores one of them without any error being indicated.

Output

The format of the output of adfgen has been chosen to be compatible with the input of existing solver systems for ADFs. The output of the program contains propositional formulas each of which represents the acceptance condition of an argument of the ADF. The output file contains an expression $"s(s)."$ and an expression $"ac(s, acceptance - condition)."$ where $"s"$ is the name of the argument and $"acceptance - condition"$ is the acceptance condition associated to the argument with identifier s. In the following different symbols which may occur in the output file are described:

> $"ac(a, c(v))"$ (resp. $"ac(a, c(f))"$) denotes that the acceptance condition of argument a is equivalent to $\top$ (resp. $\bot$).
>
> The symbols for the connectives used in the output file are $"neg"$ for $\neg$, $"and"$ for $\wedge$, $"or"$ for $\vee$ and $"xor"$ for XOR.

Example 41. The input format of the graph which is depicted in Figure 5.1 is as follows:

$$\begin{aligned} edge("b", "e"). \\ edge("c", "a"). \\ edge("a", "d"). \\ edge("d", "e"). \\ edge("a", "b"). \\ edge("e", "e"). \\ edge("b", "b"). \\ vertex("f"). \end{aligned}$$

A possible acyclic ADF generated by adfgen from input graph of Example 41, depicted in Figure 5.2, is as follows:

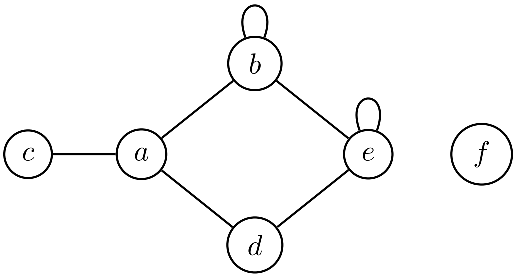

Figure 5.1: Input graph of Example 41

```
s(a).
s(e).
s(d).
s(b).
s(c).
s(f).
ac(a,and(neg(d),or(neg(c),b))).
ac(e,d).
ac(d,c(f)).
ac(b,neg(e)).
ac(c,c(f)).
ac(f,c(f)).
```

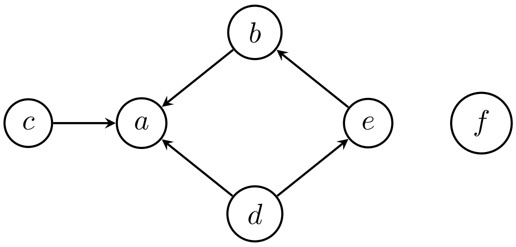

Figure 5.2: Acyclic ADF inheriting the structure of the graph of Figure 5.1

The attack symmetric ADF, depicted in Figure 5.3, is a possible output of adfgen for the graph depicted in Figure 5.1.

```
s(a).
s(e).
s(d).
s(b).
s(c).
s(f).
ac(a,or(or(neg(c),neg(d)),neg(b))).
ac(e,or(neg(b),neg(d))).
```

```
ac(d,and(neg(a),neg(e))).
ac(b,or(neg(a),neg(e))).
ac(c,neg(a)).
ac(f,c(v)).
```

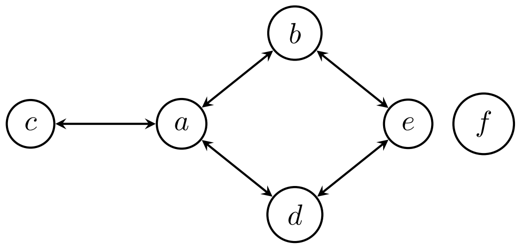

Figure 5.3: Attack symmetric ADF inheriting the structure of the graph of Figure 5.1

Since each ASADF is an ASSADF, the graph depicted in Figure 5.3 could be supposed as an ASSADF output of Example 41. The graphs corresponding to ASADFs and ASSADFs generated by adfgen for the same input graph are equal except on the acceptance conditions. However, there exist ASSADFs for a given graph which are not ASADFs. A possible ASSADF inheriting the structure of the graph of Figure 5.1 which is not an ASADF is as the follows.

```
s(a).
s(e).
s(d).
s(b).
s(c).
s(f).
ac(a,or(and(neg(c),neg(d)),b)).
ac(e,or(b,d)).
ac(d,or(neg(a),neg(e))).
ac(b,and(neg(a),neg(e))).
ac(c,a).
ac(f,c(f)).
```

For instance, in the above ASSADF, $v_i(\varphi_a) = t$ for $v_1 = \{b \mapsto t, c \mapsto f, d \mapsto f\}$, $v_2 = \{b \mapsto f, c \mapsto f, d \mapsto f\}$, $v_3 = \{b \mapsto t, c \mapsto f, d \mapsto t\}$, $v_4 = \{b \mapsto t, c \mapsto t, d \mapsto f\}$ and $v_5 = \{b \mapsto t, c \mapsto t, d \mapsto t\}$. In addition, $v_i|_t^b(\varphi_a) = t$ for $1 \leq i \leq 5$. This means, b is a supporter of a. Hence, the above ADF cannot be an ASADF.

5.2 The effect of cycles on the performance of solvers for ADFs

5.2.1 Overview of ADF systems

Given the special role of abstract argumentation in artificial intelligence and the fact that there are many reasoning problems with high computational complexity, implementation methods are an important research topic. In [21] an overview of different methods for solving abstract argumentation tasks on Dung style framework are given. Two main approaches explained in [21] are reduction based approaches and direct approaches. Only the former method has been taken up for abstract dialectical frameworks. Reduction based approaches for ADF can be classified according to the target-formalism: answer set programming (ASP) and Qualified Boolean (QADFs). Among the mentioned methods answer set programming (ASP) has been particularly a significant.

The first system for ADFs, ADFSys [27] is already based on ASP. DIAMOND [27, 26] is the successor of ADFSys. There are several versions of DIAMOND by now [2]. Since DIAMOND relies on static encodings, it is limited by the data complexity of ASP. A more recent system for ADFs, YADF [3] [19], uses dynamic encodings.

As mentioned previously, the only other existing approach to implementing reasoning on ADFs, implemented in the system QADF [4] [23], is via reduction to quantified boolean formulas, more specifically to the problem of satisfiability of QBFs, QSAT. This enables using QSAT solvers as reasoning back-ends.

5.2.2 Experimental setup

As indicated in the introduction, we have carried out initial experiments to evaluate the effect of cycles on the performance of existing solvers for ADFs. To do this evaluation first we redefine two decision problems associated to ADFs used in the solvers.

- Credulous acceptance under σ: Let $F = (A, L, C)$ be an ADF and $a \in A$ be an argument. Dose there exist a σ-interpretation v s.t. $v(\varphi_a) = t$?
- Skeptical acceptance under σ: Let $F = (A, L, C)$ be an ADF and $a \in A$ be an argument. Is it the case that $v(\varphi_a) = t$ for all σ-interpretations v?

More specifically, we carried out experiments comparing the performance of 2 different versions of DIAMOND, as well as the recent versions of YADF and QADF on credulous and skeptical acceptance problems for the admissible and preferred semantics. We ran the solvers on acyclic and non-acyclic ADFs generated by our ADF generator, adfgen.

[2] `http://diamond-adf.sourceforge.net/`
[3] `http://www.dbai.tuwien.ac.at/proj/adf/yadf/`
[4] `https://www.dbai.tuwien.ac.at/proj/adf/qadf/`

For the experiments we generated ADFs using 8 different metro networks as an input graphs. 10 different acyclic ADFs, resp. non-acyclic ADFs, are generated for each input graph. Therefore, in total 160 instances are generated, 80 instances for acyclic ADFs and 80 instances for non-acyclic ADFs.
For DIAMOND we used the last version which is known to have no errors, 0.9, as well as a recent version, goDIAMOND, which has been submitted to the second international competition for computational models of argumentation (ICCMA 2017) [5]. For YADF we used version 0.1.0, with the rule decomposition tool lpopt [10]. For QADF we used version 0.3.2 with the preprocessing tool bloqqer035 [11] and the QSAT solver DepQBF, version 4.0 [30]. Both for DepQBF and YADF we used clingo version 4.4.0.

All experiments were carried out on an Debian (8.5) machine with eight IntelXeon processors (2.33 GHz) and 48 GB of memory. In our experiments we chose a time-out of 10 minutes for each run of the reasoners on the different ADF instances.

5.2.3 Result of the experiment

It is shown in Table 5.1 that in order to decide credulous acceptance for the admissible semantics for both acyclic ADFs and non-acyclic ADFs in all systems except QADF there are no time-outs. There are significant number of time-outs for QADF on non-acyclic ADFs. goDIAMOND fares better regarding average running time for both acyclic and non-acyclic instances. For all systems, there is a substantial improvement in performance for acyclic instances.

		Time-outs	Mean
adm-acyclic	DIAMOND	0	5.2813
	goDIAMOND	0	0.0667
	YADF	0	1.5727
	QADF	4	3.1635
adm-non-acyclic	DIAMOND	0	17.641
	goDIAMOND	0	0.202
	YADF	0	2.1679
	QADF	35	5.5060

Table 5.1: Number of time-outs and Mean running times in seconds for credulous acceptance under admissible semantics.

Figure 5.4 represents the number of acyclic and non-acyclic ADF-credulous-acceptance-problems solved in running time x second s.t. $(0 < x \leq 600)$ by DIAMOND, goDIAMOND, YADF and QADF for the admissible semantics.

Table 5.2 shows that the performance of all systems for skeptical acceptance for the preferred semantics for both acyclic and non-acyclic ADFs are not so good as for credulous

[5] https://www.dbai.tuwien.ac.at/iccma17/

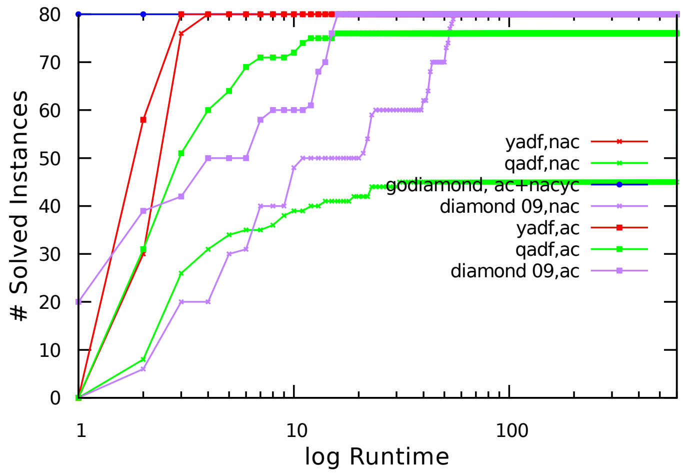

Figure 5.4: Number of instances solved in running time less than x second $(0 < x \leq 600)$ for acyclic and non-acyclic reasoning under admissible.

acceptance problems for the admissible semantics. Systems DIAMOND 0.9 and QADF have time-outs on all instances. The system goDIAMOND again fares much better than the other systems. It timed-out only on 8 instances for solving non-acyclic ADFs and was able to solve all acyclic ADFs in a very short time.

		Time-outs	Mean
prf-acyclic	DIAMOND	80	–
	goDIAMOND	0	0.144
	YADF	57	90.696
	QADF	80	–
prf-non-acyclic	DIAMOND	80	–
	goDIAMOND	8	1.2838
	YADF	40	126.128
	QADF	80	–

Table 5.2: Number of time-outs and mean running times in seconds for skeptical acceptance under the preferred semantics

Figure 5.5 represents the number of acyclic and non-acyclic ADF skeptical acceptance problems solved by YADF and goDIAMOND in running time x second $(0 < x \leq 600)$ under the preferred semantics. Since DIAMOND and QADF timed-out on all instances they are not depicted in this figure. In Figure 5.5 it seems that YADF under preferred works better on non-acyclic ADFs than acyclic ADFs. The reason is that YADF solves only on 23 acyclic ADFs, however it solves 40 non-acyclic ADFs. Table 5.2 illustrates that the mean running time of YADF on acyclic ADFs under preferred is better than for non-acyclic ADFs. That is, mean running time of solving acyclic ADFs under both admissible and preferred interpretations in all systems which are used in these experiments is less than the mean running time of solving non-acyclic ADFs.

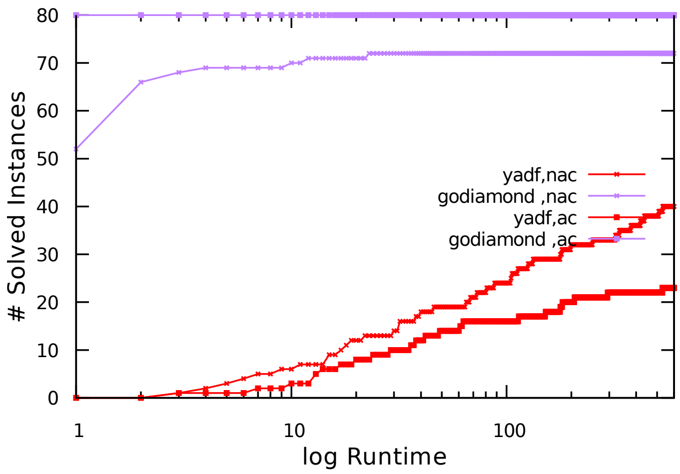

Figure 5.5: Number of instances solved in running time less than x seconds ($0 < x \leq 600$) for acyclic and non-acyclic reasoning under preferred.

It is proven by Theorem 3 that each acyclic ADF contains a unique preferred interpretation. Therefore, for acyclic ADFs each argument is credulously accepted under preferred semantics if and only if it is skeptically accepted under preferred semantics. That is, by uniqueness of preferred interpretation in acyclc ADFs, credulous acceptance and skeptical acceptance are equivalent for acyclic ADFs. In addition, credulous acceptance for admissible semantics and preferred semantics are the same. Hence, in acyclic ADFs an argument is credulously accepted under admissible semantics iff it is skeptically accepted under preferred semantics. Therefore, these two decision problem have the same complexity, theoretically. However, the experiments on which we reported in this section show that there is a significant different between the results of deciding credulous acceptance for the admissible semantics and deciding skeptical acceptance for the preferred semantics for acyclic ADFs in practice.

On the other hand, while the experiments do not clearly suggest that current ADF solvers perform better for acyclic ADFs when making use of encodings for the skeptical acceptance for the preferred semantics, it is possible to conclude that there is some difference in the case of the encodings for the admissible semantics. The latter may be due to the fact that acyclicity of ADFs is recognized ”under the hood” at least for the admissible semantics, i.e. by the ASP solvers, pre-processing, rule-decomposition or the QBF solver. Further experiments would be necessary to show exactly what causes the solvers to perform better for the admissible semantics, as well as whether also for the preferred semantics encodings there is some improvement in performance.

CHAPTER 6

Conclusion and Future Work

6.1 Summary

A motivation of the current work was to cover some of the gaps in the understanding of AFs and ADFs. For instance, to the best of our knowledge there is no equivalent of Dung's fundamental Lemma for ADFs; thus, we reformulated this important lemma and proved that it holds for ADFs in the background of this work. Also, Dung [24] showed that well-founded AFs are equivalent when evaluated under the complete, grounded, preferred, and stable semantics. In Section 3.1 we showed that the same semantics (plus the two-valued) coincide for acylic ADFs.

A further topic in this work was to verify whether properties that are known to hold for symmetric AFs [22] also hold for subclasses of ADFs. While symmetric AFs are coherent and relatively grounded, we showed that symmetric ADFs are neither coherent nor relatively grounded. To deepen our investigation on this issue we introduced more fine-grained subclasses of ADFs: attack symmetric ADFs, acyclic support symmetric ADFs and complete ADFs. We were then able to prove that, for example, attack symmetric ADFs and acyclic support symmetric ADFs are weak-coherent.

We also considered properties of symmetric AFs which are not studied in [22], but are relevant to understand the relative expressivity of subclasses of AFs. For instance, we showed that the admissible and complete semantics are not equivalent for symmetric AFs. We also studied similar questions for ADFs. For example, we were able to prove that for complete ADFs the admissible and complete semantics coincide. All results of our study on properties of subclasses of AFs and ADFs are summarised in Tables 3.1 and 3.2.

A central aspect of our work has been to compare the expressivity of subclasses of AFs and ADFs from the perspective of realizability. In the first section of Chapter 4, we showed that Σ^{σ}_{AF} is more expressive than $\Sigma^{\sigma}_{\text{SYMAF}}$ for $\sigma \in \{adm, prf, stb, com\}$ and that they are equivalent for $\sigma \in \{cf, nai, grd\}$. Then we showed some general results: for instance, while Σ^{σ}_{AF} is more expressive than $\Sigma^{\sigma}_{\text{SYMAF}}$ for $\sigma \in \{prf, stb\}$, each extension-set with cardinality two which is σ-realizable in AFs is also σ-realizable in SYMAFs for $\sigma \in \{prf, stb\}$. Moreover, it was shown that any extension-set with cardinality two is not *com*-realizable in SYMAFs.

One of the main result of Section 4.3 is that: $\Sigma^{\sigma}_{\text{ASADF}} \subsetneq \Sigma^{\sigma}_{\text{ASSADF}} \subsetneq \Sigma^{\sigma}_{BADF} \subsetneq \Sigma^{\sigma}_{ADF}$ for $\sigma \in \{adm, com, prf\}$. In addition, we showed that Σ^{mod}_{BADF} is a strict superset of $\Sigma^{mod}_{\text{ASSADF}}$. While $\Sigma^{mod}_{\text{ASSADF}}$ is a superset of $\Sigma^{mod}_{\text{ASADF}}$, we were not able to determine whether $\Sigma^{mod}_{\text{ASSADF}} \neq \Sigma^{mod}_{\text{ASADF}}$. [39] showed that Σ^{stb}_{BADF} is a strict superset of Σ^{stb}_{AF}; we proved that, while Σ^{σ}_{AF} is incomparable with $\Sigma^{\sigma}_{\text{ASADF}}$, and $\Sigma^{\sigma}_{\text{ASSADF}}$ for $\sigma \in \{adm, prf, com\}$, $\Sigma^{stb}_{AF} \setminus \{v_\epsilon\}$ is a strict subset of $\Sigma^{stb}_{\text{ASADF}}$ and consequently is a strict subset of $\Sigma^{stb}_{\text{ASSADF}}$ and Σ^{stb}_{BADF}. However, it remains to be seen whether the relations among $\Sigma^{stb}_{\text{ASADF}}$, $\Sigma^{stb}_{\text{ASSADF}}$ and Σ^{stb}_{BADF} are strict.

Although all ADFs contain the unique grounded interpretation there is no guarantee that it is *grd*-realizable in ACADFs. For instance, an interpretation-set $\mathbb{S} = \{a \mapsto u, b \mapsto u\}$ is *grd*-realizable in ADFs and it is not *grd*-realizable in ACADFs because the unique grounded interpretation of each ACADF is a two valued model by Theorem 3. Moreover, we proved that for most of the semantics, i.e. $\sigma \in \{prf, com, grd, stb, mod\}$ any interpretation-set which is σ-realizable in ACADFs is also σ-realizable in ASADFs and consequently in ASSADFs as well. However, admissibility is an exception. For instance, the interpretation-set $\mathbb{S} = \{\{a \mapsto u, b \mapsto u\}, \{a \mapsto t, b \mapsto u\}, \{a \mapsto t, b \mapsto t\}\}$ is *adm*-realizable in ACADFs but not in ASADFs and ASSADFs. At the end of Section 4.3.1 we proved that whenever the cardinality of an interpretation-set which is *stb*-realizable in ADFs is less than or equal to two, it is *stb*-realizable in ASADFs and ASSADFs.

In the last chapter of this work we reported on a modification of the generator defined in [19] so that, given an undirected graph as input, it is able to produce acyclic ADFs, attack symmetric ADFs, as well as acyclic support symmetric ADFs. We used this generator in order to evaluate the effect of cycles on the the performance of existing solvers for ADFs. Specifically, we carried out experiments to compare the performance of two versions of the solver DIAMOND as well as YADF and QADF on credulous and skeptical acceptance problems for the admissible and preferred interpretations.

As expected, our experiments show a significant improvement in the performance of ADF solvers for acyclic ADFs in comparison to non-acyclic ADFs for credulous acceptance w.r.t. the admissible semantics. On the other hand, in Theorem 3 we showed that each acyclic ADF possesses the unique preferred interpretation. That is, in acyclic ADFs each argument is credulously accepted under the preferred semantics iff it is skeptically accepted. Moreover, credulous acceptance for admissible semantics and preferred semantics are the same. Thus, credulous acceptance for admissible semantics and skeptical acceptance for the preferred semantics in acyclic ADFs have the same computational complexity, theoretically. However, our experiments show a significant difference in deciding credulous acceptance for the admissible semantics and skeptical acceptance for the preferred semantics for acyclic ADFs. This suggests that current solvers for ADFs are not yet tuned to make use of acyclicity for the preferred semantics.

6.2 Future Work

The presented work leaves open various directions for future work.

- There is the work by Polberg [32] which introduced further ADF semantics and in the future we want to check how these semantics behave in the subclasses studied in this thesis.
- The computational complexity of reasoning tasks for ADFs are well-studied [42, 43, 28]. We plan to study the computational complexity of reasoning tasks for the subclasses of ADFs considered in Chapter 3.
- There remain some open questions in regard to expressiveness as considered in Chapter 4. For instance, while it was shown that Σ^{stb}_{BADF} is a superset of $\Sigma^{stb}_{\mathrm{ASSADF}}$ and $\Sigma^{stb}_{\mathrm{ASSADF}}$ is a superset of $\Sigma^{stb}_{\mathrm{ASADF}}$, we still need to determine whether these properties are strict. In addition, the expressivity of a number of further semantics have not been studied yet.
- Finally, the results of our experiments in Chapter 5 suggest that solvers for ADFs can/should be optimised w.r.t. subclasses. Nevertheless, a more detailed experimental evaluations, also with subclasses of ADFs beyond acyclic ADFs, would also be of benefit.

List of Figures

List of Tables

Bibliography

[1] Latifa Al-Abdulkarim, Katie Atkinson, and Trevor J. M. Bench-Capon. A methodology for designing systems to reason with legal cases using abstract dialectical frameworks. *Artif. Intell. Law*, 24(1):1–49, 2016. doi: 10.1007/s10506-016-9178-1.

[2] Sergio J. Alvarado, Michael G. Dyer, and Margot Flowers. Argument comprehension and retrieval for editorial text. *Knowl.-Based Syst.*, 3(3):139–162, 1990. doi: 10.1016/0950-7051(91)90027-Y.

[3] Leila Amgoud and Henri Prade. Using arguments for making and explaining decisions. *Artificial Intelligence*, 173(3-4):413–436, 2009. doi: 10.1016/j.artint.2008.11.006.

[4] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168(1-2):162–210, 2005. doi: 10.1016/j.artint.2005.05.006.

[5] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *Knowledge Eng. Review*, 26(4):365–410, 2011. doi: 10.1017/S0269888911000166.

[6] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007. doi: 10.1016/j.artint.2007.05.001.

[7] Trevor J. M. Bench-Capon, Henry Prakken, and Giovanni Sartor. Argumentation in legal reasoning. In Simari and Rahwan [38], pages 363–382. ISBN 978-0-387-98196-3. doi: 10.1007/978-0-387-98197-0_18.

[8] Jamal Bentahar, Bernard Moulin, and Micheline Bélanger. A taxonomy of argumentation models used for knowledge representation. *Artificial Intelligence. Rev.*, 33(3):211–259, 2010. doi: 10.1007/s10462-010-9154-1.

[9] Philippe Besnard and Anthony Hunter. *Elements of Argumentation.* MIT Press, 2008. ISBN 978-0-262-02643-7.

[10] Manuel Bichler, Michael Morak, and Stefan Woltran. lpopt: A rule optimization tool for answer set programming. In Manuel V. Hermenegildo and Pedro López-García,

editors, *Logic-Based Program Synthesis and Transformation - 26th International Symposium, LOPSTR 2016, Edinburgh, UK, September 6-8, 2016, Revised Selected Papers*, volume 10184 of *Lecture Notes in Computer Science*, pages 114–130. Springer, 2016. doi: 10.1007/978-3-319-63139-4_7.

[11] Armin Biere, Florian Lonsing, and Martina Seidl. Blocked clause elimination for QBF. In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *Automated Deduction - CADE-23 - 23rd International Conference on Automated Deduction, Wroclaw, Poland, July 31 - August 5, 2011. Proceedings*, volume 6803 of *Lecture Notes in Computer Science*, pages 101–115. Springer, 2011. doi: 10.1007/978-3-642-22438-6_10.

[12] Lawrence Birnbaum. Argument molecules: A functional representation of argument structure. In David L. Waltz, editor, *Proceedings of the National Conference on Artificial Intelligence. Pittsburgh, PA, August 18-20, 1982.*, pages 63–65. AAAI Press, 1982. ISBN 0-262-51051-0.

[13] Lawrence Birnbaum, Margot Flowers, and Rod McGuire. Towards an AI model of argumentation. In Robert Balzer, editor, *Proceedings of the 1st Annual National Conference on Artificial Intelligence. Stanford University, August 18-21, 1980.*, pages 313–315. AAAI Press/MIT Press, 1980. ISBN 0-262-51050-2.

[14] Guido Boella, Dov M. Gabbay, Leendert W. N. van der Torre, and Serena Villata. Support in abstract argumentation. In Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Guillermo Ricardo Simari, editors, *Computational Models of Argument: Proceedings of COMMA 2010, Desenzano del Garda, Italy, September 8-10, 2010.*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 111–122. IOS Press, 2010. doi: 10.3233/978-1-60750-619-5-111.

[15] Richard Booth. Judgment aggregation in abstract dialectical frameworks. In Thomas Eiter, Hannes Strass, Miroslaw Truszczynski, and Stefan Woltran, editors, *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation - Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, volume 9060 of *Lecture Notes in Computer Science*, pages 296–308. Springer, 2015. doi: 10.1007/978-3-319-14726-0_20.

[16] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*, pages 102–111. AAAI Press, 2010.

[17] Gerhard Brewka, Hannes Strass, Stefan Ellmauthaler, Johannes Peter Wallner, and Stefan Woltran. Abstract dialectical frameworks revisited. In Francesca Rossi, editor, *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013*, pages 803–809. IJCAI/AAAI, 2013.

[18] 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. doi: 10.1109/MIS.2013.122.

[19] Gerhard Brewka, Martin Diller, Georg Heissenberger, Thomas Linsbichler, and Stefan Woltran. Solving advanced argumentation problems with answer-set programming. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 1077–1083. AAAI Press, 2017.

[20] Martin W. A. Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009. doi: 10.1007/s11225-009-9218-x.

[21] Günther Charwat, Wolfgang Dvorák, Sarah Alice Gaggl, Johannes Peter Wallner, and Stefan Woltran. Methods for solving reasoning problems in abstract argumentation - A survey. *Artif. Intell.*, 220:28–63, 2015. doi: 10.1016/j.artint.2014.11.008.

[22] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Symmetric argumentation frameworks. In Lluis Godo, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 8th European Conference, ECSQARU 2005, Barcelona, Spain, July 6-8, 2005, Proceedings*, volume 3571 of *Lecture Notes in Computer Science*, pages 317–328. Springer, 2005. doi: 10.1007/11518655_28.

[23] Martin Diller, Johannes Peter Wallner, and Stefan Woltran. Reasoning in abstract dialectical frameworks using quantified boolean formulas. *Argument & Computation*, 6(2):149–177, 2015. doi: 10.1080/19462166.2015.1036922.

[24] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[25] Paul E. Dunne, Wolfgang Dvorák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. *Artificial Intelligence*, 228:153–178, 2015. doi: 10.1016/j.artint.2015.07.006.

[26] Stefan Ellmauthaler and Hannes Strass. The DIAMOND system for computing with abstract dialectical frameworks. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Computational Models of Argument - Proceedings of COMMA 2014, Atholl Palace Hotel, Scottish Highlands, UK, September 9-12, 2014*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 233–240. IOS Press, 2014. doi: 10.3233/978-1-61499-436-7-233.

[27] Stefan Ellmauthaler and Johannes Peter Wallner. Evaluating abstract dialectical frameworks with ASP. In Bart Verheij, Stefan Szeider, and Stefan Woltran, editors, *Computational Models of Argument - Proceedings of COMMA 2012, Vienna, Austria, September 10-12, 2012*, volume 245 of *Frontiers in Artificial Intelligence and Applications*, pages 505–506. IOS Press, 2012. doi: 10.3233/978-1-61499-111-3-505.

[28] Sarah Alice Gaggl, Sebastian Rudolph, and Hannes Strass. On the computational complexity of naive-based semantics for abstract dialectical frameworks. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2985–2991. AAAI Press, 2015.

[29] Thomas Linsbichler, Jörg Pührer, and Hannes Strass. Characterizing realizability in abstract argumentation. *CoRR*, abs/1603.09545, 2016. URL `http://arxiv.org/abs/1603.09545`.

[30] Florian Lonsing and Armin Biere. Depqbf: A dependency-aware QBF solver. *JSAT*, 7(2-3):71–76, 2010.

[31] Martin Odersky, Lex Spoon, and Bill Venners. *Programming in Scala: Updated for Scala 2.12*. Artima Incorporation, USA, 3rd edition, 2016. ISBN 0981531687, 9780981531687.

[32] Sylwia Polberg. Understanding the abstract dialectical framework. In Loizos Michael and Antonis C. Kakas, editors, *Logics in Artificial Intelligence - 15th European Conference, JELIA2016, Larnaca, Cyprus, November 9-11, 2016, Proceedings*, volume 10021 of *Lecture Notes in Computer Science*, pages 430–446, 2016. doi: 10.1007/978-3-319-48758-8_28.

[33] Sylwia Polberg. *Developing the Abstract Dialectical Framework*. PhD thesis, Technische Universität Wien, Institut für Informationssysteme, 2017.

[34] Sylwia Polberg and Nir Oren. Revisiting support in abstract argumentation systems. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Computational Models of Argument - Proceedings of COMMA 2014, Atholl Palace Hotel, Scottish Highlands, UK, September 9-12, 2014*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 369–376. IOS Press, 2014. doi: 10.3233/978-1-61499-436-7-369.

[35] Jörg Pührer. Realizability of three-valued semantics for abstract dialectical frameworks. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 3171–3177. AAAI Press, 2015.

[36] Jörg Pührer. Argueapply: A mobile app for argumentation. In Marcello Balduccini and Tomi Janhunen, editors, *Logic Programming and Nonmonotonic Reasoning - 14th International Conference, LPNMR 2017, Espoo, Finland, July 3-6, 2017, Proceedings*, volume 10377 of *Lecture Notes in Computer Science*, pages 250–262. Springer, 2017. doi: 10.1007/978-3-319-61660-5_23.

[37] Iyad Rahwan and Kate Larson. Argumentation and game theory. *Argumentation in Artificial Intelligence*, pages 321–339, 2009.

[38] Guillermo Ricardo Simari and Iyad Rahwan, editors. *Argumentation in Artificial Intelligence.* Springer, 2009. ISBN 978-0-387-98196-3. doi: 10.1007/978-0-387-98197-0.

[39] Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence*, 205:39–70, 2013. doi: 10.1016/j.artint.2013.09.004.

[40] Hannes Strass. The relative expressiveness of abstract argumentation and logic programming. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1625–1631. AAAI Press, 2015.

[41] Hannes Strass. Expressiveness of two-valued semantics for abstract dialectical frameworks. *Artificial Intelligence Res. (JAIR)*, 54:193–231, 2015. doi: 10.1613/jair.4879.

[42] Hannes Strass and Johannes Peter Wallner. Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*, pages 101–110. AAAI Press, 2014.

[43] Hannes Strass and Johannes Peter Wallner. Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. *Artificial Intelligence*, 226:34–74, 2015. doi: 10.1016/j.artint.2015.05.003.

[44] Stephen E. Toulmin. *The Uses of Argument, Updated Edition.* Cambridge University Press, 2008. ISBN 9780521534833.