FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Advances in Abstract Argumentation

Expressiveness and Dynamics

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Thomas Linsbichler
Registration Number 0726008

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Stefan Woltran
Co-advisor: Dipl.-Ing. Dipl.-Ing. Dr.techn. Wolfgang Dvořák

The dissertation has been reviewed by:

Prof. Gerhard Brewka

Dr. habil. Matthias Thimm

Vienna, 26th April, 2017

Thomas Linsbichler

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Advances in Abstract Argumentation

Expressiveness and Dynamics

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Thomas Linsbichler
Matrikelnummer 0726008

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.techn. Stefan Woltran
Co-Betreuung: Dipl.-Ing. Dipl.-Ing. Dr.techn. Wolfgang Dvořák

Diese Dissertation haben begutachtet:

Prof. Gerhard Brewka

Dr. habil. Matthias Thimm

Wien, 26. April 2017

Thomas Linsbichler

Erklärung zur Verfassung der Arbeit

Thomas Linsbichler
Schopenhauerstraße 52/2, 1180 Wien

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, 26. April 2017

Thomas Linsbichler

Acknowledgements

While much of this thesis is about conflicts and their resolution, writing this thesis was dominated by a harmonious atmosphere and nice people I had the pleasure to meet, work, and spend time with. I want to thank all of them for making the work on this thesis enjoyable by fruitful collaborations, lively discussions, sharing their knowledge and wisdom, but also just enjoying proper coffee or going on non-scientific excursions.

In particular I want to express my gratitude to Stefan Woltran for being such a great advisor, giving me the possibility to work with the right balance of freedom and supervision, but also for taking me to football grounds I would not have been to otherwise. Moreover, I thank Wolfgang Dvořák for his invaluable support and Johannes Wallner for his help and advice, in particular at the beginning of my PhD studies. Also I want to thank Jan Maly for proofreading large parts of the thesis as well as Adrian Haret, Christof Spanring, and Martin Diller for our joint efforts on some of the results presented in this work.

I want to name a few more people I am grateful to: co-authors of mine I had the pleasure to work with, my colleagues from the Institute of Information Systems for creating such a friendly working environment, and the guys from the Intelligent Systems group at the University of Leipzig for making me feel very welcome every time I visited them. The following is a randomly sorted incomplete list of these people: Juliane Auerböck, Toni Pisjak, Sylwia Polberg, Andreas Pfandler, Nysret Musliu, Bernhard Bliem, Hannes Strass, Beatrix Buhl, Marco Maratea, Matthias Schlögel, Michael Abseher, Frank Loebe, Paul Dunne, Gerhard Brewka, Zeynep Sarıbatur, Gerald Berger, Sarah Gaggl, Theresa Csar, Tobias Kaminski, Jean-Guy Mailly, Eva Nedoma, Shqiponja Ahmetaj, Jörg Pührer, Stefan Rümmele, Günther Charwat, Wolfgang Fischl, Massimiliano Giacomin, Reinhard Pichler, Beatrix Forsthuber, Harald Beck, Stefan Ellmauthaler, and Ringo Baumann.

Finally but most importantly I want to thank my family. My parents Andrea and Ferdinand and my brother Alexander for their love and support not only during my studies but my entire life, and my wife Katrin for her love and affection.

Most finally I want to thank everybody whom I forgot to thank explicitly for their understanding.

This research has been supported by the Austrian Science Fund (FWF) through projects I1102, P25521, and I2854.

Abstract

In recent years the research field of argumentation has become a major topic in the study of artificial intelligence (AI). This is not only due to recent applications such as legal reasoning and medicine, but also because of fundamental connections to other areas of AI research such as nonmonotonic reasoning. AI and automated reasoning can be helpful to various tasks within the argumentation process, but the focus of this work is on the evaluation of the acceptability of conflicting arguments. The most prominent approach to this problem is the formal model of abstract argumentation frameworks (AFs) introduced by Dung. An AF is a directed graph where nodes represent arguments and directed edges represent conflicts between arguments. Conditions for the acceptability of arguments are given by argumentation semantics. Several semantics have been defined over the years. The central question, given an AF, is which sets of arguments (so-called extensions) can be jointly accepted under a certain semantics.

While Dung's argumentation frameworks enjoyed and still enjoy great popularity, their conceptual simplicity also imposes certain limitations, which has led to a considerable number of generalizations of Dung's AFs. In particular, abstract dialectical frameworks (ADFs) constitute a very powerful generalization of AFs by additionally assigning to each argument an acceptance condition in the form of a propositional formula.

In this work we contribute to the advancement of the study of abstract argumentation by addressing aspects of expressiveness and dynamics of argumentation semantics in AFs as well as in ADFs. In terms of expressiveness we first complement recent work on realizability in AFs. Moreover, we investigate the role of arguments that do not appear in any extension, so-called rejected arguments, and study the induced class of compact argumentation frameworks. We give full pictures of the relations between the compact AF classes and between the expressiveness of the various semantics when restricted to compact AFs. Then, we lift the study of expressiveness to the concept of input-output AFs and give, for the major semantics, exact characterizations of functions which are realizable in this setting. Finally, we present a unifying algorithmic approach to realizability capturing AFs and ADFs as well as intermediate formalisms in a modular way, which is also implemented in answer set programming. These results not only contribute to the systematic comparison of semantics, but can also provide the theoretical basis for the advancement of solving techniques for problems in argumentation.

Taking into account the dynamic nature of argumentation, we study two central issues therein: revision and splitting. For revision we apply the seminal AGM theory of belief change to argumentation. We are the first to present a representation theorem for revision operators which guarantee to result in a single framework. For AFs we give a generic solution which applies to many prominent semantics. For ADFs we study revision under preferred and admissible semantics as well as a novel hybrid approach. We also present concrete belief change operators and analyze their computational complexity. Finally, we study splitting of ADFs, aiming for optimization of computation by incremental computation of semantics. We provide suitable techniques for directional splitting under all standard semantics of ADFs as well as for general splitting under selected semantics.

Kurzfassung

Die Argumentationstheorie hat sich in den vergangenen Jahren als eines der zentralen Themen auf dem Gebiet der Künstlichen Intelligenz (KI) etabliert. Dies ist nicht nur auf jüngste Anwendungen im Rechtswesen oder in der Medizin, sondern auch auf grundlegende Überschneidungen mit anderen Gebieten der KI-Forschung, wie zum Beispiel dem Nichtmonotonen Schließen, zurückzuführen. Während durch KI und automatisiertes Schließen verschiedene Aspekte des Argumentationsprozesses unterstützt werden können, liegt das Hauptaugenmerk dieser Arbeit auf der Evaluierung in Konflikt stehender Argumente. Die von Dung eingeführten *abstract argumentation frameworks* (AFs) stellen hierfür das am weitesten verbreitete formale Modell dar. Ein AF ist ein gerichteter Graph, dessen Knoten Argumente repräsentieren und dessen gerichtete Kanten Konflikte zwischen Argumenten darstellen. Argumentationssemantiken, derer viele in den letzten Jahren eingeführt wurden, beschreiben die Bedingungen für die Akzeptanz von Argumenten. Für ein gegebenes AF lautet die zentrale Frage, welche Mengen von Argumenten (sogenannte *extensions*) in Bezug auf eine Semantik gemeinsam akzeptiert werden können.

Auch wenn sich Dungs AFs anhaltender Popularität erfreuen, führt ihr konzeptioneller Minimalismus zu gewissen Einschränkungen, was zu zahlreichen Versuchen, diese zu erweitern, geführt hat. Eine sehr mächtige Generalisierung von AFs stellen *abstract dialectical frameworks* (ADFs) dar, in welchen jedes Argument um eine Akzeptanzbedingung in Form einer aussagenlogischen Formel erweitert wird.

In dieser Arbeit leisten wir einen Beitrag zur Weiterentwicklung der Abstrakten Argumentationstheorie, indem wir uns mit Aspekten der Ausdrucksstärke und Dynamik von Argumentationssemantiken, sowohl für AFs als auch für ADFs, beschäftigen. Im Bereich der Ausdrucksstärke ergänzen wir zuerst jüngste Resultate zu *realizability* in AFs. Weiters untersuchen wir die Rolle von Argumenten, die in keiner *extension* vorkommen, sogenannter *rejected arguments*, und studieren die sich dadurch ergebenden Klassen von *compact argumentation frameworks*. Wir präsentieren einen vollständigen Überblick der Beziehungen zwischen den *compact-AF*-Klassen sowie zwischen der Ausdrucksstärke verschiedener Semantiken in diesen Klassen. Danach wenden wir die Untersuchung der Ausdrucksstärke auf die sogenannten *input-output AFs* an und präsentieren exakte Charakterisierungen der darin realisierbaren Funktionen. Schlussendlich beschreiben wir einen algorithmischen Ansatz für das *realizability*-Problem, der auf modulare Art und Weise AFs, ADFs, sowie dazwischenliegende Formalismen abdeckt. Eine Implemen-

tierung in *answer set programming* ist verfügbar. All diese Resultate tragen nicht nur zur systematischen Analyse der verschiedenen Semantiken bei, sondern können auch als theoretische Basis für die Weiterentwicklung von Softwaresystemen zur Lösung von Argumentationsproblemen dienen.

Unter Berücksichtigung des dynamischen Charakters von Argumentation beschäftigen wir uns mit zwei dafür maßgeblichen Aspekten: *revision* und *splitting*. Im Bereich von *revision* wenden wir die als Standard etablierte AGM-Theorie für *belief change* auf die Argumentationstheorie an. In dieser Arbeit zeigen wir erstmals ein Repräsentationstheorem für *revision operators*, welche nur ein AF bzw. ADF als Resultat liefern. Für AFs präsentieren wir eine generische Lösung, die auf viele Standardsemantiken angewandt werden kann. Für ADFs untersuchen wir *revision* für die *preferred* und *admissible* Semantik sowie einen hybriden Ansatz. Weiters präsentieren wir konkrete *revision operators* und untersuchen deren Komplexität. Schließlich untersuchen wir *splitting* für ADFs, mit dem Ziel eine Optimierung durch inkrementelle Berechnung der Semantiken zu erreichen. Wir erzielen positive Resultate für *directional splitting* für alle Standardsemantiken, sowie für *general splitting* für ausgewählte Semantiken.

Contents

CHAPTER 1

Introduction

1.1 Setting the Stage

The study of argumentation and its role in human reasoning lies in the intersection of disciplines such as philosophy, logic, and legal reasoning. It is "concerned with how assertions are proposed, discussed, and resolved in the context of issues upon which several diverging opinions may be held" [33]. In particular, argumentation has become an important subfield of artificial intelligence (AI) [177, 39]. This is not only due to the intrinsic interest of this topic and recent applications, but also because of fundamental connections to other areas of AI research, in particular knowledge representation [35], multi-agent systems [152], and nonmonotonic reasoning [94]. Computational models of argumentation aim to provide methods for automated reasoning within the argumentation process. Among other things that can be concerned with the identification of arguments and their interrelations, the distinction of legitimate from invalid arguments, the resolution of conflicts between arguments, and, based on these tasks, also with drawing conclusions from the arguments. Applications lie in the fields of legal reasoning [34], medicine [137], decision support [6], and case-based reasoning [79].

As an illustrative example of argumentation and its connection to artificial intelligence we consider an imaginary discussion about the legitimacy of awarding the Nobel Prize in literature to the musician Bob Dylan (which has indeed been the case in 2016). The arguments brought forward during this discussion are listed in Table 1.1. Someone could come up with argument a, supporting the decision of the Nobel Committee. As a counterargument, the statement b, challenging the superiority of Dylan's songwriting could be brought forward. Statements c and d are uttered in order to defend a by attacking the argument b. Seemingly unrelated to this stream of arguments, someone might come up with the traditionalist argument e, conflicting with argument a. AI can support the analysis and resolution of this argumentative dialogue in various steps. First, assuming the dialogue is given in natural language, argument mining techniques can be

Table 1.1: Statements from an imaginary discussion about Bob Dylan and the Nobel Prize.

a	Bob Dylan is the greatest songwriter of the 20th century, therefore he deserves the Nobel Prize.
b	The Beatles had better songs.
c	The Beatles consisted of four people.
d	The Beatles didn't write anything close to "It's Alright, Ma (I'm Only Bleeding)".
e	The Nobel price for literature should be reserved for writers, therefore Dylan should not get the Nobel price.

employed to extract the arguments from the text (see e.g. [59]). Then, the identification of conflicts between arguments, e.g. in the form of rebuts and undercuts [172] has to make use of the structure of arguments [173]. Based on these conflicts, coherent positions, i.e. sets of jointly acceptable arguments can be identified in an automated way. From these, conclusions on the original issue of the argumentation dialogue can be drawn.

The argumentation process can therefore be structured into the following steps [63]:

1. Identify and construct arguments from a knowledge base.
2. Determine the conflicts among the arguments.
3. Evaluate the acceptability of the different arguments.
4. Conclude or identify the justified conclusions.

For the steps (1) and (2) various approaches can be found in the literature. They are concerned with instantiating argumentation systems from possibly inconsistent knowledge bases. Among them we mention deductive argumentation [38], assumption-based argumentation (ABA+) [47, 79], ASPIC+ [155], and extended defeasible logic programming (E-DeLP) [72]. In order to (3) evaluate the acceptability of arguments, all of these approaches then abstract away from the inner structure of arguments. The evaluation of these abstract argumentation systems is where the focus of this thesis lies in.

The by far most prominent approach for the formalization of abstract arguments and their relations are *(abstract) argumentation frameworks*, AFs for short, as introduced by Dung [94]. An AF is composed of a set of abstract *arguments* and a directed conflict relation among these arguments, containing the *attacks*. The instantiation of the discussion in Table 1.1 could contain the arguments a, b, c, d, and e (in this case coinciding with the statements, which is not necessarily the case) and attacks from b to

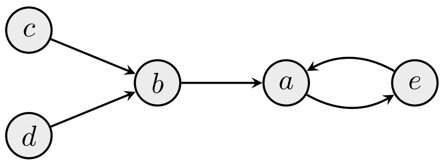


Figure 1.1: The AF modeling the discussion from Table 1.1.

a, c to b, d to b, as well as a mutual attack between a and e. We use the fact that AFs are conceptually directed graphs for their depiction. Figure 1.1 shows the AF described by representing arguments by nodes and attacks by directed edges.

The evaluation of AFs is concerned with conflict resolution, i.e. finding (maximal) sets of jointly acceptable arguments, so-called *extensions*. Different conditions that make a set of arguments acceptable are captured by different argumentation *semantics*. The most basic concept underlying nearly all semantics is conflict-freeness: a set of arguments is *conflict-free* (an extension under conflict-free semantics) if there is no attack between any two arguments of the set. Extensions under the *naive* semantics are then these conflict-free sets which are maximal with respect to set inclusion. Considering the AF depicted in Figure 1.1, the naive extensions are given by $\{a, c, d\}$, $\{b, e\}$, and $\{c, d, e\}$. Another prominent semantics is the *stable* semantics, which, besides conflict-freeness, requires a set of arguments to attack all arguments not contained in the set. Again considering the AF from Figure 1.1, we see that the stable extensions indeed differ from the naive extensions. The stable extensions are $\{a, c, d\}$ and $\{c, d, e\}$, both having no attack among members and attacking all "outside" arguments, i.e. $\{b, e\}$ and $\{a, b\}$, respectively. The naive extension $\{b, e\}$, however, is not stable since it fails to attack c and d.

Following the original semantics introduced by Dung [94], i.e. conflict-free, admissible, complete, grounded, stable, and preferred semantics, a wealth of alternative semantics have been proposed in the literature [196, 65, 95, 16, 64, 61, 13, 106]. Subsequently, the choice of semantics was often based on the desired treatment of particular examples. In order to get more insights on the distinctive features of the different semantics available, a *systematic comparison of semantics* under various aspects has been initiated and is still in progress. Most notably, the work by Baroni and Giacomin [11] provided a comprehensive account of argumentation semantics by introducing certain principles and studying their fulfillment by the different semantics. Moreover, the computational complexity of reasoning tasks under semantics was studied [97, 105], showing remarkable differences between semantics. Other aspects in the analysis of semantics involve the characterization of semantics by means of equations [37] or the fulfillment of rationality postulates [63]. Finally, the work on *realizability* [146, 99] analyzes the capabilities of semantics to express certain outcomes in terms of sets of extensions. This will be the basis for the first part of this work.

Due to the fact that AFs increasingly become the centerpiece of advanced higher-level argumentation systems, there is a growing interest in efficient solving techniques for reasoning tasks within AFs. This is also witnessed by 18 software systems participating in the first solver competition[1] [192] and the second edition being conducted at the time of writing [124]. As the reasoning problems involved are, in general, intractable, advanced algorithms are required to cope with real-world-sized data within reasonable performance. A recent survey on solving methods for AFs [71] distinguishes between reduction approaches and direct approaches. Reduction approaches translate problem instances into instances of problems of other formalisms in order to exploit existing sophisticated software. Target formalisms of existing solvers are among Boolean satisfiability (SAT) [36, 104, 69], constraint satisfaction (CSP) [42, 43], and answer set programming (ASP) [113, 122]. Direct approaches (see [162] for an example), on the other hand, implement genuine algorithms for the respective reasoning tasks. While the results of the first competition saw mainly reduction approaches in the top positions, a recent in-depth analysis [70] suggests that also existing direct approaches are most efficient in certain domains. It is one of the objectives of this work to provide theoretical basics for the advancement of such direct solving techniques.

Despite currently available systems work on static instances of abstract argumentation frameworks, argumentation is an inherently *dynamic* process. Therefore recent years have seen an increasing number of studies on different problems in the dynamics of argumentation frameworks. These works deal with enforcement [23, 20, 24, 21, 41, 78, 198], i.e. the question of whether and how to make sure that certain sets of arguments become extensions, variants thereof [142, 161], different forms of specific syntactic change [40, 68], AGM-style revision [76, 77, 25, 83], logical frameworks to express change [51, 93], and other topics [45, 164, 178, 179]. All these works consider scenarios where the argumentation framework under consideration is undergoing some change. This can be of syntactic and semantic form (intervention and observation according to Rienstra [178]). In the former case new arguments or attacks are added, e.g. since another point is being made in a discussion or the instantiated defeasible theory is being extended. Here the main challenge is of computational nature. Semantic information about the AF before the change might be beneficial for the computation of semantics of the AF involving the changes. The latter form of change is concerned with revision of an argumentation framework due to new information describing (a part of) the desired outcome of the framework. Here one of the main challenges is the incorporation of this new knowledge in the argumentation framework, which usually underlies certain constraints such as minimality of change.

A reason why Dung's argumentation frameworks enjoyed and still enjoy great popularity is their conceptual simplicity. This, however, also imposes certain limitations to their abilities to model argumentation scenarios. This is because the relations between arguments might be more divers than individual attack. For instance, an argument might not be strong enough to defeat another argument, but need a second argument to jointly defeat it. Getting back to the discussion in Table 1.1 one could take the stance that neither of

[1]International Competition on Computational Models of Argumentation (ICCMA) `http://argumentationcompetition.org/`

the statements c and d are strong enough to defend a from b, but both together do the job – a situation that can not be directly modeled in AFs. Or, one might want to model support explicitly as a relation between arguments instead of packing it into the (hidden) structure of arguments. For instance, the argument a from Table 1.1 might be regarded as two statements a_1, “Bob Dylan is the greatest songwriter of the 20th century”, and a_2, “he deserves the Nobel Prize”, with a_1 supporting a_2. Although this approach of modeling the structure of arguments as support is often approached with scepticism (see e.g. [174]), certain scenarios might call for such a treatment. Therefore, since the publication of [94], a considerable number of generalizations of AFs have been proposed. These include formalisms that extend AFs by the ability to model preferences [5, 153, 7], weights [98, 75], collective attacks [158], or recursive attacks [15]. Moreover, AFs have been enriched by various forms of support [46, 163, 171], for instance in bipolar argumentation frameworks [66, 67]. An overview of AF generalizations has been given in [56]. Moreover, a comprehensive account of intertranslatability between formalisms among the plethora of generalizations of Dung’s AFs has been presented by Polberg [169, 170].

Among the most powerful generalizations of AFs are *abstract dialectical frameworks* (ADFs), first introduced by Brewka and Woltran [53] and further refined in [55]. These ADFs offer any type of links between arguments: individual attack (as in AFs), collective attacks (as in SETAFs [158]), and individual and collective support, to name just a few. This is achieved by associating each argument with a so-called acceptance condition. The actual relationship between arguments is then specified by these acceptance conditions, usually in the form of propositional formulas. For instance, the acceptance condition of an argument receiving several individual attacks would be the conjunction of negated atoms, one for each argument. Or, if two arguments c and d jointly defeat argument b, then acceptance condition of b would be $\neg(c \wedge d)$. By providing such a flexible way of modeling relations between arguments, ADFs unify several of the different approaches mentioned before. In particular, they generalize AFs in a principled and systematic way. The semantics of ADFs [185] have shown to be proper generalizations of AF semantics, and are, in general, three-valued, similar to labelling-based semantics of AFs [64]. Within their brief lifespan ADFs, the “lovechild of AFs and logic programs” [188], have been employed in several different contexts. These include the instantiation of defeasible theories [186], applications in legal reasoning [1, 2, 3] as well as text exploration [60], and connections to other area such as judgement aggregation [48] and causal calculus [44]. The wide range of modeling capacities of ADFs naturally come with the price of increased complexity [190, 123], with reasoning tasks for ADFs lying one level higher in the polynomial hierarchy than the corresponding tasks for AFs. This makes efficient solving even more challenging, but first attempts have already been presented. They are based on ASP, like DIAMOND [116] and YADF [57], or, as QADF [88, 90], on quantified Boolean formulas (QBF).

1.2 Contributions

The purpose of this work is to study aspects of expressiveness and dynamics of argumentation formalisms. The main objectives are to get a better understanding of the particularities of semantics and to provide theoretical foundations for advanced solving methods. Taking into account the dynamic nature of argumentation, we study two central issues therein: revision and splitting. For revision we study how one can incorporate new information into argumentation framework while following the principle of minimal change. Splitting is concerned with whether the semantics of argumentation frameworks can be computed incrementally, in the positive case allowing for optimization of computation when syntactic change is involved.

We present every main topic of this thesis, i.e. *expressiveness*, *revision*, and *splitting*, for both AFs and ADFs. The choice of these formalisms is due to the simplicity and prominence of AFs on the one hand, and the modeling power of ADFs on the other. We are interested in how the extended capabilities of ADFs compared to AFs affect the central properties of semantics we study in this work.

Expressiveness. The first major topic is the investigation of the expressiveness of argumentation formalisms. We do so by studying the *signatures*, and several variants thereof, of argumentation formalisms under certain semantics. The signature of a semantics σ in a formalism $\mathcal{F}$ is the collection of all results that can be achieved by evaluating a knowledge base of $\mathcal{F}$ under σ:

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

Let us, for the moment consider Dung's AFs as the formalism in question. Then the aim is to find simple criteria to decide whether a given set of extensions is contained in the signature of a particular semantics. In other words, one is interested in the following problem: Given a semantics σ together with a collection of sets of arguments, $\mathbb{S}$, what characteristics determine if there is any AF whose σ-extensions (i.e. the result of evaluating the AF under σ) are exactly the sets forming $\mathbb{S}$? These characteristics are reflected in the results on signatures: thus the above question can be answered yes, if and only if, $\mathbb{S} \in \Sigma^{\sigma}_{\mathrm{AF}}$.

Studying signatures in AFs has been initiated in [146]. Strass [188] and Pührer [175] analyzed the expressiveness of ADF under two-valued semantics and three-valued semantics, respectively. In this work we continue this line of work by lifting it to a much more general setting. We combine the mentioned works into a *unifying* framework, and at the same time extend them to formalisms and semantics not considered in the respective papers: we treat several formalisms, namely AFs, SETAFs, bipolar ADFs, and ADFs, while the previous works all used different approaches and techniques. This is possible because all of these formalisms can be seen as subclasses of ADFs that are obtained by suitably restricting the acceptance conditions. We will present a general algorithm for deciding realizability for the mentioned formalisms under standard semantics: that is the decision problem, for formalism $\mathcal{F}$ and semantics σ, if a given set of interpretations is contained in the signature of $\mathcal{F}$ under σ.

Our results on realizability thus can be used in the following two ways, given a set of interpretations V:

(i) if $V \in \Sigma^{\sigma}_{\mathcal{F}}$, then there is at least one instance of $\mathcal{F}$ which as V as results of the evaluation under σ;

(ii) if $V \notin \Sigma^{\sigma}_{\mathcal{F}}$, then there is no instance of $\mathcal{F}$ whose σ-results are exactly V. Thus, for every $\mathsf{kb} \in \mathcal{F}$ either there is some $v \in V$ for which $v \notin \sigma(\mathsf{kb})$ or there is some $v \in \sigma(\mathsf{kb})$ for which $v \notin V$.

First, these results are important for constructing AFs. Indeed, knowing whether a set of interpretations V is contained in $\Sigma^{\sigma}_{\mathcal{F}}$ is a necessary condition which should be checked before actually looking for an $\mathcal{F}$-framework kb which realizes V under σ, i.e. $\sigma(\mathsf{kb}) = V$.

This is of high importance when dynamic aspects of argumentation are considered, details of which will be become clearer in a few pages. As an example, suppose an ADF D possesses as its σ-interpretations a set V and one asks for an adaptation of the framework D such that its σ-interpretation are given by $V \cup \{v\}$, i.e. one interpretation is to be added. The addition of v to V may, for instance, be desired by some agent on the grounds that v contains a valuation of arguments which it wishes to be believed by other agents. Before considering the adapted framework's structure, it is obviously crucial to know whether an appropriate framework exists at all, i.e. whether $V \cup \{v\} \in \Sigma^{\sigma}_{\text{ADF}}$.

Second, these results add to the systematic comparison of semantics, even between formalisms, by means of different properties. So far such properties have been limited to AFs and largely focused on aspects of single extensions $S \in \mathbb{S}$ rather than on sets of such. Furthermore, our results add to the growing body of work considering generic treatments of argumentation semantics, that is with respect to shared properties rather than from the perspective of distinguishing features. For instance, we show that most semantics σ, both in AFs and ADFs are closed under intersection of extensions and interpretations, respectively (more formally, for all F_1, $F_2 \in \mathcal{F}$, there exists an $F \in \mathcal{F}$ such that $\sigma(F) = \sigma(F_1) \cap \sigma(F_2)$, whenever $\sigma(F_1) \cap \sigma(F_2) \neq \emptyset$).

Besides the uniform approach for deciding realizability in AFs, SETAFs and (bipolar) ADFs, we conduct a more fine grained analysis of expressiveness of AFs. In particular, we study the role of rejected arguments, i.e. arguments in a given AF that do not appear in any extension. In order to have a handle for analyzing the effect of rejected arguments, we introduce the class of compact argumentation frameworks: an AF is compact (with respect to a semantics σ), if each of its arguments appears in at least one σ-extension. Consider again the AF in Figure 1.1, call it F, and recall the naive extensions of F: $nai(F) = \{\{a, c, d\}, \{c, d, e\}, \{b, e\}\}$. Every argument of F occurring in at least one naive extension means that no argument is rejected with respect to naive semantics and therefore F is compact for *nai*. On the other hand, recall that

$stb(F) = \{\{a, c, d\}, \{c, d, e\}\}$. Hence b is rejected with respect to stable semantics and F is not compact for *stb*.

Although rejected arguments are natural ingredients in typical argumentation scenarios, it is of interest to understand in which ways rejected arguments contribute to the "strength" of a particular semantics. In terms of expressiveness, the natural question is whether any AF F can be transformed to an equivalent AF G, i.e. $\sigma(F) = \sigma(G)$ for a given semantics σ, that is compact. For naive semantics, it is rather easy to see that any AF can be transformed into an equivalent compact AF by just removing all self-attacking arguments. In other words, the same outcome (in terms of the naive extensions) can be achieved by a simplified AF without rejected arguments. For other semantics, that are considered more mature, we will see that this transformation is not possible in general. In this case we can conclude that the full range of expressiveness of σ indeed relies on the concept of rejected arguments.

By introducing the class of compact AFs, for each semantics σ, we contribute to a stream of research identifying certain fragments of AFs that show favorable behaviour. For instance, syntactic subclasses of AFs such as acyclic, symmetric, odd-cycle-free or bipartite AFs have been considered, as for some of these classes different semantics collapse [73, 96]. Moreover, there are also several classes defined via properties of extensions. Most prominent among those subclasses is the class of coherent AFs [97], i.e. AFs where stable and preferred extensions coincide. Further examples of semantic subclasses can be found in [12, 104].

A promising application of the results on compact AFs lies in the field of concrete software systems for computing semantics of AFs. Preprocessing steps that remove rejected arguments might be beneficial to the runtime of computing the extensions (which afterwards should however be interpreted in terms of the original AF), as it leads to a guaranteed reduction of search space. Moreover, if an AF is known to have no rejected arguments then all of its arguments are contained in at least one extension and so credulous as well as skeptical reasoning become easy tasks.

As another variant of expressiveness in AFs we study the functional completeness of AF semantics from an input-output viewpoint. Baroni et al. [17] have shown that an AF can be viewed as a set of partial interacting sub-frameworks each characterized by its input-output behavior, i.e. a semantics-dependent function which maps each labelling of the "input" arguments (the external arguments affecting the sub-framework) into the set of labellings prescribed for the "output" arguments (the arguments of the sub-framework affecting the external ones). As it turns out, sub-frameworks with the same input-output behavior can be safely exchanged under complete, grounded, stable and (under some mild conditions) preferred semantics. That means that replacing a sub-framework with an equivalent one does not affect the justification status of the invariant arguments: semantics of this kind are called transparent in [17].

As a simple example, consider an argumentation framework including a chain of 4 arguments $a_1, \ldots, a_4$ where for $i \in \{2, 3, 4\}$, a_{i-1} attacks a_i and a_i does not receive other attacks, and a_1 is unattacked. This chain can be seen as a sub-framework with input argument a_1 and output argument a_4, which under any transparent semantics can be replaced by any even-length chain without affecting the justification status of the arguments outside the sub-framework. A natural question concerns the expressive power of transparent semantics in the context of an interacting sub-framework: given a so-called I/O-specification, i.e. a function describing an input-output behaviour by mapping extensions (resp. interpretations) to sets of extensions (resp. interpretations), is there an AF with designated input and output arguments realizing this function under a given semantics?

While expressiveness from an input-output perspective further contributes to the comparison of semantics, it also motivated by other aspects. First, a functional characterization provides a common ground for different representations of the same sub-framework, as in metalevel argumentation [154] where meta-level arguments making claims about object-level arguments allow for equivalent characterizations of the same framework at different levels of abstraction. One may also want to translate a different formalism to an AF or vice versa, e.g. to express a logical system as an AF or provide an argument-free representation of a given AF for human/computer interaction issues. In all of these cases, it is important to know whether an input-output behavior is realizable under a given argumentation semantics. Finally, our results can be of importance in the setting of strategic argumentation [191], where a player may exploit the fact that for some set of arguments certain outcomes are achievable (or non achievable) independently of other arguments. For example, an agent may desire to achieve some goal, i.e. ensure that a certain argument is justified. Considering arguments brought up by other agents as input arguments, our results enable the agent to verify whether the goal is achievable and provide one particular way for the agent to bring up further arguments in order to succeed.

Dynamics. In the realm of dynamics we study with two topics, revision and splitting, where the first deals with the proper incorporation of change in a semantic form, and the other mainly aims at optimizing solving techniques to react to syntactic change.

The first problem we address in this work is revising an AF when new, trusted information is provided. Based on the AGM paradigm in belief change [4, 126], we mean by revision an operation which incorporates the new information following certain principles (captured by the AGM postulates) while bringing only minimal change to the original AF.

To the best of our knowledge, the issue was first addressed explicitly by Coste-Marquis et al. [76]. In their work, AF revision is defined as follows: given a semantics, an AF and a revision formula encoding desired changes in the status of some arguments, find *a set of AFs* satisfying the revision formula, whose extensions are as close as possible to the extensions of the input AF. Following the AGM approach, rationality postulates for a revision operator on AFs can be formulated and Coste-Marquis et al. [76] also provide a *representation theorem.* Such a result establishes a close link between obeying the postulates and exploiting a particular type of ranking on extensions of AFs in order to compute the output of the revision. This approach is thus similar to the model-based propositional revision by Katsuno and Mendelzon [141]. Our approach is inspired by the work of Coste-Marquis et al. [76], with the notable difference that we study AF revision operators which

- take an AF as the first input and another AF as the second input, and
- produce a *single* AF as output.

The motivation for this is twofold. First, it is in accordance with standard AGM revision in fragments, where both the original formula and the revising formula stem from the same language (see, for example, revision in the Horn fragment [84]). Second, revision yielding a single AF is not only natural and standard, but also makes concepts of iterated revision [81, 183] amenable to argumentation. Indeed, for an iterated application of belief change operations, it is desirable that the input and output are of the same type. The same point holds for persuasion, where some current state of discourse needs to be updated such that an agent is convinced to accept a certain argument: it has been emphasized that modeling persuasion can benefit from applying change operations in argumentation [136]. Thus, understanding belief change of abstract argumentation formalisms can pave the way towards a general theory of formal persuasion.

We follow the same principle when studying revision of ADFs. That is, we consider revising ADFs when new information is provided in the form of an ADF and resulting again in a single ADF. Due to the three-valued nature of ADF semantics, standard revision operators from the literature are not directly transferable to ADF revision. Therefore we define a three-valued version of Dalal's famous operator [80] and show that it is meaningful in the sense that it fulfills all postulates. Moreover, we present a novel approach of combining two semantics, admissible and preferred in this case, when revising ADFs.

Restricting the output of AF revision operators to a single knowledge base poses significant challenges, as representation theorems from the propositional belief change literature are not easily applicable in the new context. We will make use of insights obtained from studying expressiveness of AF and ADF semantics in order to obtain representation results for revision.

The other main topic within the dynamics in argumentation, splitting, is concerned with the question whether, and under which conditions, a semantics of a formalism allow for incremental computation of its results. The concept was first introduced in the context of logic programs [145] and then studied for other formalisms, such for AFs by Baumann [19]. In this work we investigate whether ADF semantics are amenable for splitting. We show that, in a restricted setting called *directional splitting*, all semantics behave rather nicely and we obtain conceptually easy splitting results. The general setting turns out to be more involved; we show valid *general splitting* techniques for two-valued models and admissible semantics.

Splitting is a fundamental principle both from a theoretical and practical point of view. From a theoretical stance, splitting gives insights on whether local evaluation of a semantics is possible, even if this is not directly apparent from its definition. On the practical side, splitting techniques can be useful for solving in two different ways. In a static setting, computing the semantics incrementally can boost the performance of the evaluation by dividing one large task into several smaller tasks. In a dynamic setting, splitting results allow us to recompute only those parts of the knowledge base which have been affected, when the knowledge base undergoes change and we are interested in the semantic results.

To summarize, the main contributions of this thesis are as follows:

- We complement the work on realizability in AFs by (i) considering two additional semantics, complete and resolution-based grounded [16]; (ii) studying certain closure properties of signatures; (iii) considering quantitative aspects of realizability; and (iv) clarifying the complexity of realizability.
- Moreover, we consider the subclass of compact AFs and give full pictures of the relations (i) between the compact AF classes under the various semantics; and (ii) between signatures of the various semantics when realizability is restricted to compact AFs.
- Then, for realizability in the input-output setting we characterize all realizable two-valued I/O-specification for the majority of considered semantics and characterize all realizable three-valued I/O-specification for preferred and grounded semantics.
- Wrapping up the results on expressiveness, we present a unifying algorithmic approach to realizability which captures AFs and ADFs as well as the intermediate formalisms SETAFs and bipolar ADFs. We do so in a modular way which allows for an extension to other semantics and formalisms.
- We have implemented this approach in answer set programming and used the implementation to obtain several novel results on the relative expressiveness of the abovementioned formalisms.
- For revision, we present a representation result for rational revision operators by rankings of interpretations for the general class of proper I-maximal semantics.
- For revision of ADFs we (i) provide a representation result for revision under preferred semantics; (ii) show that there is only one rational revision operator for admissible semantics; and (iii) present, due to unintuitive behaviour discussed for the former approaches, a hybrid approach to revision of ADFs, combining preferred and admissible semantics.
- Moreover, we address the complexity of Dalal's operator for revision of AFs, showing completeness results up to Θ_3^{P}.

- Finally, for splitting of ADFs, we show that all considered semantics are amenable for directional splitting (i.e. only allowing division between strongly connected components) in a conceptually elegant way and provide general splitting results for two-valued models and admissible interpretations.

1.3 Outline

This thesis is organized as follows.

In Section 2 we introduce the basis and basics of our work – the central formalisms of this work, AFs and ADFs, as well as the methodology with which we analyze them.

In Section 3 we tackle various aspects of expressiveness. After providing some preliminaries in Section 3.1 we will complement previous work on realizability in AFs in Section 3.2, study compact AFs and the issue of realizability therein in Section 3.3, and view AFs and realizability from an input-output perspective in Section 3.4. In Section 3.5 we review and combine recent work on the realizability in ADFs allowing us to present an unifying algorithmic approach to realizability in Section 3.6.

In Section 4 we are concerned with revision of argumentation formalisms. We first review the general approach of AGM belief change in Section 4.1 and then study revision of AFs capturing a wide range of semantics in Section 4.2. Then we deal with revision of ADFs in Section 4.3, first for preferred semantics, then for admissible semantics, and finally introduce a hybrid approach. We study the complexity of Dalal's revision operator for AFs in Section 4.4 and discuss some further issues in Section 4.5.

We study splitting in Section 5 by first reviewing previous work on splitting AFs in Section 5.1 and then presenting splitting results for ADFs in Section 5.2, first directional splitting and then general splitting.

Finally, in Section 6 we will summarize and conclude the presented results, discuss related work, and attempt an outlook on future research directions.

1.4 Publications

Most of the results presented in this work have been published. In the following we list the publications which contain contributions that can also be found in this thesis. We will point to the relevant references at the beginning of each chapter.

[31] Ringo Baumann, Wolfgang Dvořák, Thomas Linsbichler, Christof Spanring, Hannes Strass, and Stefan Woltran. On rejected arguments and implicit conflicts: The hidden power of argumentation semantics. *Artif. Intell.*, 241:244–284, 2016.

[89] Martin Diller, Adrian Haret, Thomas Linsbichler, Stefan Rümmele, and Stefan Woltran. An extension-based approach to belief revision in abstract argumentation.

In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2926–2932. AAAI Press, 2015.

[102] Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. *Artif. Intell.*, 228: 153–178, 2015.

[110] Wolfgang Dvořák, Thomas Linsbichler, Emilia Oikarinen, and Stefan Woltran. Resolution-based grounded semantics revisited. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, pages 269–280. IOS Press, 2014.

[130] Massimiliano Giacomin, Thomas Linsbichler, and Stefan Woltran. On the functional completeness of argumentation semantics. In Chitta Baral, James P. Delgrande, and Frank Wolter, editors, *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*, pages 43–52. AAAI Press, 2016.

[147] Thomas Linsbichler. Splitting Abstract Dialectical Frameworks. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, pages 357–368. IOS Press, 2014.

[148] Thomas Linsbichler and Stefan Woltran. Revision of abstract dialectical frameworks: Preliminary report. In Sarah Gaggl, Juan Carlos Nieves, and Hannes Strass, editors, *Proceedings of the 1st international Workshop on Argumentation in Logic Programming and Non-Monotonic Reasoning (Arg-LPNMR 2016)*, 2016.

[150] Thomas Linsbichler, Jörg Pührer, and Hannes Strass. A uniform account of realizability in abstract argumentation. In Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen, editors, *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*, pages 252–260. IOS Press, 2016.

[31] is a long version of [30] and [149]. [102] is an extended version of [101]. For [89], a long version in the form of a technical report is available [91].

The author of this thesis has co-authored further articles, which, while being related to this work, do not contain results directly included here. They are listed below.

[32] Ringo Baumann, Thomas Linsbichler, and Stefan Woltran. Verifiability of argumentation semantics. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Proceedings of the 6th International Conference on Computational Models of Argument (COMMA 2016)*, pages 83–94. IOS Press, 2016.

[57] 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 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*, pages 1077–1083. AAAI Press, 2017.

[58] Remi Brochenin, Thomas Linsbichler, Marco Maratea, Johannes P. Wallner, and Stefan Woltran. Abstract solvers for Dung's argumentation frameworks. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Theory and Applications of Formal Argumentation – 3rd International Workshop (TAFA 2015), Revised Selected Papers*, pages 40–58. Springer, 2015.

[103] Paul E. Dunne, Christof Spanring, Thomas Linsbichler, and Stefan Woltran. Investigating the relationship between argumentation semantics via signatures. In Subbarao Kambhampati, editor, *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pages 1051–1057. IJCAI/AAAI Press, 2016.

[111] Wolfgang Dvořák, Sarah A. Gaggl, Thomas Linsbichler, and Johannes P. Wallner. Reduction-based approaches to implement Modgil's Extended Argumentation 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*, pages 249–264. Springer, 2015.

[124] Sarah A. Gaggl, Thomas Linsbichler, Marco Maratea, and Stefan Woltran. Introducing the second international competition on computational models of argumentation. In Matthias Thimm, Federico Cerutti, Hannes Strass, and Mauro Vallati, editors, *Proceedings of the 1st International Workshop on Systems and Algorithms for Formal Argumentation (SAFA 2016)*, pages 4–9. CEUR-WS.org, 2016.

CHAPTER 2

Background

This chapter introduces the central concepts and notation the remainder of this work is built upon. We will first, in Section 2.1 give some preliminaries on propositional logic and order theory and define the kind of objects we will be working with. Then we will introduce the necessary basics for abstract argumentation frameworks and abstract dialectical frameworks in Sections 2.2 and 2.3, respectively. Finally, we will briefly review the necessary background on computational complexity (of AFs) in Section 2.4.

2.1 Logic, Orders, Interpretations

We assume basic knowledge of the syntax and semantics of propositional logic. For a comprehensive introduction we refer to [118].

For propositional formulas we make use of the standard connectives such as negation ($\neg$), logical and ($\wedge$), logical or ($\vee$), implication ($\rightarrow$), and equivalence ($\leftrightarrow$) and evaluate formulas with respect to standard semantics of propositional logic. Given a formula ψ and a set of atoms S, we write $S \models \psi$ if S is a model of ψ, i.e. if ψ evaluates to true when atoms in S are considered true and all other atoms are considered false. Moreover, we denote the set of *models* of ψ by $\mathsf{Mod}(\psi)$. Given formulas ψ and μ we write $\psi \models \mu$ if $\mathsf{Mod}(\psi) \subseteq \mathsf{Mod}(\mu)$ and $\psi \equiv \mu$ if $\mathsf{Mod}(\psi) = \mathsf{Mod}(\mu)$. For disjoint sets of atoms $S_1, \ldots, S_n$ we write $\psi(S_1, \ldots, S_n)$ if the atoms occurring in ψ coincide with $S_1 \cup \cdots \cup S_n$. The set of all propositional formulas over atoms $\mathfrak{A}$ is given by $\mathsf{P}_{\mathfrak{A}}$.

A formula is in *conjunctive normal form* (CNF) if it is of the form $\bigwedge_{c \in C} \bigvee_{x \in c} x$ with $C \subseteq 2^{\mathfrak{A}}$ where each $c \in C$ is called a *clause* and each $x \in c$ is a literal (i.e. an atom a or its negation $\neg a$). We may abbreviate the formula in CNF as $\bigwedge_{c \in C} c$. A formula is in *disjunctive normal form* (DNF) if it is of the form $\bigvee_{d \in D} \bigwedge_{x \in d} x$, where each $d \in C$ is called a *monom* and each $x \in d$ is a literal. We may abbreviate the formula in DNF as

$\bigvee_{d \in D} d$. Finally, we denote the syntactic transformation of a formula ψ into an equivalent formula in CNF (resp. DNF) as *cnf*(ψ) (resp. *dnf*(ψ)).

Moreover, we make use of standard mathematical concepts like functions, preorders, partially ordered sets, and lattices. Given a function $f : X \mapsto Y$ we denote the *update of f with a pair* $(x, y) \in X \times Y$ by $f|^x_y : X \mapsto Y$ with $f|^x_y(x) = y$, and $f|^x_y(z) = f(z)$ if $z \neq x$. For a function $f : X \mapsto Y$ and $y \in Y$, its preimage is $f^{-1}(y) = \{x \in X \mid f(x) = y\}$.

Let S be a set. A *preorder* (on S) is a reflexive, transitive binary relation $\preceq \subseteq (S \times S)$. A *(non-strict) partial order* (on S) is a preorder on S that is antisymmetric. A preorder on S is *total* if $a \preceq b$ or $b \preceq a$ for every $a, b \in S$. Given a preorder $\preceq$ on S, we write $a \prec b$ for $a, b \in S$ if $a \preceq b$ but $b \not\preceq a$ and $a \approx b$ for $a, b \in S$ if $a \preceq b$ and $b \preceq a$. A *strict partial order* (on S) is a irreflexive, transitive, antisymmetric relation $\prec \subseteq (S \times S)$. The partial order $\preceq$ associated to the strict partial order $\prec$ is given by $a \preceq b$ if and only if $a \prec b$ or $a = b$ for $a, b \in S$.

A *partially ordered set* is a pair $(S, \preceq)$ with $\preceq$ a partial order on S. A partially ordered set $(S, \preceq)$ is

- a *complete lattice* if and only if every $S' \subseteq S$ has both a greatest lower bound (glb) $\bigsqcap S' \in S$ and a least upper bound (lub) $\bigsqcup S' \in S$;
- a *complete meet-semilattice* if and only if every non-empty subset $S' \subseteq S$ has a greatest lower bound $\bigsqcap S' \in S$ (the *meet*) and every ascending chain $C \subseteq S$ has a least upper bound $\bigsqcup C \in S$.

Throughout this work, we assume a countably infinite domain $\mathfrak{A}$ of arguments. Unless stated differently, $A \subseteq \mathfrak{A}$ is a finite set of arguments. In Sections 2.2 and 2.3 we will introduce formalisms to model relations between arguments A with the ultimate goal to obtain a *acceptance status* of the arguments. Interpretations describe such a status by assigning one of the three truth values *true* ($\mathbf{t}$), *false* ($\mathbf{f}$) or *undecided*[1] ($\mathbf{u}$) to each statement.

Definition 1. An *interpretation* (over A) is a mapping $v : A \mapsto \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. The set of all interpretations (over A) is denoted by $\mathcal{V}$. The set of all interpretations over $S \subset A$ is denoted by $\mathcal{V}(S)$. An *interpretation-set* is a set of interpretations over the same finite set of arguments.

When listing sets of interpretations in examples, for the sake of readability we represent three-valued interpretations by sequences of truth values, tacitly assuming that the underlying vocabulary is given and has an associated total ordering. For example, for $A = \{a, b, c\}$ we represent the interpretation $\{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{u}\}$ by the sequence $\mathbf{tfu}$.

In general, interpretations are three-valued. If the truth value $\mathbf{u}$ is not assigned we call them two-valued.

[1] Sometimes the truth value $\mathbf{u}$ is also referred to as *unknown*.

Definition 2. An interpretation v is *two-valued* if $v(a) \in \{\mathbf{t}, \mathbf{f}\}$ for each $a \in A$. The set of all two-valued interpretations (over A) is denoted by $\mathcal{V}_2$.

We introduce some further notation for interpretations.

Definition 3. Given an interpretation v over A and a truth value $\mathbf{x} \in \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$, we refer to the set of arguments $a \in A$ such that $v(a) = \mathbf{x}$ as $v^{\mathbf{x}} = \{a \in A \mid v(a) = \mathbf{x}\}$. The interpretation assigning the same value $\mathbf{x} \in \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ to each argument is denoted by $v_{\mathbf{x}} : A \mapsto \{\mathbf{x}\}$. The *negation* of an interpretation v, denoted by $\neg v$, is defined such that $(\neg v)^{\mathbf{t}} = v^{\mathbf{f}}$, $(\neg v)^{\mathbf{f}} = v^{\mathbf{t}}$, and $(\neg v)^{\mathbf{u}} = v^{\mathbf{u}}$.

Definition 4. Given an interpretation v over $S \subseteq A$ and a set $S' \subset S$, we define $v|_{S'}$ as the interpretation over S' such that $v|_{S'}(a) = v(a)$ for each $a \in S'$. For sets $S_1, S_2 \subseteq A$ with $S_1 \cap S_2 = \emptyset$ and interpretations v_1 over S_1 and v_2 over S_2 we denote by $v_1 \cup v_2$ the interpretation over $S_1 \cup S_2$ such that $(v_1 \cup v_2)(a) = v_1(a)$ if $a \in S_1$ and $(v_1 \cup v_2)(a) = v_2(a)$ if $a \in S_2$.

Two-valued interpretations v can be used to evaluate propositional formulas φ following the standard semantics of propositional logic. Therefore they assign truth values $v(\varphi) \mapsto \{\mathbf{t}, \mathbf{f}\}$ to propositional formulas according to standard evaluation.

The three truth values are partially ordered according to their information content.

Definition 5. The truth values are strictly ordered by $<_i$ such that $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$ and no other pair in $<_i$. The *information ordering* $\leq_i$ is the partial order associated to $<_i$. The information ordering extends to interpretations $v_1, v_2 \in \mathcal{V}$ over $S \subseteq A$ in a way that

$$v_1 \leq_i v_2 \text{ iff } v_1(a) \leq_i v_2(a) \text{ for all } a \in S.$$

We say for two interpretations v_1, v_2 that v_2 *extends* v_1 iff $v_1 \leq_i v_2$. The set of all two-valued interpretations that extend a given interpretation v is denoted by $[v]_2$, i.e.

$$[v]_2 = \{v' \in \mathcal{V}_2 \mid v \leq_i v'\}.$$

Intuitively, the information ordering means that the Boolean truth values, $\mathbf{t}$ and $\mathbf{f}$, contain more information than the truth value undecided.

The pair $(\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}, \leq_i)$ forms a complete meet-semilattice with the information meet operation $\sqcap_i$. This meet can intuitively be interpreted as *consensus* and assigns $\mathbf{t} \sqcap_i \mathbf{t} = \mathbf{t}$, $\mathbf{f} \sqcap_i \mathbf{f} = \mathbf{f}$, and returns $\mathbf{u}$ otherwise.

The set $\mathcal{V}$ of all interpretations over A forms a complete meet-semilattice $(\mathcal{V}, \leq_i)$ with respect to the information ordering $\leq_i$. The consensus meet operation $\sqcap_i$ of this semilattice is given by

$$v_1 \sqcap_i v_2 = v \text{ such that } v(a) = v_1(a) \sqcap_i v_2(a) \text{ for all } a \in A.$$

The least element of $(\mathcal{V}, \leq_i)$ is the interpretation $v_\mathbf{u}$ mapping all statements to undecided – the least informative interpretation. The $\leq_i$-maximal elements of the meet-semilattice $(\mathcal{V}, \leq_i)$ are the two-valued interpretations $\mathcal{V}_2$. For each interpretation v, the elements of $[v]_2$ form an $\leq_i$-antichain with greatest lower bound $v = \bigsqcap_i [v]_2$.

Definition 6. Given a set of interpretations V over $S \subseteq A$, we define $\max_{\leq_i} V = \{v \in V \mid \nexists v' \in V : v <_i v'\}$. Moreover, for any truth value $\mathbf{x} \in \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$, we define $V^\mathbf{x} = \{v^\mathbf{x} \mid v \in V\}$. Finally the restriction of V to $S' \subseteq S$ is $V|_{S'} = \{v|_{S'} \mid v \in V\}$.

Example 1. Consider the set of interpretations $V = \{v_1, v_2, v_3\}$ over $A = \{a, b, c\}$ with $v_1 = \mathbf{tuu}$, $v_2 = \mathbf{fuu}$, and $v_3 = \mathbf{ftu}$. Then we have $\max_{\leq_i} V = \{v_1, v_3\}$. Moreover, $V^\mathbf{t} = \{\emptyset, \{a\}, \{b\}\}$, $V^\mathbf{f} = \{\emptyset, \{a\}\}$, and $V^\mathbf{u} = \{\{c\}, \{b, c\}\}$. ♢

We use the following notions to compare interpretations.

Definition 7. Given interpretations $v_1, v_2 \in \mathcal{V}$, we say that v_1 and v_2 are

- *comparable* if $v_1 \leq_i v_2$ or $v_2 \leq_i v_1$;
- *incomparable* if they are not comparable;
- *compatible* if $v_1^\mathbf{t} \cap v_2^\mathbf{f} = v_1^\mathbf{f} \cap v_2^\mathbf{t} = \emptyset$;
- *incompatible* if they are not compatible.

A set of interpretations V is called (pairwise) incomparable (resp. incompatible) if all $v_1, v_2 \in V$ with $v_1 \neq v_2$ are incomparable (resp. incompatible).

Note that if a pair of interpretations is comparable then it is also compatible, while the reverse does not hold. Conversely, a pair (resp. a set) of interpretations being incompatible implies that it is incomparable, but not the other way round.

Example 2. Consider the interpretations $v_1 = \mathbf{tu}$ and $v_2 = \mathbf{uf}$ over $A = \{a, b\}$. It holds that v_1 and v_2 are compatible since $v_1^\mathbf{t} \cap v_2^\mathbf{f} = v_1^\mathbf{f} \cap v_2^\mathbf{t} = \emptyset$, but they are incomparable since neither $v_1 \leq_i v_2$ nor $v_2 \leq_i v_1$. Moreover, no set $V \supseteq \{v_1, v_2\}$ containing the interpretations v_1 and v_2 is incompatible. ♢

Lemma 1. *Any set of two-valued interpretations $V \subseteq \mathcal{V}_2$ is incompatible.*

Proof. Let $V \subseteq V_2$ and $v_1, v_2 \in V$ with $v_1 \neq v_2$. Since both, v_1 and v_2, are two-valued, there must be some $a \in A$ with $v_1(a) = \mathbf{t}$ and $v_2(a) = \mathbf{f}$, or $v_1(a) = \mathbf{f}$ and $v_2(a) = \mathbf{t}$. Hence v_1 and v_2 are incompatible. As they were chosen arbitrarily we conclude that V is incompatible. □

On the two classical truth values $\mathbf{t}$ and $\mathbf{f}$, we define the *truth ordering* $\mathbf{f} <_t \mathbf{t}$. Consequently, the operations $\sqcup_t$ and $\sqcap_t$ are defined as $\mathbf{f} \sqcup_t \mathbf{t} = \mathbf{t}$ and $\mathbf{f} \sqcap_t \mathbf{t} = \mathbf{f}$. These operations extend to two-valued interpretations as usual, i.e. $(v_1 \sqcup_t v_2)(a) = v_1(a) \sqcup_t v_2(a)$ and $(v_1 \sqcap_t v_2)(a) = v_1(a) \sqcap_t v_2(a)$. Again, the reflexive version of $<_t$ is denoted by $\leq_t$. The pair $(\mathcal{V}_2, \leq_t)$ of two-valued interpretations ordered by the truth ordering forms a complete lattice with glb $\sqcap_t$ and lub $\sqcup_t$. This complete lattice has the least element $v_{\mathbf{f}}$, the interpretation mapping all arguments to false, and the greatest element $v_{\mathbf{t}}$, mapping all arguments to true, respectively.

When dealing with interpretations which are two-valued, i.e. assigning only the values $\mathbf{t}$ and $\mathbf{f}$, we will identify an interpretation v just by the arguments which are assigned $\mathbf{t}$, i.e. by the set $S = v^{\mathbf{t}}$. In particular, we do so when presenting work on abstract argumentation frameworks (cf. Section 2.2) under extension-based semantics.

For collections of sets of arguments, we define notions for the set of all arguments occurring in the collections and the set of all pairs of arguments occurring together in some set. Moreover, we will call such collections extension-sets in the finite case.

Definition 8. Given $\mathbb{S} \subseteq 2^{\mathfrak{A}}$, we use

- $Args_{\mathbb{S}}$ to denote $\bigcup_{S \in \mathbb{S}} S$, and
- $Pairs_{\mathbb{S}}$ to denote $\{(a, b) \mid \exists S \in \mathbb{S} : \{a, b\} \subseteq S\}$.

$\mathbb{S}$ is called an *extension-set* (over $\mathfrak{A}$) if $Args_{\mathbb{S}}$ is finite.

While $|\mathbb{S}|$ denotes the number of extensions in $\mathbb{S}$, $\|\mathbb{S}\|$ stands for $|Args_{\mathbb{S}}|$. The restriction of $\mathbb{S}$ to some $T \subseteq Args_{\mathbb{S}}$ is given by $\{S \cap T \mid S \in \mathbb{S}\}$.

The remaining definitions recall properties of extension-sets presented in [146].

Definition 9. Let $\mathbb{S} \subseteq 2^{\mathfrak{A}}$. The *downward-closure*, $dcl(\mathbb{S})$, of $\mathbb{S}$ is given by $\{S' \subseteq S \mid S \in \mathbb{S}\}$. We call $\mathbb{S}$

- *downward-closed* if $\mathbb{S} = dcl(\mathbb{S})$, and
- *incomparable* if all elements $S \in \mathbb{S}$ are pairwise incomparable, i.e. for each $S, S' \in \mathbb{S}$, $S \subseteq S'$ implies $S = S'$.

Note that the notion of incomparability is defined both for sets of interpretations and for sets of sets of arguments. However, there is no one-to-one correspondence, as sets of arguments do not necessarily have a distinct corresponding interpretation.

Definition 10. An extension-set $\mathbb{S} \subseteq 2^{\mathfrak{A}}$ is *tight* if for all $S \in \mathbb{S}$ and $a \in Args_{\mathbb{S}}$ it holds that if $S \cup \{a\} \notin \mathbb{S}$ then there exists an $s \in S$ such that $(a, s) \notin Pairs_{\mathbb{S}}$.

Definition 11. A set $\mathbb{S} \subseteq 2^{\mathfrak{A}}$ is called *conflict-sensitive*[2] if for each $A, B \in \mathbb{S}$ such that $A \cup B \notin \mathbb{S}$ it holds that $\exists a, b \in A \cup B : (a, b) \notin \mathit{Pairs}_{\mathbb{S}}$.

2.2 Abstract Argumentation Frameworks

In this section we recall the basic definitions of *abstract argumentation frameworks* (AFs) [94] as well as subsequent work on argumentation *semantics* such as [196, 65, 95, 16, 64]. For an excellent overview on argumentation semantics we refer to [14].

Definition 12. An *(abstract) argumentation framework* (AF) is a pair $F = (A, R)$ where $A \subseteq \mathfrak{A}$ is a finite set of arguments and $R \subseteq A \times A$ is the *attack relation*. The collection of all AFs over $\mathfrak{A}$ is given by $AF_{\mathfrak{A}}$.

Let $F = (B, Q)$ be an AF. We use A_F to refer to B and R_F to refer to Q. Moreover, we write $a \rightarrow_F b$, and say that a attacks b, if $(a, b) \in R_F$; for $S \subseteq A_F$, we write $S \rightarrow_F a$ (resp. $a \rightarrow_F S$), and say that S attacks a (resp. a attacks S), if there exists some $s \in S$ such that $s \rightarrow_F a$ (resp. $a \rightarrow_F s$). Moreover, we will occasionally denote symmetric attacks $(a, b), (b, a) \in R_F$ as $\langle a, b\rangle \in R_F$.

Definition 13. Given an AF $F \in AF_{\mathfrak{A}}$ and a set of arguments $S \subseteq A_F$, the *range* of S (in F) is defined as $S^+_F = S \cup \{a \mid S \rightarrow_F a\}$.

Definition 14. Given an AF $F \in AF_{\mathfrak{A}}$, an argument $a \in A_F$ is *defended* (in F) by a set $S \subseteq A_F$ (or, S defends a) if for each $b \in A_F$ such that $b \rightarrow_F a$, also $S \rightarrow_F b$. A set of arguments $T \subseteq A_F$ is defended (in F) by S if each $a \in T$ is defended by S (in F).

We will make use of the following observation.

Lemma 2 ([102])**.** *Given an AF $F = (A, R)$ and two sets of arguments $S, T \subseteq A$. If S defends itself in F and T defends itself in F, then $S \cup T$ defends itself in F.*

Extension-based semantics of AFs, as defined by Dung [94], map each AF to a set of two-valued interpretations. We call these interpretations *extensions* and denote them just by sets of arguments, containing the arguments which are accepted. That is, a semantics σ is a function $\sigma : AF_{\mathfrak{A}} \mapsto 2^{2^{\mathfrak{A}}}$, where the elements $E \in \sigma(F)$ are the extensions of F under σ (or σ-extensions for short).

In the following we introduce the semantics we study in this work. These are *conflict-free*, *admissible*,[3] *naive*, *stable*, *complete*, *grounded*, *preferred*, *semi-stable* [65], *stage* [196], *ideal* [95], and *resolution-based grounded* [16] semantics, which we will abbreviate by *cf*,

[2] An equivalent property was called *adm-closed* in [146, 101]. In the interest of adequacy to its content we stick to the term *conflict-sensitive*.

[3] In the literature, conflict-freeness and admissibility are often regarded as properties rather than semantics. We will use the properties, but at the same time treat them as semantics with $\mathit{cf}(F)$ and $\mathit{adm}(F)$ denoting the conflict-free and admissible extensions, respectively.

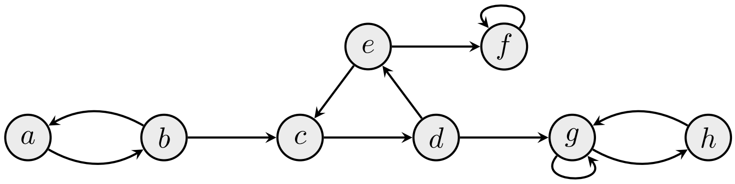


Figure 2.1: An argumentation framework.

adm, *nai*, *stb*, *com*, *grd*, *prf*, *sem*, *stg*, *idl*, and *grd**, respectively. For a given semantics σ and an AF F, we denote the set of σ-extensions of F by $\sigma(F)$.

Definition 15. Given an AF $F \in AF_{\mathfrak{A}}$, a set of arguments $S \subseteq A_F$ is *conflict-free* (in F), $S \in cf(F)$, if there are no arguments $a, b \in S$ such that $(a,b) \in R_F$. A conflict-free set $S \in cf(F)$ is *admissible* (in F), $S \in adm(F)$, if S defends itself.

- $S \in nai(F)$, if $S \in cf(F)$ and there is no $T \in cf(F)$ with $T \supset S$;
- $S \in stb(F)$, if $S \in cf(F)$ and $S \rightarrow_F a$ for all $a \in A_F \setminus S$;
- $S \in com(F)$, if $S \in adm(F)$ and $a \in S$ for all $a \in A$ defended by S;
- $S \in grd(F)$, if $S = \bigcap com(F)$;
- $S \in prf(F)$, if $S \in adm(F)$ and there is no $T \in adm(F)$ with $T \supset S$,
- $S \in sem(F)$, if $S \in adm(F)$ and there is no $T \in adm(F)$ with $T_F^+ \supset S_F^+$;
- $S \in stg(F)$, if $S \in cf(F)$ and there is no $T \in cf(F)$ with $T_F^+ \supset S_F^+$;
- $S \in idl(F)$, if $S \in adm(F)$, $S \subseteq \bigcap prf(F)$ and there is no $T \in adm(F)$ with $T \supset S$ and $T \subseteq \bigcap prf(F)$.

Note that grounded and ideal semantics are *unique status semantics* since every AF has exactly one extension under these semantics. All other semantics considered in this work are *multiple status semantics*. AFs may have no extensions under stable semantics, while all other semantics always yield at least one extension.

Example 3. To illustrate the semantics, consider the following AF:

$$\begin{aligned}F = (&\{a,b,c,d,e,f,g,h\},\\ &\{(a,b),(b,a),(b,c),(c,d),(d,e),(d,g),(e,c),(e,f),(f,f),(g,g),(g,h),(h,g)\}).\end{aligned}$$

F is depicted in Figure 2.1, where nodes represent arguments and directed edges represent attacks. First, the conflict-free sets of F are as follows:

$$\begin{aligned}cf(F) = \{&\emptyset,\{a\},\{b\},\{c\},\{d\},\{e\},\{h\},\{a,c\},\{a,d\},\{a,e\},\{a,h\},\{b,d\},\{b,e\},\\ &\{b,h\},\{c,h\},\{d,h\},\{e,h\},\{a,c,h\},\{a,d,h\},\{a,e,h\},\{b,d,h\},\{b,e,h\}\}.\end{aligned}$$

Note that no set containing f or g can be conflict-free, since both f and g are self-attacking. Among the conflict-free sets, the following sets are admissible:

$$adm(F) = \{\emptyset, \{a\}, \{b\}, \{h\}, \{a,h\}, \{b,d\}, \{b,h\}, \{b,d,h\}\}.$$

The empty set is always conflict-free and admissible. The conflict-free set $\{a,d\}$, for instance, is not admissible since d is attacked by c in F, but $\{a,d\}$ does not attack c, i.e. it does not defend d. The naive extensions are just the $\subset$-maximal conflict-free sets:

$$nai(F) = \{\{a,c,h\}, \{a,d,h\}, \{a,e,h\}, \{b,d,h\}, \{b,e,h\}\}.$$

For stable semantics, it can be checked, that there is no conflict-free set of arguments in F attacking all other arguments, hence

$$stb(F) = \emptyset.$$

The complete extensions of F are those admissible sets, which do not defend any argument not contained in the set:

$$com(F) = \{\emptyset, \{a\}, \{h\}, \{a,h\}, \{b,d,h\}\}.$$

For instance, the admissible set $\{b,d\}$ is not complete since it defends h. As no argument of a is unattacked, the grounded extension is empty:

$$grd(F) = \{\emptyset\}.$$

The preferred extensions are just the $\subseteq$-maximal admissible sets, which always coincide with the $\subseteq$-maximal complete extensions:

$$prf(F) = \{\{a,h\}, \{b,d,h\}\}.$$

The semi-stable and stage extensions of F are given as follows:

$$\begin{aligned} sem(F) =& \{\{b,d,h\}\} \\ stg(F) =& \{\{a,e,h\}, \{b,e,h\}, \{b,d,h\}\} \end{aligned}$$

Note that the set of semi-stable extensions are always a subset of the set of preferred extensions, and likewise for stage and naive. $\{b,d\}$ is then the only semi-stable extension of F by $\{a,h\}^+_F = \{a,b,g,h\} \subset \{a,b,c,d,e,g,h\} = \{b,d,h\}^+_F$. Likewise, for instance, $\{a,c,h\} \notin stg(F)$ since $\{a,c,h\}^+_F = \{a,b,c,d,g,h\} \subset \{a,b,c,d,e,g,h\} = \{b,d,h\}^+_F$. Finally, $\{h\} = \bigcap prf(F)$ and $\{h\}$ is admissible, hence

$$idl(F) = \{\{h\}\}.$$

$\diamondsuit$

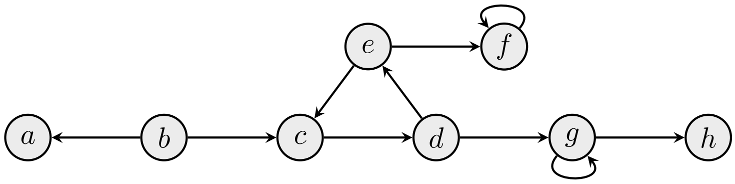


Figure 2.2: A resolution of the AF in Figure 2.1.

The family of *resolution-based* semantics introduced by Baroni et al. [16] is a parametric approach defined based on the notion of a resolution.[4] A resolution results from selecting exactly one direction for every symmetric attack and removing the attack going in the other direction.

Definition 16. Given an AF $F \in AF_{\mathfrak{A}}$, a *resolution* of F is an AF F' with $R_{F'} \subseteq R_F$, such that each $(a, a) \in R_F$ is also contained in $R_{F'}$ and for each $(a, b) \in R_F$ with $a \neq b$ either $(a, b) \in R_{F'}$ or $(b, a) \in R_{F'}$, but not both. We denote the set of all resolutions of F as $\gamma(F)$.

The extensions of an AF F under the resolution-based version of a semantics σ are now given by the subset-minimal σ-extensions among all σ-extensions of any resolution of F.

Definition 17. Given a semantics σ, the *resolution-based* σ semantics is given by σ^* such that, for any AF $F \in AF_{\mathfrak{A}}$,

- $S \in \sigma^*(F)$, if $S \in \bigcup_{F' \in \gamma(F)} \sigma(F')$ and there is no $T \in \bigcup_{F' \in \gamma(F)} \sigma(F')$ with $T \subset S$.

We will consider the grounded instance of this family of semantics grd^*, which is also the most prominent one.

Example 4. Consider again the AF F depicted in Figure 2.1. For computing $grd^*(F)$, we need to check all resolutions of F. Since there are two symmetric attacks in F, it has four resolutions. One resolution $F' \in \gamma(F)$ is depicted in Figure 2.2. It holds that $grd(F') = \{b, d, h\}$. For the the other resolutions we get the grounded extensions $\{a\}$, $\{a, h\}$, and $\{b, d, h\}$, respectively. We obtain

$$grd^*(F) = \{\{a\}, \{b, d, h\}\},$$

as the resolution-based grounded semantics selects the minimal elements among the grounded extensions of resolutions. We recall at this place that the resolution-based grounded semantics obviously is *multiple-status*. $\diamond$

[4] We use a slightly different, but equivalent, presentation compared to Baroni et al. [16].

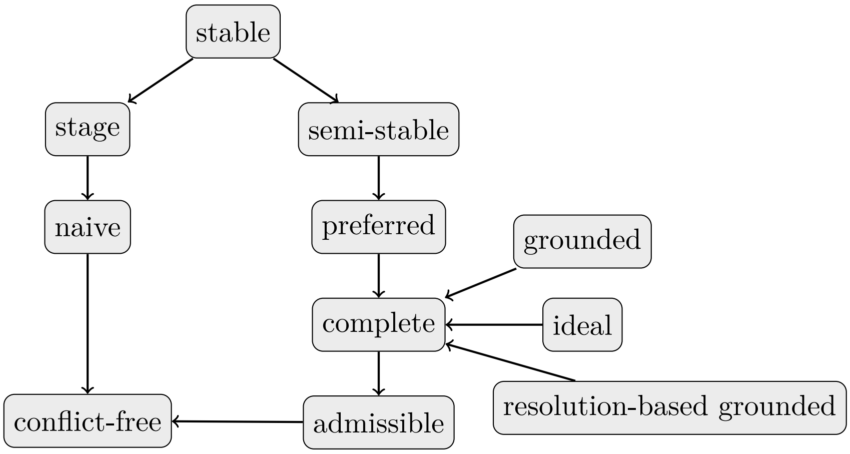


Figure 2.3: Relations between semantics of AFs.

Further notable semantics not considered in this work include the *cf2* [13, 121] and *stage2* [106] (both being instantiations of the SCC-recursive schema for argumentation semantics [13]), as well as the *eager* semantics [61] (a parametric version of ideal semantics [100]), *prudent* semantics [74] and *strongly admissible* sets [11, 62].

An alternative definition of most of the semantics from Definition 15 can be given via the characteristic function [94] of argumentation frameworks.

Definition 18. Given an AF $F \in AF_{\mathfrak{A}}$, the *characteristic function* $\Gamma_F : 2^{A_F} \mapsto 2^{A_F}$ is defined as

$$\Gamma_F(S) = \{a \in A_F \mid a \text{ is defended by } S\}.$$

The original semantics of [94] are then certain fixed points of the characteristic function.

Proposition 1 ([94])**.** *Given an AF $F \in AF_{\mathfrak{A}}$ and a conflict-free set of arguments $S \in cf(F)$, it holds that*

- *$S \in grd(F)$ iff S is the least fixed point of Γ_F;*
- *$S \in adm(F)$ iff $S \subseteq \Gamma_F(S)$;*
- *$S \in com(F)$ iff $S = \Gamma_F(S)$;*
- *$S \in prf(F)$ iff $S = \Gamma_F(S)$ and there is no $T \in cf(F)$ with $T \supset S$ and $T = \Gamma_F(T)$.*

In particular, this means that we can compute the grounded extension of an AF F in polynomial time by an algorithm which, starting with the empty set, iteratively applies the characteristic function until a fixed point is reached.

Proposition 2. *In accordance with Figure 2.3, for any AF $F \in AF_{\mathfrak{A}}$, the following relations hold:*

- $stb(F) \subseteq stg(F) \subseteq nai(F) \subseteq cf(F)$;
- $stb(F) \subseteq sem(F) \subseteq prf(F) \subseteq com(F) \subseteq adm(F) \subseteq cf(F)$;
- $grd(F) \subseteq com(F)$;
- $idl(F) \subseteq com(F)$;
- $grd^*(F) \subseteq com(F)$.

For each semantics there is also a three-valued version of the semantics, giving a more fine-grained view of the acceptance status of arguments. This concept was introduced by the work on *labellings* by Caminada and Gabbay [64]. A labelling of an AF is an assignment of one label among *in*, *out*, and *undec* to each of the arguments. In this work, however, we identify labellings by three-valued interpretations as introduced in Section 2.1. Essentially, just the names of the labels change by that. At some points we will still refer to three-valued interpretations in the AF context as labellings.

As already mentioned there is a labelling-based counterpart to each extension-based semantics. The following one-to-one-correspondence between extensions and three-valued interpretations defines the labelling-based semantics. The set of σ-interpretations (or σ-labellings) of an AF F is denoted by $\sigma_3(F)$.

Definition 19. Given an AF $F \in AF_{\mathfrak{A}}$, let $S \subseteq A_F$. The labelling corresponding to S is called $e2l(S)$ and is defined as $(e2l(S))^{\mathbf{t}} = S$, $(e2l(S))^{\mathbf{f}} = S_F^+ \setminus S$, and $(e2l(S))^{\mathbf{u}} = A_F \setminus S_F^+$. For a semantics σ, its three-valued (or labelling-based) version σ_3 is defined such that $E \in \sigma(F)$ iff $e2l(E) \in \sigma_3(F)$.

In words, the labelling corresponding to an extension is such that an argument is labelled *in* if it is contained in the extension, *out* if it is attacked by an argument which is contained in the extension, and *undec* otherwise.

Caminada and Gabbay [64] define labelling-based semantics independently from the extension-based ones by giving conditions for arguments to be $\mathbf{t}$, $\mathbf{f}$, and $\mathbf{u}$, respectively. However, they show a correspondence to extension-based semantics, which gives rise to a characterization as in Definition 19.

Example 5. Again consider the AF from Figure 2.1. Recall from Example 3 that $prf(F) = \{\{a, h\}, \{b, d, h\}\}$. The interpretations, or labellings, corresponding to $\{a, h\}$ and $\{b, d, h\}$ are

$$
\begin{aligned}
v_1 =& \{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{u}, d \mapsto \mathbf{u}, e \mapsto \mathbf{u}, f \mapsto \mathbf{u}, g \mapsto \mathbf{f}, h \mapsto \mathbf{t}\}, \text{ and} \\
v_2 =& \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}, d \mapsto \mathbf{t}, e \mapsto \mathbf{f}, f \mapsto \mathbf{u}, g \mapsto \mathbf{f}, h \mapsto \mathbf{t}\},
\end{aligned}
$$

respectively. We write that $prf_3(F) = \{v_1, v_2\}$. $\Diamond$

We define a few syntactic operations on AFs.

Definition 20. Given AFs $F, F' \in AF_{\mathfrak{A}}$ and a set of arguments $S \subseteq A_F$, define

- the union of F and F', $F \cup F' = (A_F \cup A_{F'}, R_F \cup R_{F'})$;
- the restriction of F to S, $F|_S = (S, R_F \cap (S \times S))$;
- the subtraction of S from F, $F - S = (A_F \setminus S, R_F \cap ((A_F \setminus S) \times (A_F \setminus S)))$.

Being syntactically a directed graph, it can be of interest to identify the strongly connected components (SCCs) of an AF.

Definition 21. Given an AF $F \in AF_{\mathfrak{A}}$, let ϱ_F be the relation defined over $A_F \times A_F$ such that $\varrho_F(a, b)$ holds iff $x = y$ or there are directed paths from a to b and from b to a in F. The set of *strongly connected components* of F is given by $SCCs(F) = \{F|_{[a]} \mid a \in A\}$, where $[a]$ is the equivalence class of a in ϱ_F.

In words the SCCs of an AF F are given by the AFs corresponding to the equivalence classes of the reachability relation ϱ_F.[5]

Finally, certain syntactic subclasses of AFs will be of interest.

Definition 22. An AF $F \in AF_{\mathfrak{A}}$ is

- *symmetric* if for each $(a, b) \in R_F$ also $(b, a) \in R_F$;
- *self-attack-free* if there is no $a \in A_F$ such that $(a, a) \in R_F$;

AFs with sets of attacking arguments. A straightforward generalization of AFs was introduced by Nielsen and Parsons [158]. The basic idea of their formalism is that attacks are not performed by single arguments but by sets of arguments.

Definition 23. A SETAF is a pair $S = (A, X)$ where A is a finite set of arguments and $X \subseteq (2^A \setminus \{\emptyset\}) \times A$ is the *(set) attack* relation.

A set attack from a set of arguments B to argument a is successful only if all arguments in B are accepted. Based on this convention, we define what it means for an argument to be (un)acceptable.

Definition 24. Let S be a SETAF. Given a statement $a \in A$ and an interpretation v, we say that a is *acceptable* (in S) with respect to v if and only if $\forall (B, a) \in X \exists a' \in B : v(a') = \mathbf{f}$ and a is *unacceptable* (in S) with respect to v if and only if $\exists (B, a) \in X \forall a' \in B : v(a') = \mathbf{t}$.

[5] Note that here we deviate from standard notions in graph theory, where an SCC denotes just the vertices without the involved edges.

We define three-valued counterparts of the semantics introduced by Nielsen and Parsons [158], following the same conventions as in labelling-based semantics of AFs [64] and argumentation formalisms in general (cf. Definition 19).

Definition 25. For an interpretation $v : A \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ it holds that

- $v \in \mathit{adm}_3(S)$ iff for all $a \in A$, a is acceptable wrt. v if $v(a) = \mathbf{t}$ and a is unacceptable wrt. v if $v(a) = \mathbf{f}$;
- $v \in \mathit{com}_3(S)$ iff for all $a \in A$, a is acceptable wrt. v iff $v(a) = \mathbf{t}$ and a is unacceptable wrt. v iff $v(a) = \mathbf{f}$;
- $v \in \mathit{prf}_3(S)$ iff v is $\leq_i$-maximal admissible;
- $v \in \mathit{stb}_3(S)$ iff $v \in \mathit{adm}(F)$ and $\nexists a \in A : v(a) = \mathbf{u}$.

2.3 Abstract Dialectical Frameworks

In this section we give the basic definitions of *abstract dialectical frameworks* (ADFs) as introduced in [53, 55]. A more comprehensive account of ADF semantics and their origins in approximation fixpoint theory [87] can be found in [185].

We begin with the original notation for ADFs from [53].

Definition 26. An *abstract dialectical framework* (ADF) is a tuple $D = (A, L, C)$ where

- $A \subseteq \mathfrak{A}$ is a finite set of arguments,
- $L \subseteq A \times A$,
- $C = \{C_a\}_{a \in S}$ is a set of total functions $C_a : 2^{par_D(a)} \mapsto \{\mathbf{t}, \mathbf{f}\}$, the *acceptance condition* of a.

For an argument $a \in A$, $par_D(a) = \{b \in A \mid (b, a) \in L\}$.

We will, however, use the following alternative, more compact, notation for ADFs, which is also common in the literature (see e.g. [175]):

Definition 27. An *abstract dialectical framework* (ADF) D is a set of tuples $\{\langle a, \varphi_a \rangle\}_{a \in A}$ where A is the set of arguments and φ_a is a propositional formula over A – the acceptance condition of a. The collection of all ADFs over arguments $\mathfrak{A}$ is given by $ADF_{\mathfrak{A}}$.

It can be seen that these two notions are equivalent. First, the set of links L is implicitly given by the atoms occurring in the acceptance conditions. That is, whenever argument b occurs as an atom in the acceptance condition φ_a of argument a, we have that $(b, a) \in L$.

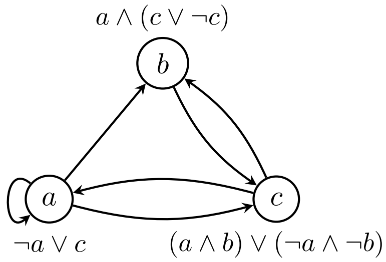


Figure 2.4: An abstract dialectical framework.

Second, a propositional formula φ_a is just a compact way of representing the Boolean function C_a.

For an ADF D, we will occasionally refer to its set of arguments by A_D, to its set of links by L_D, and to the acceptance condition of an argument $a \in A_D$ by φ_a^D. Given a set of arguments $S \subseteq A_D$ such that there is no link $(b, c) \in L_D$ with $b \in A_D \subseteq S$ and $c \in S$, $D|_S$ denotes the restriction of D to S, i.e. $D|_S = \{\langle a, \varphi_a \rangle \mid a \in S\}$.

Example 6. Consider the ADF

$$D = \{\langle a, \neg a \vee c\rangle, \langle b, a \wedge (c \vee \neg c)\rangle, \langle c, (a \wedge b) \vee (\neg a \wedge \neg b)\rangle\}.$$

The set of links which is implicitly given by the acceptance conditions φ_a, φ_b, and φ_c is $L = \{(a, a), (a, b), (a, c), (b, c), (c, a), (c, b)\}$. The annotated directed graph in Figure 2.4 depicts D. It does so by representing, similarly as for AFs, arguments by nodes and links by directed edges. The acceptance conditions are written next to the nodes corresponding to the respective arguments. We will use this graphical representation of ADFs throughout this work. $\Diamond$

While links in ADFs are abstract in the sense that their meaning is determined solely by the acceptance condition of the arguments, we distinguish certain link types.

Definition 28. Given an ADF D, a link $(b, a) \in L_D$ is

- *supporting* (in D) iff for all $v \in V$, we have $v(\varphi_a) = \mathbf{t}$ implies $v|_{\mathbf{t}}^{b}(\varphi_a) = \mathbf{t}$;
- *attacking* (in D) iff for all $v \in V$, we have $v(\varphi_a) = \mathbf{f}$ implies $v|_{\mathbf{t}}^{b}(\varphi_a) = \mathbf{f}$.

The set of supporting links in D is denoted by $sup(D)$. The set of attacking links in D is denoted by $att(D)$.

Intuitively, a link from b to a is supporting if accepting b, and leaving the acceptance status of all other arguments unchanged, never switches a from being accepted to being rejected. Likewise, link (b, a) is attacking if acceptance of b never causes a to be accepted, when it was rejected before. It can be seen there is also the possibility for a link to be neither supporting nor attacking, called a *dependent* link, or to be both supporting and attacking, called a *redundant* link. The following example illustrates the link types.

Example 7. Again consider the ADF D discussed in Example 6 and depicted in Figure 2.4. We obtain the following sets of supporting and attacking links:

- $sup(D) = \{(c, a), (a, b), (c, b)\}$;
- $att(D) = \{(a, a), (c, b)\}$.

Observe that the links (a, c) and (b, c) are dependent. Consider, for instance the link (a, c): for interpretation $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{f}\}$ we have $v_1(c) = \mathbf{t}$, but $v_1|^a_{\mathbf{t}}(c) = \mathbf{f}$, hence (a, c) is not supporting; for interpretation $v_2 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}\}$ we have $v_2(c) = \mathbf{f}$, but $v_2|^a_{\mathbf{t}}(c) = \mathbf{t}$, hence (a, c) is not attacking. Moreover, the link $(c, b) \in sup(D) \cap att(D)$ is redundant. As the name suggests, the acceptance status of c has no influence on the evaluation of φ_b. $\Diamond$

Bipolar ADFs are now defined as ADFs which contain only supporting and attacking links.

Definition 29. A *bipolar ADF* (BADF) is an ADF D such that $sup(D) \cup att(D) = L_D$. It is denoted by $(D, L_D \setminus att(D), L_D \setminus sup(D))$.

In order to utilize the computational advantages of bipolar ADFs (see [190]) the link types have to be known, since determining whether a link is supporting (resp. attacking) is intractable in general. Therefore we explicitly list the links which are not attacking and the links which are not supporting for bipolar ADFs. The remaining links are then redundant.

Example 8. Continuing Example 7 by again considering the AF D from Figure 2.4, we observe that D is not bipolar, since $(a, c), (b, c) \notin sup(D) \cup att(D)$. Let us alter the acceptance condition of c to $\varphi_c = \neg a \wedge \neg b$ and let D' denote the newly obtained *ADF*. We get $sup(D') = \{(c, a), (a, b), (c, b)\}$ and $att(D') = \{(a, a), (c, b), (a, c), (b, c)\}$ and therefore observe that D' is bipolar. The notation with explicit link types is then $(D', \{(c, a), (a, b)\}, \{(a, a), (a, c), (b, c)\})$. $\Diamond$

The semantics of ADFs are defined via the characteristic operator Γ over three-valued interpretations.

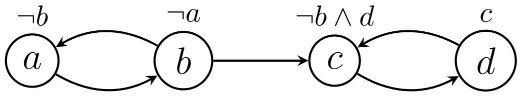


Figure 2.5: ADF D discussed in Example 9.

Definition 30. Given an ADF D and an interpretation v, the *characteristic operator* $\Gamma_D : \mathcal{V} \mapsto \mathcal{V}$ is defined as

$$\Gamma_D(v) = v' \text{ such that } v'(a) = \bigsqcap_{w \in [v]_2} w(\varphi_a).$$

Intuitively, the operator returns, for each argument a, the consensus truth value of the evaluation of the acceptance formula φ_a under each two-valued interpretation extending v. It generalizes the characteristic function for AFs (cf. Definition 18).

The semantics of ADFs can now be defined as follows:

Definition 31. Given an ADF D, an interpretation v is

- *admissible* for D iff $v \leq_i \Gamma_D(v)$,
- *complete* for D iff $v = \Gamma_D(v)$,
- *preferred* for D iff v is admissible for D and each $v' \in \mathcal{V}$ with $v <_i v'$ is not admissible for D,
- *grounded* for D iff v is complete for D and each $v' \in \mathcal{V}$ with $v' <_i v$ is not complete for D,
- a *(two-valued) model* of D iff v is two-valued and $v = \Gamma_D(v)$,
- a *stable model* of D iff v is a model of D and $v^{\mathbf{t}} = w^{\mathbf{t}}$, where w is the grounded interpretation of $D^{-v} = \{\langle a, \varphi_a[x/\bot : v(x) = \mathbf{f}]\rangle \mid a \in v^{\mathbf{t}}\}$.

We denote the admissible, complete, preferred, and grounded interpretations, and two-valued and stable models, of an ADF D by $adm_3(D)$, $com_3(D)$, $prf_3(D)$, $grd_3(D)$, $mod_3(D)$, and $stb_3(D)$, respectively. We therefore have counterparts to most of the AF semantics given in Definition 15, in their three-valued form (cf. Definition 19).

Observe that we denote the (two-valued) models and the stable models of ADFs by mod_3 and stb_3, respectively, although they are obviously two-valued. We do so in the interest of uniformity among ADF semantics, for which the candidates are, in general, three-valued interpretations.

Example 9. Consider the ADF

$$D = \{\langle a, \neg b\rangle, \langle b, \neg a\rangle, \langle c, \neg b \wedge d\rangle, \langle d, \neg c\rangle\}$$

depicted in Figure 2.5. Moreover, let

$$v_1 = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}, d \mapsto \mathbf{u}\}.$$

Concerning the application of $\Gamma_D(v_1)$, observe that we can find, for each acceptance condition φ of D, two-valued interpretations extending v_1 under which φ evaluates to $\mathbf{t}$ (resp. $\mathbf{f}$). For instance, consider $\varphi_a = \neg b$. For any $w_1 \in [v_1]_2$ such that $w_1(b) = \mathbf{f}$ we have that $w_1(\varphi_a) = \mathbf{t}$, and for any $w_1' \in [v_1]_2$ such that $w_1(b) = \mathbf{t}$ we have that $w_1(\varphi_a) = \mathbf{f}$. Hence $\bigsqcap_{w\in[v_1]_2} w(\varphi_a) = \mathbf{u}$. This means that $\Gamma_D(v_1) = v_1$ and consequently, by definition of grounded interpretations, $grd_3(D) = \{v_1\}$. The other interpretations such that the characteristic operator returns an interpretation with at least as much information, i.e. $\Gamma_D(v) \geq_i v$ are as follows:

$$\begin{aligned}
v_1 =&\{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}, d \mapsto \mathbf{u}\},\\
v_2 =&\{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{u}, d \mapsto \mathbf{u}\},\\
v_3 =&\{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{f}, d \mapsto \mathbf{f}\},\\
v_4 =&\{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{t}, d \mapsto \mathbf{t}\},\\
v_5 =&\{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{u}, d \mapsto \mathbf{u}\},\\
v_6 =&\{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}, d \mapsto \mathbf{u}\},\\
v_7 =&\{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}, d \mapsto \mathbf{f}\},\\
v_8 =&\{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}, c \mapsto \mathbf{f}, d \mapsto \mathbf{f}\}.
\end{aligned}$$

The admissible interpretations of D are therefore $adm_3(D) = \{v_1, \ldots, v_8\}$. On the other hand, let $v_9 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}, d \mapsto \mathbf{u}\}$ and observe that $\Gamma_D(v_9) = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{u}, d \mapsto \mathbf{f}\}$. For the evaluation of φ_c we have that for $w_9 = \mathbf{ftft}$ it holds that $w_9(\varphi_c) = \mathbf{t}$ and for $w_9' = \mathbf{ftff}$ it holds that $w_9(\varphi_c) = \mathbf{f}$, hence $\bigsqcap_{w\in[v_9]_2} w(\varphi_c) = \mathbf{u}$. It follows that v_9 is not an admissible interpretation of D. In order to check the complete interpretations consider v_5 and observe that for all $w \in [v_5]_2$ we have, due to $v_5(b) = \mathbf{t}$, that $w(b) = \mathbf{t}$ and, consequently, $\Gamma_D(v_5)(c) = \bigsqcap_{w\in[v_5]_2} w(\varphi_c) = \mathbf{f}$. Hence $\Gamma_D(v_5) >_i v_5$ and v_5 is not complete. Indeed, the complete interpretations of D are $com(D) = \{v_1, v_2, v_3, v_4, v_7, v_8\}$. The $\leq_i$-maximal elements among the admissible interpretations give rise to the preferred interpretations: $prf_3(D) = \{v_3, v_4, v_7\}$. As all preferred interpretations are two-valued, we have $mod_3(D) = prf_3(D)$. Finally we are interested in the stable models. To this end we check for each two-valued model if it is also a stable model. Considering v_3, we get $D^{-v_3} = \{\langle a, \neg\bot\rangle\}$. Hence $grd(D^{-v_3}) = \{\mathbf{t}\}$ and v_3 is stable. On the other hand, consider v_4 and observe that $D^{-v_4} = \{\langle a, \neg\bot\rangle, \langle c, \neg\bot \wedge d\rangle, \langle d, c\rangle\}$. We get $grd(D^{-v_4}) = \{\mathbf{tff}\}$ and therefore v_4 is not stable. As also v_7 can checked to be stable, we conclude that $stb_3(D) = \{v_3, v_7\}$. $\diamond$

The ADF semantics presented in Definition 31 are proper generalizations of the corresponding AF semantics. To state this formally we first introduce the ADF associated

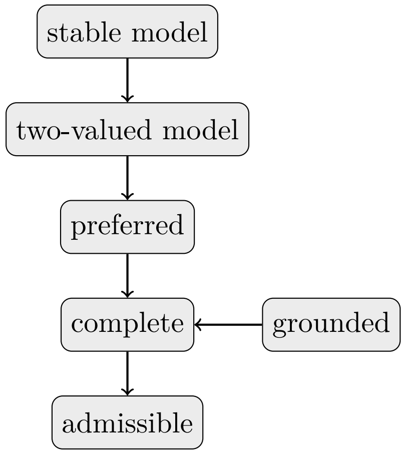


Figure 2.6: Relations between semantics of ADFs.

to an AF. Capturing the nature of attacks, the acceptance conditions of this ADF are conjunctions of negated atoms.

Definition 32. Given an AF $F = (A, R)$, the *ADF associated to F* is given by $D_F = \{\langle a, \varphi_a \rangle \mid a \in A\}$ where

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

for each argument $a \in A$.

The correspondence between semantics is established by the following result.

Proposition 3 ([55])**.** *For any AF $F = (A, R)$ it holds that $\sigma_3(F) = \sigma_3(D_F)$, where $\sigma \in \{adm, com, prf, stb\}$. Moreover it holds that $mod_3(D_F) = stb_3(D_F)$.*

The intuition behind stable models, inspired by logic programming [129], is to disallow cyclic support within a model. Due to the lack of support relations in ADFs associated to AFs, two-valued models and stable models coincide. Therefore both are proper generalizations of the stable semantics of AFs.

The relations between semantics also carry over from AFs to ADFs:

Proposition 4 ([55])**.** *In accordance with Figure 2.6, for any ADF $D \in ADF_{\mathfrak{A}}$, the following relations hold:*

- $stb_3(D) \subseteq mod_3(D) \subseteq prf_3(D) \subseteq com_3(D) \subseteq adm_3(D)$;
- $grd_3(F) \subseteq com_3(F)$.

Also SETAFs can be seen as a certain subclass of ADFs. Sets of attacking arguments are captured in the acceptance conditions of these ADFs as conjunctions of disjunctions of negated atoms.

Definition 33. Given a SETAF $S = (A, X)$, the *ADF associated to S* is given by $D_S = \{\langle a, \varphi_a \rangle \mid a \in A\}$ where

$$\varphi_a = \bigwedge_{(B,a)\in X} \bigvee_{b\in B} \neg b$$

for each argument $a \in A$.

Proposition 5. *For any SETAF $S = (A, X)$ it holds that $\sigma_3(S) = \sigma_3(D_S)$, where $\sigma_3 \in \{adm_3, com_3, prf_3, stb_3\}$. Moreover it holds that $mod_3(D_s) = stb_3(D_S)$.*

Proof. Given interpretation v and argument a, it holds that $\Gamma_{D_S}(v)(a) = \mathbf{t}$ if and only if $\forall w \in [v]_2 : w(\varphi_a) = \mathbf{t}$. The latter holds, by construction of φ_a, if and only if $\forall (B, a) \in X \exists b \in B : v(b) = \mathbf{f}$, meaning, by Definition 24, that a is acceptable wrt. v. On the other hand, it holds that $\Gamma_{D_S}(v)(a) = \mathbf{f}$ if and only if $\forall w \in [v]_2 : w(a) = \mathbf{f}$. The latter holds if and only if $\exists (B, a) \in X\ \forall b \in B : v(b) = \mathbf{t}$, which means that a is unacceptable wrt. v. Hence $\sigma_3(S) = \sigma_3(D_S)$ for $\sigma_3 \in \{adm_3, com_3, prf_3\}$ and $stb_3(S) = mod_3(D_S)$. Moreover, given some two-valued model $v \in mod_3(D_S)$, we have to check the grounded interpretation of $D^{-v} = \{\langle a, \varphi_a[x/\bot : v(x) = \mathbf{f}]\rangle \mid a \in v^{\mathbf{t}}\}$ in order to verify if also $v \in stb_3(D_S)$. Observing that $\forall (B, a) \in X \exists b \in B : v(b) = \mathbf{f}$, we get that $ac_a^{D^{-v}} \equiv \top$. Hence $grd_3(D^{-v}) = \{a \mapsto \mathbf{t} \mid a \in v^{\mathbf{t}}\}$. We conclude that $stb_3(D_s) = mod_3(D_S) = stb_3(S)$. □

Note that the ADFs associated to AFs and SETAFs, respectively, only contain attacking links and are therefore bipolar ADFs.

For two ADFs D and D', their union $D \cup D'$ is defined straightforwardly, but only for the case that their sets of arguments are disjoint, i.e. $A_D \cup A_{D'} = \emptyset$. Likewise, given some $S \subseteq A_D$, the restriction of D to S, $D|_S$, is only defined if for all $a \in S$ it holds that $par_D(a) \subseteq S$, i.e. S receives no links from $A_D \setminus S$.

2.4 Complexity

We assume familiarity with standard complexity concepts, such as P, NP and completeness. For comprehensive introductions we refer to [166] and [10]. In this section we recall the notions needed in this work.

Given a complexity class $\mathcal{C}$, a $\mathcal{C}$-oracle decides a given problem from $\mathcal{C}$ in one computation step. The class $\mathsf{P}^{\mathcal{C}}$ contains the problems that can be decided in polynomial time by a deterministic Turing machine with unrestricted access to a $\mathcal{C}$-oracle. The class $\mathsf{NP}^{\mathcal{C}}$ contains the problems that can be decided in polynomial time by a non-deterministic

$$\begin{tikzcd}[column sep=small]
 & \mathsf{NP} \arrow[dr, phantom, sloped, "\subseteq"] & & \Sigma_2^{\mathsf{P}} \arrow[dr, phantom, sloped, "\subseteq"] & & \Sigma_3^{\mathsf{P}} \arrow[dr, phantom, sloped, "\subseteq"] & \\
L \subseteq \mathsf{P} \arrow[ur, phantom, sloped, "\subseteq"] \arrow[dr, phantom, sloped, "\subseteq"] & & \Theta_2^{\mathsf{P}} \subseteq \Delta_2^{\mathsf{P}} \arrow[ur, phantom, sloped, "\subseteq"] \arrow[dr, phantom, sloped, "\subseteq"] & & \Theta_3^{\mathsf{P}} \subseteq \Delta_3^{\mathsf{P}} \arrow[ur, phantom, sloped, "\subseteq"] \arrow[dr, phantom, sloped, "\subseteq"] & & \cdots \\
 & \mathsf{coNP} \arrow[ur, phantom, sloped, "\subseteq"] & & \Pi_2^{\mathsf{P}} \arrow[ur, phantom, sloped, "\subseteq"] & & \Pi_3^{\mathsf{P}} \arrow[ur, phantom, sloped, "\subseteq"] &
\end{tikzcd}$$

Figure 2.7: Relations between complexity classes in the polynomial hierarchy.

Turing machine with unrestricted access to a $\mathcal{C}$-oracle. Finally, the class $\mathsf{coNP}^{\mathcal{C}}$ contains the problems whose complementary problems can be decided in polynomial time by a non-deterministic Turing machine with unrestricted access to a $\mathcal{C}$-oracle. This gives rise to the classes of the *polynomial hierarchy* as follows:

- $\Sigma_0^{\mathsf{P}} = \Pi_0^{\mathsf{P}} = \Delta_0^{\mathsf{P}} = P$;
- $\Delta_k^{\mathsf{P}} = \mathsf{P}^{\Sigma_{k-1}^{\mathsf{P}}}$ for $k \geq 1$;
- $\Sigma_k^{\mathsf{P}} = \mathsf{NP}^{\Sigma_{k-1}^{\mathsf{P}}}$ for $k \geq 1$;
- $\Pi_k^{\mathsf{P}} = \mathsf{coNP}^{\Sigma_{k-1}^{\mathsf{P}}}$ for $k \geq 1$.

In particular, note that $\Sigma_1^{\mathsf{P}} = \mathsf{NP}$ and $\Delta_2^{\mathsf{P}} = \mathsf{P}^{\mathsf{NP}}$. The classes Δ_k^{P} have been refined by the classes Θ_k^{P}, in which the number of oracle calls is bounded by $\mathcal{O}(\log n)$, where n is the size of the input. They are therefore sometimes also denoted as $\Delta_k^{\mathsf{P}}[\mathcal{O}(\log n)]$. Finally, L is the class of problems that can be decided by a Turing machine restricted to use an amount of memory which is logarithmic in the size of the input. The relations between the classes are depicted in Figure 2.7.

The complexity classes in the polynomial hierarchy have complete problems involving quantified Boolean formulas (QBFs).

Definition 34. By a *k-existential QBF* we denote a QBF of the form

$$Q_1 X_1 \ldots Q_k X_k \; \varphi(X_1, \ldots, X_k)$$

with $Q_1 = \exists$, $Q_2, \ldots, Q_k \in \{\exists, \forall\}$, $Q_i \neq Q_{i+1}$ for $1 \leq i < k$, and

(*i*) if $Q_k = \forall$ then φ is in DNF containing no monoms which are trivial for $X_1 \cup \cdots \cup X_{k-1}$;

(*ii*) if $Q_k = \exists$ then φ is in CNF containing no clauses which are trivial for $X_1 \cup \cdots \cup X_{k-1}$.

We call a monom m (resp. a clause c) *trivial* for X if all atoms occurring in m (resp. c) are contained in X.

Table 2.1: Complexity of reasoning with AFs.

σ	Cred_σ	Skept_σ	Ver_σ
cf	in L	trivial	in L
nai	in L	in L	in L
adm	NP-c	trivial	in L
com	NP-c	P-c	in L
prf	NP-c	Π_2^P-c	coNP-c
grd	P-c	P-c	P-c
stb	NP-c	coNP-c	in L
stg	Σ_2^P-c	Π_2^P-c	coNP-c
sem	Σ_2^P-c	Π_2^P-c	coNP-c
idl	in Θ_2^P	in Θ_2^P	in Θ_2^P
grd^*	NP-c	coNP-c	P-c

In particular, a 1-existential QBF is of the form $\exists X \varphi(X)$ with φ being in CNF without empty clauses. It is true if and only if $\varphi(X)$ is satisfiable.

The classes Θ_{k+1}^P (for $k \geq 1$) have the following complete problems [114, 197, 184], which we will make use of in the hardness proofs in Section 4.4:

GIVEN: k-existential QBFs $\Phi_1, \ldots, \Phi_m$ such that Φ_i being false implies Φ_{i+1} being false for $1 \leq i < m$,
DECIDE: whether $\max\{1 \leq i \leq m \mid \Phi_i \text{ is true}\}$ is odd.

The main reasoning tasks within argumentation formalisms are credulous and sceptical acceptance as well as the verification problem, each of them parametrized by a semantics σ. For AFs they are defined as follows:

- Cred_σ: Given an AF F and an argument $a \in A_F$, decide whether there exists some $E \in \sigma(F)$ such that $a \in E$.
- Skept_σ: Given an AF F and an argument $a \in A_F$, decide whether for all $E \in \sigma(F)$ it holds that $a \in E$.
- Ver_σ: Given an AF F and a set of arguments $S \subseteq A_F$, decide whether $S \in \sigma(F)$.

The complexity of the reasoning problems for AFs has been studied in [92, 97, 65, 108, 105] and is summarized in Table 2.1 for the semantics considered in this work. For a complexity class $\mathcal{C}$, we write $\mathcal{C}$-c if the problem is complete for $\mathcal{C}$.

The complexity of the respective reasoning tasks for ADFs is usually one level higher in the polynomial hierarchy. However, it turns out that this does not hold for bipolar

ADFs, which have (almost) the same complexity as AFs. A comprehensive analysis of the complexity of ADFs can be found in [190, 123].

CHAPTER 3

Expressiveness

There are various ways to assess the capabilities of knowledge representation formalisms. One way is to check their ability to model certain instances within application areas. Another, more systematic approach is to study the computational complexity of the involved reasoning tasks. In this section we study another property of knowledge representation formalisms and their semantics, namely the expressiveness in terms of the outcomes that can be achieved. Realizability is the ability of a formalism under a semantics to express specific desired sets of models. Signatures capture the exact expressiveness of a formalism under a semantics by collecting all sets of models that can be realized.

In formal argumentation, this line of research was initiated in [146] by not only comparing, but also exactly characterizing the expressiveness of most of the standard semantics of AFs. After introducing some preliminary notions in Section 3.1, we will complement this work on realizability in AFs in Section 3.2 in several ways: first we will study the expressiveness of two semantics that have been disregarded in [146], the complete and the resolution-based grounded semantics; then, we will establish some implications from the exact characterizations of signatures, in particular closure properties which will be of importance in Chapter 4; finally we will deal with complexity issues. Section 3.3 considers the intuitive subclass of compact AFs, where the set of arguments is assumed to be fixed, and shows the influence of this restriction on the expressiveness of semantics in AFs. Viewing AFs from an input-output perspective [17], we study the ability of semantics to map assignments of input arguments to sets of assignments of output arguments in Section 3.4.

The remainder of this chapter is then concerned with the question whether the increased modeling capacity of more advanced argumentation formalisms, in particular ADFs, also carries over to the expressiveness in our terms. Section 3.5 reviews and combines recent work on the realizability in ADFs [175, 188] and states a few implications thereof. Section 3.6 then presents an algorithmic approach to realizability which captures AFs and ADFs as well as the intermediate formalisms of SETAFs and bipolar ADFs in a modular way. It also allows us to derive additional relations with respect to realizability both within and between formalisms.

In the following we provide an overview of the publications concerning the results presented in this chapter. Section 3.2 is made up of results from [102] and [110]. Section 3.3 contains results from [30, 149, 31] as well as complementary unpublished results. The main results presented in Section 3.4 have been published in [130]. Section 3.5 reviews work by Pührer [175] and Strass [187, 189, 188] and states some observations combining these streams of work. Finally, Section 3.6 is based on [150].

3.1 Preliminaries

In this section we recall and define the general idea of realizability and signatures as originally introduced for AFs in [146]. The central notion is that of realizability, that is, the ability of a formalism to express a certain state of affairs.

Definition 35. Let $\mathcal{F}$ be a formalism and σ be a semantics of that formalism. A set of interpretations $\mathbb{I}$ is *realizable in $\mathcal{F}$ under σ* if there is some $\mathsf{kb} \in \mathcal{F}$ with $\sigma(\mathsf{kb}) = \mathbb{I}$. We say that kb *realizes $\mathbb{I}$ in $\mathcal{F}$ under σ.*

The task of realizing a set of interpretations $\mathbb{I}$ under a semantics σ is to find a concrete knowledge base kb which has $\sigma(\mathsf{kb}) = \mathbb{I}$. Intuitively, this is the inverse operation of evaluating a knowledge base under the semantics.

Signatures have been introduced to characterize the expressiveness of a formalism under a semantics by collecting all sets of interpretations which are realizable, i.e. the sets of interpretations which can be the outcome of the evaluation of a knowledge base of the formalism.

Definition 36. Let $\mathcal{F}$ be a formalism and σ be a semantics of that formalism. The *signature* $\Sigma^{\sigma}_{\mathcal{F}}$ of σ in $\mathcal{F}$ is defined as

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

An exact characterization of the signature of a semantics σ in a formalism $\mathcal{F}$ is achieved by finding a condition for sets of interpretations $\mathbb{I}$ which is necessary and sufficient for $\mathbb{I}$ to be realizable in $\mathcal{F}$ under σ. In other words, the condition γ has to be such that $\sigma(\mathsf{kb})$ fulfills γ for every $\mathsf{kb} \in \mathcal{F}$ and there is some $\mathsf{kb} \in \mathcal{F}$ with $\sigma(\mathsf{kb}) = \mathbb{I}$ whenever $\mathbb{I}$ is a set of interpretations fulfilling γ. Moreover, it is desirable that γ is minimal in the sense that it does not contain redundant subconditions.

Finally, we define the generic function for realizing specific sets of interpretations in a given formalism, leaving the exact construction of the realizing knowledge base open.

Definition 37. Let $\mathcal{F}$ be a formalism and σ a semantics of that formalism. The *realizing function* $\rho_\sigma^{\mathcal{F}}$ maps sets of interpretations for $\mathcal{F}$ to knowledge bases of $\mathcal{F}$ such that

- $\rho_\sigma^{\mathcal{F}}(\mathbb{I}) = \mathsf{kb}$ with $\sigma(\mathsf{kb}) = \mathbb{I}$ if $\mathbb{I} \in \Sigma_{\mathcal{F}}^{\sigma}$, and
- unspecified otherwise.

3.2 General Realizability in AFs

In the following, we begin with recalling results on realizability in AFs from from [146]. Then we will first complement these results by considering realizability under complete and resolution-based grounded semantics. After that we will argue that preferred and semi-stable are among the most expressive "reasonable" semantics. Finally, we will study certain closure properties of signatures, give quantitative limits of realizability, and show the complexity of deciding realizability.

3.2.1 Recalling Previous Results

The signatures of most of the standard semantics in AFs have been characterized in [146]. Here the objects of interest are sets of sets of arguments, called *extension-sets*, for which necessary and sufficient conditions for realizability under the respective semantics have been proven. We recall them for conflict-free, naive, stable, stage, admissible, preferred and semi-stable semantics in the following theorem. There have been some problems with the results for complete semantics in [146]. Therefore the next subsection is dedicated to complete semantics.

Theorem 1 ([146])**.** *The signatures are given by the following collections of extension-sets:*

$$\begin{aligned}
\Sigma_{AF}^{cf} &= \{\mathbb{S} \subseteq 2^{\mathfrak{A}} \mid \mathbb{S} \neq \emptyset,\ \mathbb{S} \text{ is downward-closed and tight}\}\\
\Sigma_{AF}^{nai} &= \{\mathbb{S} \subseteq 2^{\mathfrak{A}} \mid \mathbb{S} \neq \emptyset,\ \mathbb{S} \text{ is incomparable and } dcl(\mathbb{S}) \text{ is tight}\}\\
\Sigma_{AF}^{stb} &= \{\mathbb{S} \subseteq 2^{\mathfrak{A}} \mid \mathbb{S} \text{ is incomparable and tight}\}\\
\Sigma_{AF}^{stg} &= \{\mathbb{S} \subseteq 2^{\mathfrak{A}} \mid \mathbb{S} \neq \emptyset,\ \mathbb{S} \text{ is incomparable and tight}\}\\
\Sigma_{AF}^{adm} &= \{\mathbb{S} \subseteq 2^{\mathfrak{A}} \mid \mathbb{S} \neq \emptyset,\ \mathbb{S} \text{ is conflict-sensitive and contains } \emptyset\}\\
\Sigma_{AF}^{prf} &= \{\mathbb{S} \subseteq 2^{\mathfrak{A}} \mid \mathbb{S} \neq \emptyset,\ \mathbb{S} \text{ is incomparable and conflict-sensitive}\}\\
\Sigma_{AF}^{sem} &= \{\mathbb{S} \subseteq 2^{\mathfrak{A}} \mid \mathbb{S} \neq \emptyset,\ \mathbb{S} \text{ is incomparable and conflict-sensitive}\}
\end{aligned}$$

Besides these exact characterizations of signatures, [146] also featured canonical realizations for extension-sets contained in the signatures. For instance, realizing a set of interpretations $\mathbb{S} \in \Sigma_{\mathrm{AF}}^{\sigma}$ for $\sigma \in \{cf, nai\}$ is achieved by the AF $F_{cf}(\mathbb{S}) = (Args_{\mathbb{S}}, (Args_{\mathbb{S}} \times Args_{\mathbb{S}}) \setminus Pairs_{\mathbb{S}})$, i.e. the AF containing the arguments that occur in any element of $\mathbb{S}$ and an attack between any pair of arguments which does not occur together in an element of $\mathbb{S}$. Constructions for the other semantics build up on F_{cf}, but use more involved concepts in addition. The general notion of a realizing function for AFs follows Definition 37. It also leaves the exact constructions unspecified but gives a concrete AF for extension-sets which are not realizable:

Definition 38. Given a semantics σ, the AF realizing function $\rho_\sigma^{\mathrm{AF}} : 2^{2^{\mathfrak{A}}} \mapsto AF_{\mathfrak{A}}$ maps extension-sets to AFs such that

- $\rho_\sigma^{\mathrm{AF}}(\mathbb{S}) = F$ with $\sigma(F) = \mathbb{S}$ if $\mathbb{S} \in \Sigma_{\mathrm{AF}}^{\sigma}$, and
- $\rho_\sigma^{\mathrm{AF}}(\mathbb{S}) = (\emptyset, \emptyset)$ otherwise.

The signatures given in Theorem 1 are related to each other as follows.

Theorem 2 ([146])**.** *The signatures of AF semantics are related as follows:*

- $\Sigma_{AF}^{nai} \subset \Sigma_{AF}^{stg} = \Sigma_{AF}^{stb} \setminus \{\emptyset\} \subset \Sigma_{AF}^{sem} = \Sigma_{AF}^{prf}$,
- $\Sigma_{AF}^{cf} \subset \Sigma_{AF}^{adm}$,
- $\Sigma_{AF}^{\sigma} \cap \Sigma_{AF}^{\tau} = \{\{\emptyset\}\}$ *for* $\sigma \in \{nai, stg, stb, sem, prf\}$ *and* $\tau \in \{cf, adm\}$.

These relations are in line with the ones obtained from intertranslatability of argumentation semantics [109, 107]. There the authors are interested, given two AF semantics σ and τ, in translations from AFs to AFs such that the τ-extensions of the transformed AF coincide with the σ-extensions of the original AF. If such a translation exists, then τ is at least as expressive as σ, that is $\Sigma_{\mathrm{AF}}^{\sigma} \subseteq \Sigma_{\mathrm{AF}}^{\tau}$ in our terms.

Remark 1. We use this opportunity for a clarification of an issue which lead to major misunderstandings. That is the fact that there is a fundamental difference, given two semantics σ and τ, between the statement $\sigma(F) \subseteq \tau(F)$ for each $F \in AF_{\mathfrak{A}}$ and the statement $\Sigma_{\mathrm{AF}}^{\sigma} \subseteq \Sigma_{\mathrm{AF}}^{\tau}$. The latter statement makes no claims about the σ- and τ-extensions of a particular AF, but says that for each AF F, there exists an AF G such that $\sigma(F) = \tau(G)$. For instance, it holds for each AF $F \in AF_{\mathfrak{A}}$ that $sem(F) \subseteq prf(F)$ and for certain AFs this relation is even proper, but on the other hand we have $\Sigma_{\mathrm{AF}}^{sem} = \Sigma_{\mathrm{AF}}^{prf}$, i.e. semi-stable and preferred semantics are able to express exactly the same set of extension-set.

3.2.2 Complete Semantics

An exact characterization of the signature of complete semantics in AFs turns out to be way more intricate. We narrow it down by giving necessary conditions for extension-sets to be contained in the signature.

In contrast to admissible semantics, extension-sets obtained from complete semantics are not necessarily conflict-sensitive, as illustrated in the following example.

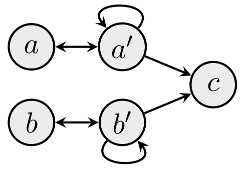


Figure 3.1: Argumentation framework F discussed in Examples 10 and 12.

Example 10. Consider the AF F depicted in Figure 3.1. It can be checked that $\mathit{com}(F) = \{\emptyset, \{a\}, \{b\}, \{a, b, c\}\}$. Moreover observe that $\{a\} \cup \{b\} \notin \mathit{com}(F)$, but $(a, a), (a, b), (b, a), (b, b) \in \mathit{Pairs}_{\mathit{com}(F)}$. Hence $\mathit{com}(F)$ is not conflict-sensitive. ◊

We introduce a property which is less restrictive than conflict-sensitive, which we then show to be a necessary condition for extension-sets to be realizable under complete semantics and therefore to be contained in the *com*-signature.

Definition 39. Given an extension-set $\mathbb{S} \subseteq 2^{\mathfrak{A}}$ and $E \subseteq \mathfrak{A}$. We define the *completion-sets* $\mathbb{C}_{\mathbb{S}}(E)$ of E in $\mathbb{S}$ as the set of $\subseteq$-minimal sets $S \in \mathbb{S}$ with $E \subseteq S$.

In words, completion-sets just give the "next" (in terms of supersets) elements contained in $\mathbb{S}$.

Definition 40. A set $\mathbb{S} \subseteq 2^{\mathfrak{A}}$ is called *com-closed* if for each $\mathbb{T} \subseteq \mathbb{S}$ the following holds: if $(a, b) \in \mathit{Pairs}_{\mathbb{S}}$ for each $a, b \in \mathit{Args}_{\mathbb{T}}$, then $\mathit{Args}_{\mathbb{T}}$ has a unique completion-set in $\mathbb{S}$, i.e. $|\mathbb{C}_{\mathbb{S}}(\mathit{Args}_{\mathbb{T}})| = 1$.

The following example illustrates the idea of com-closed extension-sets.

Example 11. Consider the following extension-set:

$$\mathbb{S} = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b, d\}, \{a, c, e\}, \{b, c, f\}\}.$$

In order to check if $\mathbb{S}$ is com-closed, we have to consider the sets $\{a\} \cup \{b\}$, $\{a\} \cup \{c\}$, $\{b\} \cup \{c\}$, and $\{a\} \cup \{b\} \cup \{c\}$, as all pairs of arguments in these sets are contained in $\mathit{Pairs}_{\mathbb{S}}$, but the sets themselves are not contained in $\mathbb{S}$. We observe that $\mathbb{C}_{\mathbb{S}}(\{a\} \cup \{b\}) = \{\{a, b, d\}\}$, $\mathbb{C}_{\mathbb{S}}(\{a\} \cup \{c\}) = \{\{a, c, e\}\}$, and $\mathbb{C}_{\mathbb{S}}(\{b\} \cup \{c\}) = \{\{b, c, f\}\}$, but $\mathbb{C}_{\mathbb{S}}(\{a\} \cup \{b\} \cup \{c\}) = \emptyset$. Hence $\mathbb{S}$ is not com-closed. Adding, for instance, $\{a, b, c, g\}$ to $\mathbb{S}$ solves the problem. Then, $\mathbb{C}_{\mathbb{S} \cup \{\{a,b,c,g\}\}}(\{a\} \cup \{b\} \cup \{c\}) = \{\{a, b, c, g\}\}$ and the other completion sets remain unchanged, hence $\mathbb{S} \cup \{\{a, b, c, g\}\}$ is com-closed. Finally, adding another set $\{a, b, c, h\}$ means that $\mathbb{C}_{\mathbb{S} \cup \{\{a,b,c,g\},\{a,b,c,h\}\}}(\{a\} \cup \{b\} \cup \{c\}) = \{\{a, b, c, g\}, \{a, b, c, h\}\}$, i.e. there is no unique completion set for $\{a\} \cup \{b\} \cup \{c\}$, hence $\mathbb{S} \cup \{\{a, b, c, g\}, \{a, b, c, h\}\}$ is not com-closed. ◊

The intuitive meaning of an extension-set being com-closed is the following. Consider an extension-set $\mathbb{S}$ and a set of elements $\mathbb{T}$ thereof. Now assume $\mathbb{S}$ gives no evidence of a conflict between arguments in $Args_{\mathbb{T}}$. Then, in contrast to the case when $\mathbb{S}$ is conflict-sensitive, $\mathbb{S}$ does not necessarily have to contain $Args_{\mathbb{T}}$, but has to contain a unique superset of $Args_{\mathbb{T}}$, the completion-set.

Lemma 3. *Each conflict-sensitive extension-set is com-closed.*

Proof. Consider a conflict-sensitive extension-set $\mathbb{S}$ and an arbitrary subset $\mathbb{T} \subseteq \mathbb{S}$. Then $(a, b) \in Pairs_{\mathbb{S}}$ for each $a, b \in Args_{\mathbb{T}}$ implies $Args_{\mathbb{T}} \in \mathbb{S}$, i.e. $\mathbb{C}_{\mathbb{S}}(Args_{\mathbb{T}}) = \{Args_{\mathbb{T}}\}$. Hence $\mathbb{S}$ is com-closed. $\square$

Note that in case of incomparable sets, the notions conflict-sensitive and com-closed coincide. In anticipation of the following result, this coincidence reflects the fact that all preferred extensions are complete.

Proposition 6. *For each AF F, it holds that $com(F) \neq \emptyset$, $com(F)$ is com-closed and $\bigcap com(F) \in com(F)$.*

Proof. First note that there is always at least one complete extension, namely the grounded extension. Moreover the grounded extension is the unique $\subseteq$-minimal complete extension and hence $\bigcap com(F) \in com(F)$. Finally consider a set of complete extensions $\mathbb{E} \subseteq com(F)$ such that $(a, b) \in Pairs_{com(F)}$ for each $a, b \in Args_{\mathbb{E}}$. By Lemma 2, $Args_{\mathbb{E}}$ is an admissible set and thus can be extended to a unique complete extension $E' \supseteq Args_{\mathbb{E}}$ by iteratively adding all defended arguments. Therefore $com(F)$ is com-closed. $\square$

Example 12. Again consider the AF F depicted in Figure 3.1 and recall that $com(F) = \{\emptyset, \{a\}, \{b\}, \{a, b, c\}\}$. While we have seen in Example 10 that $com(F)$ is not conflict-sensitive, it turns out that it is com-closed, confirming Proposition 6. In particular, $\{a\} \cup \{b\}$ has a unique completion set, i.e. $\mathbb{C}_{com(F)}(\{a\} \cup \{b\}) = \{\{a, b, c\}\}$. $\diamondsuit$

Each extension-set $\mathbb{S}$ which is realizable under the admissible semantics, can also be realized under the complete semantics: given the AF F realizing $\mathbb{S}$ under *adm*, we extend F in such a way that for each $a \in Args_{\mathbb{S}}$ we add a self-attacking argument with an attack from and to a. This has the effect that every argument needs itself to be defended, hence every admissible extension is also complete. The formal details can be taken from the translation from *adm* to *com* in [108]. Together with Example 10 the following result immediately follows.

Theorem 3. *It holds that $\Sigma_{AF}^{adm} \subset \Sigma_{AF}^{com}$.*

However, the property of being com-closed (cf. Definition 40) so far only gives an approximation of the sets which can be realized. It is not a sufficient condition for realizability under complete semantics, as the following example shows.

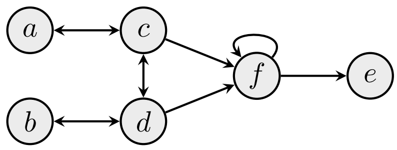


Figure 3.2: AF F such that $grd^*(F) = \{\{a,b\},\{a,d,e\},\{b,c,e\}\} \notin \Sigma^{stb}_{\text{AF}}$.

Example 13. Let $\mathbb{S} = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a,b,c\}, \{a,d,e\}, \{b,d,f\}, \{x,c\}, \{x,d\}\}$. $\mathbb{S}$ is com-closed and satisfies $\bigcap \mathbb{S} = \emptyset \in \mathbb{S}$. We argue that $\mathbb{S}$ is not *com*-realizable. Towards a contradiction consider an AF F such that $com(F) = \mathbb{S}$. First, as $\{x,c\},\{x,d\} \in \mathbb{S}$ and no superset of $\{x,c,d\}$ is contained in $\mathbb{S}$, there must be a conflict between c and d. In case $(c,d) \in R_F$ and $(d,c) \notin R_F$, $\{x,d\}$ is not admissible (as $(x,c) \in Pairs_{\mathbb{S}}$). Similar for $(d,c) \in R_F$ and $(c,d) \notin R_F$. Thus $(c,d),(d,c) \in R_F$. Now consider $\{a,b\}$ which must be admissible as $\{a\}$ and $\{b\}$ are admissible and $(a,b) \in Pairs_{\mathbb{S}}$. But as $\{a,b\} \notin \mathbb{S}$ it has to defend argument c ($\mathbb{C}_{\mathbb{S}}(\{a,b\}) = \{\{a,b,c\}\}$). But we have $(d,c) \in R_F$ and $(a,d),(b,d) \in Pairs_{\mathbb{S}}$. Hence $\{a,b\}$ cannot defend c, meaning that some $S \subseteq \{a,b\}$ with $c \notin S$ is a complete extension of F, a contradiction to $com(F) = \mathbb{S}$. $\Diamond$

Reflecting on Theorem 3 and Example 13, we conclude that, in order to exactly characterize Σ^{com}_{AF}, it is necessary to come up with a condition which is more restrictive than com-closed but less demanding than conflict-sensitive. The exact formulation of this condition is subject to future research.

3.2.3 Resolution-based Grounded Semantics

Realizability under the resolution-based grounded semantics was left open in [102]. In this subsection we relate its signature to the signatures of the other semantics and show a rather strong necessary condition for realizability.

The first observation is that evaluation under the resolution-based grounded semantics can lead to outcomes which are not possible under stable semantics.

Proposition 7. *It holds that* $\Sigma^{grd^*}_{AF} \not\subseteq \Sigma^{stb}_{AF}$.

Proof. Consider the AF F depicted in Figure 3.2. One can check that $grd^*(F) = \{\{a,b\},\{a,d,e\},\{b,c,e\}\}$. However, $grd^*(F)$ is not tight, since we have $\{a,b\} \in grd^*(F)$ and $\{a,b,e\} \notin grd^*(F)$, but both (a,e) and (b,e) are contained in $Pairs_{grd^*(F)}$. Thus, $grd^*(F) \notin \Sigma^{stb}_{\text{AF}}$ and, consequently, $\Sigma^{grd^*}_{\text{AF}} \not\subseteq \Sigma^{stb}_{\text{AF}}$. $\square$

In other words, the resolution-based grounded semantics is capable of realizing extension-sets which are not realizable by stable semantics or weaker semantics such as stage or naive semantics. Next, we will show that, in contrast, $\Sigma^{grd^*}_{\text{AF}} \subseteq \Sigma^{prf}_{\text{AF}}$ holds.

To this end we recall some definitions and results from [16], using slightly different notation though. The initial SCCs of an AF are those strongly connected components which receive no attacks from an argument of any other SCC (cf. Definition 21).

Definition 41. Given an AF F, the set of *initial SCCs* $\mathfrak{I}(F) \subseteq SCCs(F)$ contains exactly those SCCs of F which have no incoming attacks, i.e. $F' \in \mathfrak{I}(F)$ if and only if there is no $(a, b) \in R_F$ with $a \in (A_F \setminus A_{F'})$ and $b \in A_{F'}$. The set of *relevant initial SCCs* $\mathfrak{I}^*(F) \subseteq \mathfrak{I}(F)$ contains exactly those $G \in \mathfrak{I}(F)$ which are self-attack-free, symmetric and the underlying graph is acyclic.

The following lemma containing an alternative, recursive definition of grd^* is immediate by Theorem 2 and Lemma 9 of [16].

Lemma 4. *Let $F \in AF_{\mathfrak{A}}$ such that $grd^*(F) \neq grd(F)$. It holds that*

1. *$\mathfrak{I}^*(cut_{grd(F)}(F))$ is non-empty; and*

2. *$S \in grd^*(F)$ iff $S = (T \cup U \cup V)$ where $T = grd(F)$, $U \in stb(\bigcup \mathfrak{I}^*(cut_{grd(F)}(F)))$, and $V \in grd^*(cut_{(T \cup U)}(F))$.*

Lemma 4 gives a more computational perspective of resolution-based grounded semantics. Intuitively, it states that the resolution-based grounded extensions can be computed by iteratively (i) computing the grounded extension and eliminating its range and (ii) computing the stable extensions of the initial SCCs and eliminating their range. It also helps to show the following important observation.

Lemma 5. *Given an AF $F \in AF_{\mathfrak{A}}$ and sets of arguments $S_1, S_2 \subseteq A_F$ with $S_1 \neq S_2$. Then, $S_1, S_2 \in grd^*(F)$ implies $S_1 \rightarrow_F S_2$.*

Proof. Consider an AF F with $S_1, S_2 \in grd^*(F)$ such that $S_1 \neq S_2$. First observe that $grd(F) \neq grd^*(F)$. Now let $T = grd(F)$ and $F' = \bigcup \mathfrak{I}^*(cut_T(F))$ (note that $\mathfrak{I}^*(cut_T(F))$ is non-empty by Lemma 4.1). We follow by Lemma 4.2 that, for $i \in \{1, 2\}$, $T \subset S_i$ and $\exists U_i \in stb(F') : U_i \subseteq S_i$. If $U_1 \neq U_2$ we are done because U_1, being stable in F', must attack all arguments in $U_2 \setminus U_1$. On the other hand if $U_1 = U_2$, let $F'' = cut_{(T \cup U_1)}(F) = cut_{(T \cup U_2)}(F)$. We have, also by Lemma 4.2, that $(S_i \setminus (T \cup U_i)) \in grd^*(F'')$ and we can reason as above. Since F is finite, we must reach a point where indeed $U_1 \neq U_2$ and U_1 attacks U_2. Hence $S_1 \rightarrow_F S_2$. □

This is enough to show that everything realizable under the resolution-based grounded semantics is also realizable under the preferred semantics.

Proposition 8. *It holds that $\Sigma_{AF}^{grd^*} \subseteq \Sigma_{AF}^{prf}$.*

Proof. For any AF F, $grd^*(F)$ is by definition an incomparable and non-empty extension-set. It remains to show that $grd^*(F)$ is conflict-sensitive. By Lemma 5, for any distinct $S_1, S_2 \in grd^*(F)$ it is the case that $S_1 \rightarrow_F S_2$ holds. Hence, by conflict-freeness of resolution-based grounded extensions, $\exists s_1, s_2 \in (S_1 \cup S_2) : (s_1, s_2) \notin Pairs_{grd^*(F)}$. $\square$

The following result concerning realizability shows certain and severe limits of expressiveness the resolution-based grounded semantics suffers from.

Proposition 9. *Let F be an AF and $S \subset A_F$. There are no pairwise disjoint sets $S_1, S_2, S_3 \subseteq A_F$ such that $\{(S \cup S_1), (S \cup S_2), (S \cup S_3)\} \subseteq grd^*(F)$.*

Proof. Assume there are pairwise disjoint sets $S_1, S_2, S_3 \subseteq A_F$, i.e. $S_1 \cap S_2 = S_1 \cap S_3 = S_2 \cap S_3 = \emptyset$, such that $\{(S \cup S_1), (S \cup S_2), (S \cup S_3)\} \subseteq grd^*(F)$. Let $T = grd(F)$. By Lemma 4, for $F' = \bigcup \mathfrak{I}^*(cut_T(F))$ and $i \in \{1, 2, 3\}$, it holds that $T \subseteq S$ and $\exists U_i \in stb(F') : U_i \subseteq (S \cup S_i)$. Note that each U_i has full range in F' (i.e. $(U_i)^+_{F'} = A_{F'}$). Hence $(A_{F'} \setminus U_i) \cap S_i = \emptyset$ for each $i \in \{1, 2, 3\}$, since otherwise, if there existed an $s_i \in (A_{F'} \setminus U_i) \cap S_i$, this s_i would have to be attacked by U_i, a contradiction to conflict-freeness of $S \cup S_i$.

First assume all U_1, U_2, and U_3 are pairwise different. All of them being stable in F' and S_1, S_2, S_3 being pairwise disjoint means that, for every $i \in \{1, 2, 3\}$, each $u_i \in (U_i \setminus S)$ must be attacked by each $U_j \setminus S$ for $j \in (\{1, 2, 3\} \setminus \{i\})$. But this means that there must be a cycle in the undirected graph underlying F', a contradiction.

Next assume that two U_is coincide and the third is different, w.l.o.g. $U_1 = U_2$ and $U_1 \neq U_3$. $U_1 = U_2$ means, since $S_1 \cap S_2 = \emptyset$, that $U_1, U_2 \subseteq S$. $U_1 \neq U_3$ means, on the other hand, that there must be some $u_3 \in (U_3 \cap S_3)$. But this contradicts U_1 and U_2 being stable, since S cannot attack u_3 due to $(S \cup S_3) \in grd^*(F)$.

Finally, assume $U_1 = U_2 = U_3$ (i.e. $U_i \subseteq S$) and let $F'' = cut_{(T \cup U_i)}(F)$ for any $i \in \{1, 2, 3\}$, and $S' = S \setminus (T \cup U_i)$. By Lemma 4.2 we get $\{(S' \cup S_1), (S' \cup S_2), (S' \cup S_3)\} \subseteq grd^*(F'')$. Hence we can reason as above. Since F is finite and $U_1, U_2, U_3 \neq \emptyset$, we must arrive at a contradiction of the former two cases at some point. $\square$

This already suggests quite strong limitations concerning the structural diversity of extension-sets under the resolution-based grounded semantics. Two particular cases of this impossibility are given by the following corollaries.

Corollary 1. *Let $\mathbb{S}$ be an extension-set containing three pairwise disjoint sets S_1, S_2, and S_3. There is no AF F such that $grd^*(F) \supseteq \{S_1, S_2, S_3\}$.*

Corollary 2. *Given an AF F and arguments $a, b \in A_F$ such that $\{a\}, \{b\} \in grd^*(F)$. Then, $grd^*(F) = \{\{a\}, \{b\}\}$.*

Proof. If $\{a\}, \{b\} \in grd^*(F)$ then any further extension would have to be either disjoint or not incomparable to $\{a\}$ and $\{b\}$, both contradictions to previous observations. $\square$

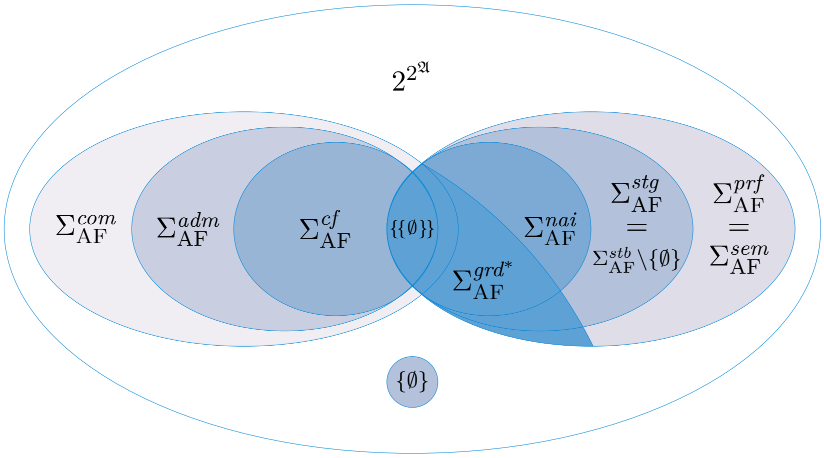


Figure 3.3: Venn diagram illustrating relations between signatures.

Example 14. Consider, as in Proposition 9, sets of arguments $S, S_1, S_2, S_3 \subseteq \mathfrak{A}$ such that $S_1 \cap S_2 = S_1 \cap S_3 = S_2 \cap S_3 = \emptyset$. In contrast to *grd**, the extension-set given by $\{(S \cup S_1), (S \cup S_2), (S \cup S_3)\}$ is realizable under *nai*, *stb*, *prf*, *sem*, and *stg*. The realizing AF is $F = (S \cup S_1 \cup S_2 \cup S_3, \{(s_i, s_j) \mid i, j \in \{1,2,3\}, i \neq j, s_i \in S_i, s_j \in S_j\})$. It is easy to see that each $S \cup S_i$ is maximally conflict-free and has full range in F. $\diamondsuit$

As a concrete instance, our results show that there is no AF F, such that $grd^*(F) = \{\{a\}, \{b\}, \{c\}\}$, while the other semantics are indeed able to express this extension-set, in particular, stable and preferred semantics when applied to a clique $\{a, b, c\}$.

The following relations now immediately follow.

Corollary 3. *$\Sigma_{AF}^{grd^*} \subset \Sigma_{AF}^{prf}$ and $\Sigma_{AF}^{stb} \not\subseteq \Sigma_{AF}^{grd^*}$.*

We leave an exact characterization of $\Sigma_{\mathrm{AF}}^{grd^*}$ for future work.

3.2.4 Relations and Upper Bounds of Expressiveness

The previous considerations on complete and resolution-based grounded semantics complement the picture of relations between signatures presented in Theorem 2. The relations are depicted in the Venn diagram in Figure 3.3. The right side of the figure shows the signatures of these semantics providing only incomparable extension-sets. The only extension-set they have in common with the signatures of conflict-free and admissible semantics is the one only containing the empty extension. On the other hand, the intersection with $\Sigma_{\mathrm{AF}}^{com}$ contains all extension-sets $\mathbb{S}$ with $|\mathbb{S}| = 1$. This is also the signature of single status semantics such as grounded and ideal. Finally note the isolated position of $\{\emptyset\}$, i.e. the extension-set containing no extension. It is only contained in $\Sigma_{\mathrm{AF}}^{stb}$.

All semantics considered in this work as well as most semantics in the literature[1] follow the principle of conflict-freeness of extensions. That means that the set of extensions of any AF F must be contained in $cf(F)$. A related property is shared by those semantics which always yield incomparable extension-sets, i.e. the naive, stable, stage, preferred, semi-stable and resolution-based grounded semantics. That is, given an arbitrary AF F, between any two extensions of F there must be at least one attack. If there was not, their union would be an extension instead. This property can be seen as another principle in the spirit of [11] which should be fulfilled by every reasonable semantics for abstract argumentation. The following proposition shows that these principles are captured by the concept of conflict-sensitivity, i.e. extension-sets under such semantics are always conflict-sensitive.

Proposition 10. *Consider an arbitrary semantics $\sigma : AF_{\mathfrak{A}} \mapsto 2^{2^{\mathfrak{A}}}$ such that for any $F \in AF_{\mathfrak{A}}$ it holds that $\sigma(F) \subseteq cf(F)$ and for all $S_1, S_2 \in \sigma(F)$ with $S_1 \neq S_2$ there exist $a, b \in S_1 \cup S_2$ with $(a, b) \in R_F$. Then for each AF F it holds that $\sigma(F)$ is conflict-sensitive.*

Proof. Let $F \in AF_{\mathfrak{A}}$ and $S_1, S_2 \in \sigma(F)$. By assumption there exist w.l.o.g. $a \in S_1$ and $b \in S_2$ with $(a, b) \in R$. Now since $\sigma(F) \subseteq cf(F)$, there is no $T \in \sigma(F)$ with $T \supseteq \{a, b\}$, hence $(a, b) \notin Pairs_{\sigma(F)}$. Therefore $\sigma(F)$ is conflict-sensitive. □

The characterization of the signatures of preferred and semi-stable semantics also shows that these semantics enjoy maximal expressiveness within reasonable semantics as defined in Proposition 10. That is, no semantics which does not allow conflicts within extensions and at the same time always guarantees a conflict between two different extensions can express more than preferred and semi-stable semantics, respectively.

Theorem 4. *Consider an arbitrary semantics $\sigma : AF_{\mathfrak{A}} \mapsto 2^{2^{\mathfrak{A}}}$ such that for any $F \in AF_{\mathfrak{A}}$ it holds that $\sigma(F) \neq \emptyset$, $\sigma(F) \subseteq cf(F)$ and for all $S_1, S_2 \in \sigma(F)$ with $S_1 \neq S_2$ there exist $a, b \in S_1 \cup S_2$ with $(a, b) \in R_F$. It holds that $\Sigma^{\sigma}_{AF} \subseteq \Sigma^{prf}_{AF}$ and $\Sigma^{\sigma}_{AF} \subseteq \Sigma^{sem}_{AF}$.*

Proof. Given any AF F, the fact that for all $S_1, S_2 \in \sigma(F)$ with $S_1 \neq S_2$ there are $a, b \in S_1 \cup S_2$ with $(a, b) \in R$ means, since also $S_1, S_2 \in cf(F)$, that $a \notin S_2$ and $b \notin S_1$ (or the other way round), hence $\sigma(F)$ is incomparable. Therefore the result follows from Theorem 1 and Proposition 10. □

3.2.5 Closure Properties of Signatures

With the results on signatures at hand, we now provide some interesting implications. In particular, we will study closure of signatures under subset as well as under intersection. Being of interest of its own, it will be fundamental in Section 4.2, when rationality postulates require revision operators to realize a certain subset of an extension-set or the intersection of extension-sets.

[1] Exceptions can be found in [138, 9, 133].

We begin by studying the question whether an arbitrary subset of an extension-set can be realized. More practically, that is, given an AF F, whether we can change F in a way to remove certain extensions. The following two lemmas will allow us to answer this question positively for stable, stage, preferred and semi-stable semantics.

Lemma 6. *For an incomparable extension-set $\mathbb{S} \subseteq 2^{\mathfrak{A}}$ it holds that if $\mathbb{S}$ is tight then each $\mathbb{S}' \subseteq \mathbb{S}$ is tight.*

Proof. Consider some tight and incomparable extension-set $\mathbb{S}$ and some $\mathbb{S}' \subseteq \mathbb{S}$. Then $Pairs_{\mathbb{S}'} \subseteq Pairs_{\mathbb{S}}$ and, as $\mathbb{S}$ is incomparable, $S \cup \{a\} \notin \mathbb{S}$ iff $S \cup \{a\} \notin \mathbb{S}'$ for all $S \in \mathbb{S}$, $a \in Args_{\mathbb{S}}$. Thus, since $\mathbb{S}$ is tight by the hypothesis, $\mathbb{S}'$ is tight. □

Lemma 7. *For an incomparable extension-set $\mathbb{S} \subseteq 2^{\mathfrak{A}}$, it holds that if $\mathbb{S}$ is conflict-sensitive then each $\mathbb{S}' \subseteq \mathbb{S}$ is conflict-sensitive.*

Proof. Recall that for incomparable $\mathbb{S}$, checking conflict-sensitivity reduces to check for each $A, B \in \mathbb{S}$ with $A \neq B$, whether there exist $a, b \in A \cup B$ such that $(a, b) \notin Pairs_{\mathbb{S}}$. It is easy to see that this property still holds for $\mathbb{S}' \subseteq \mathbb{S}$, since then, $Pairs_{\mathbb{S}'} \subseteq Pairs_{\mathbb{S}}$. □

The above lemmas give rise to the following result.

Theorem 5. *Given an arbitrary AF F it holds that*

- *for any $\mathbb{S} \subseteq stb(F)$ there exists an AF F' such that $stb(F') = \mathbb{S}$;*
- *for any $\mathbb{S} \subseteq \sigma(F)$ with $\mathbb{S} \neq \emptyset$ there exists an AF F' such that $\sigma(F') = \mathbb{S}$, where $\sigma \in \{stg, prf, sem\}$.*

Proof. Inspecting the characterizations of signatures in Theorem 1, the result follows from Lemma 6 for *stb* and *stg* and from Lemma 7 for *prf* and *sem*. □

Resolution-based grounded semantics appears to be closed under subset as well, due to the lack of an exact characterization we have to leave the definite answer open though. Conflict-free, admissible and complete as well as naive semantics are not closed under subset.

Example 15. First consider the AF $F = (\{a, b\}, \{(a, b), (b, a)\})$ and a semantics $\sigma \in \{cf, adm, com\}$. It holds that $\sigma(F) = \{\emptyset, \{a\}, \{b\}\}$. Now let $\mathbb{S} = \{\{a\}, \{b\}\} \subseteq \sigma(F)$ and observe $\mathbb{S} \notin \Sigma^{\sigma}_{\mathrm{AF}}$, since $\mathbb{S}$ is not downward-closed (for $\sigma = cf$), $\emptyset \notin \mathbb{S}$ (for $\sigma = adm$), and $\bigcap \mathbb{S} \notin \mathbb{S}$ (for $\sigma = com$).

For naive semantics consider the AF G depicted in Figure 3.4 which has $nai(G) = \{\{a, b', c'\}, \{a', b, c'\}, \{a', b', c\}, \{a', b', c'\}\}$. Now it holds that the extension-set $\mathbb{T} = \{\{a, b', c'\}, \{a', b, c'\}, \{a', b', c\}\} \subseteq nai(G)$ is not realizable under the naive semantics: $\{a', b'\} \in dcl(\mathbb{T})$ and $\{a', b', c'\} \notin dcl(\mathbb{T})$, but $(a', c'), (b', c') \in Pairs_{dcl(\mathbb{T})} = Pairs_{\mathbb{T}}$, hence

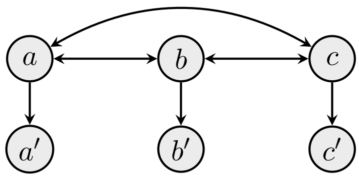


Figure 3.4: AF G with $nai(G) = \{\{a, b', c'\}, \{a', b, c'\}, \{a', b', c\}, \{a', b', c'\}\}$.

$dcl(\mathbb{T})$ is not tight. In practical terms, any AF trying to realize $\mathbb{T}$ must not have attacks between arguments $\{a', b', c'\}$, since $(a', b'), (b', c'), (a', c') \in Pairs_{\mathbb{T}}$, hence there is some naive extension $E \supseteq \{a', b', c'\}$. Therefore $\mathbb{T} \notin \Sigma^{nai}_{\mathrm{AF}}$. ♢

The second result examines the question whether the intersection of two extension-sets under a given semantics σ can always be realized. The answer is clearly yes for those semantics which are closed under subset (cf. Theorem 5). With the exact characterizations of signatures in Theorem 1 at hand we can answer this question positively also for the other semantics, except complete, for which we will provide a counterexample.

Theorem 6. *Given arbitrary AFs F_1 and F_2 it holds that*

- *there exists an AF F such that $\sigma(F) = \sigma(F_1) \cap \sigma(F_2)$ if $\sigma(F_1) \cap \sigma(F_2) \neq \emptyset$ for $\sigma \in \{cf, adm, nai, stg, prf, sem\}$;*
- *there exists an AF F such that $stb(F) = stb(F_1) \cap stb(F_2)$.*

Proof. Let $\mathbb{S}_1 = \sigma(F_1)$, $\mathbb{S}_2 = \sigma(F_2)$, and $\mathbb{S} = \mathbb{S}_1 \cap \mathbb{S}_2$. We know that $\mathbb{S}_1$ and $\mathbb{S}_2$ fulfill the properties according to the signature of σ. We have to show that $\sigma(F_1) \cap \sigma(F_2)$ also satisfies the properties according to the signature of σ. Existence of an AF F with the desired extensions is then a consequence of Theorem 1.

cf: Let $\mathbb{S} = cf(F_1) \cap cf(F_2)$. It is easy to see that $\mathbb{S}$ is downward-closed since $cf(F_1)$ and $cf(F_2)$ are downward-closed. So assume $\mathbb{S}$ is not tight, i.e. there is some $S \in \mathbb{S}$ and $a \in Args_{\mathbb{S}}$ with $S \cup \{a\} \notin \mathbb{S}$ but $\forall s \in S : (a, s) \in Pairs_{\mathbb{S}}$. This means that $S \in cf(F_1)$ and $S \in cf(F_2)$, but there is an $i \in \{1, 2\}$ such that $S \cup \{a\} \notin cf(F_i)$. Since $Pairs_{cf(F_i)} \supseteq Pairs_{\mathbb{S}}$, $cf(F_i)$ is not tight, a contradiction to Theorem 1.

adm: Towards a contradiction, assume $\mathbb{S} = adm(F_1) \cap adm(F_2)$ is not conflict-sensitive, i.e. there are $A, B \in \mathbb{S}$ such that $(A \cup B) \notin \mathbb{S}$, but for all $a, b \in (A \cup B)$, $(a, b) \in Pairs_{\mathbb{S}}$. Then there is some $i \in \{1, 2\}$, such that $A, B \in adm(F_i)$ but $(A \cup B) \notin adm(F_i)$. On the other hand, $Pairs_{adm(F_i)} \supseteq Pairs_{\mathbb{S}}$, hence $\forall a, b \in (A \cup B) : (a, b) \in Pairs_{adm(F_i)}$. Therefore $adm(F_i)$ is not conflict-sensitive, a contradiction to Theorem 1.

nai: Let $\mathbb{S} = nai(F_1) \cap nai(F_2)$ and assume that $dcl(\mathbb{S})$ is not tight, i.e. there is some $S \in dcl(\mathbb{S})$ and $a \in Args_{\mathbb{S}}$ with $S \cup \{a\} \notin dcl(\mathbb{S})$ but $\forall s \in S : (a, s) \in Pairs_{\mathbb{S}}$. This means that there exists an $S' \supseteq S$ with $S' \in nai(F_1)$ and $S' \in nai(F_2)$ and therefore

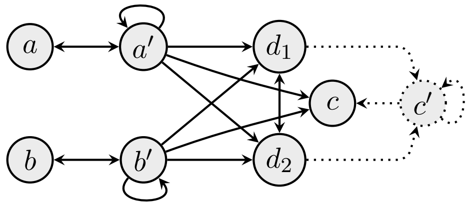


Figure 3.5: AFs F_1 (everything) and F_2 (excluding the dotted part) with $(com(F_1) \cap com(F_2)) \notin \Sigma_{\text{AF}}^{com}$.

$S \in dcl(nai(F_1))$ and $S \in dcl(nai(F_2))$. Moreover, for some $i \in \{1,2\}$ it holds that $\forall T \supseteq (S \cup \{a\}) : T \notin nai(F_i)$ and therefore $(S \cup \{a\}) \notin dcl(nai(F_i))$. Finally, since $Pairs_{nai(F_i)} \supseteq Pairs_{\mathbb{S}}$, $\forall s \in S : (a,s) \in Pairs_{nai(F_i)}$. These observations sum up to $dcl(nai(F_i))$ not being tight, a contradiction to Theorem 1.

The result for *stb*, *stg*, *prf*, and *sem* follows from Theorem 5. □

Interestingly, the complete semantics turns out to be not closed under this form of intersection, as the following example illustrates.

Example 16. Consider the extension-sets $\mathbb{S} = \{\emptyset, \{a\}, \{b\}, \{a,b,c,d_1\}, \{a,b,c,d_2\}\}$, $\mathbb{S}_1 = \mathbb{S} \cup \{\{a,b\}\}$, and $\mathbb{S}_2 = \mathbb{S} \cup \{\{a,b,c\}\}$. $\mathbb{S}_1$ and $\mathbb{S}_2$ are realizable under the complete semantics. Corresponding AFs are depicted in Figure 3.5: $\mathbb{S}_1$ are the complete extensions of the entire AF, and $\mathbb{S}_2$ the ones of the AF without the dotted part. However, $\mathbb{S} = \mathbb{S}_1 \cap \mathbb{S}_2$ is not com-closed (since $\mathbb{C}_{\mathbb{S}}(\{a,b\}) = \{\{a,b,c,d_1\}, \{a,b,c,d_2\}\}$ does not provide a unique completion-set) and therefore, by Proposition 6, no AF F exists such that $com(F) = \mathbb{S}$. ◊

Again, the question for resolution-based grounded semantics remains open. For single-status semantics such as grounded and ideal semantics the question is rather trivial: if $\sigma(F_1) = \sigma(F_2)$ ($\sigma \in \{grd, idl\}$), then obviously $\sigma(F_1) \cap \sigma(F_2) \in \Sigma_{\text{AF}}^{\sigma}$; otherwise, however, $\sigma(F_1) \cap \sigma(F_2) = \emptyset \notin \Sigma_{\text{AF}}^{\sigma}$.

A summary of the closure properties for AF semantics is presented in Table 3.1.

Table 3.1: Closure of AF semantics. $\subseteq$: given AF F, whether any $\mathbb{S} \subseteq \sigma(F)$ is realizable. $\cap$: given AFs F and F', whether $\mathbb{S} = \sigma(F) \cap \sigma(F')$ is realizable. $y^\dagger$ expresses the restriction that $\mathbb{S} \neq \emptyset$.

	cf	*adm*	*nai*	*prf*	*com*	*stb*	*stg*	*sem*	*grd**	*grd*	*idl*
$\subseteq$	n	n	n	$y^\dagger$	n	y	$y^\dagger$	$y^\dagger$	?	$y^\dagger$	$y^\dagger$
$\cap$	$y^\dagger$	$y^\dagger$	$y^\dagger$	$y^\dagger$	n	y	$y^\dagger$	$y^\dagger$	?	$y^\dagger$	$y^\dagger$

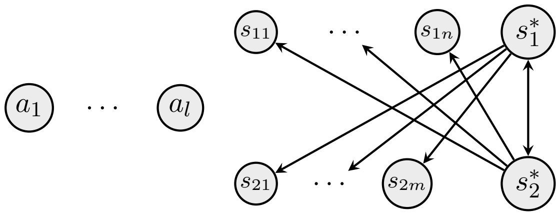


Figure 3.6: Canonical realization of $\{S_1, S_2\}$ (with S_1 and S_2 incomparable) under *grd**. $(S_1 \cap S_2) = \{a_1, \ldots, a_l\}$, $(S_1 \setminus S_2) = \{s_1^*, s_{11}, \ldots, s_{1n}\}$, $(S_2 \setminus S_1) = \{s_2^*, s_{21}, \ldots, s_{2m}\}$.

3.2.6 Quantitative Diversity

Our next results concern limits in expressing multiple extensions. Our first result is positive in the sense that as long as at most two (incomparable) sets of arguments are involved, all semantics satisfying incomparability of extensions are capable to deliver such extensions. Under admissible and complete semantics, any extension-set containing two arbitrary sets of arguments together with $\emptyset$ is realizable.

Proposition 11. *For any extension-set $\mathbb{S}$ with $|\mathbb{S}| \leq 2$,*

1. $\mathbb{S} \in \Sigma^{\sigma}_{AF}$ *for* $\sigma \in \{nai, stb, stg, prf, sem, grd^*\}$ *if* $\mathbb{S}$ *is incomparable and* $\mathbb{S} \neq \emptyset$,
2. $\mathbb{S} \cup \{\emptyset\} \in \Sigma^{\tau}_{AF}$ *for* $\tau \in \{adm, com\}$.

Proof. (1) We begin by showing the claim for naive semantics. By Theorem 1, we need to show that $dcl(\mathbb{S})$ is tight, which trivially holds for $|\mathbb{S}| = 1$. For $\mathbb{S} = \{S_1, S_2\}$, let $S \in dcl(\mathbb{S})$ and $a \in Args_{\mathbb{S}}$ such that $S \cup \{a\} \notin dcl(\mathbb{S})$. W.l.o.g. assume $S \subseteq S_1$. Then, $a \in S_2 \setminus S_1$ and $S \not\subseteq S_2$, meaning that there is some $s \in S \setminus S_2$. Since $|\mathbb{S}| = 2$, $(a, s) \notin Pairs_{\mathbb{S}}$. Hence $dcl(\mathbb{S})$ is tight.

For $\sigma \in \{stb, stg, prf, sem\}$ the claim now follows from $\Sigma^{nai}_{\mathrm{AF}} \subseteq \Sigma^{\sigma}_{\mathrm{AF}}$.

For *grd** we show the result, due to the lack of an exact characterization of the signature, via a concrete realization. To this end consider an incomparable extension-set $\mathbb{S}$ with $|\mathbb{S}| \leq 2$ and $\mathbb{S} \neq \emptyset$. In case $|\mathbb{S}| = 1$, i.e. $\mathbb{S} = \{S\}$, the AF $(S, \emptyset)$ trivially realizes $\mathbb{S}$ under *grd**. In case $|\mathbb{S}| = 2$, i.e. $\mathbb{S} = \{S_1, S_2\}$, note that, since S_1 and S_2 are incomparable, $S_1 \setminus S_2 \neq \emptyset$ and $S_2 \setminus S_1 \neq \emptyset$. Let $s_1^* \in (S_1 \setminus S_2)$ and $s_2^* \in (S_2 \setminus S_1)$ be dedicated arguments contained in only one set. We define the AF

$$F = \{S_1 \cup S_2, \{(s_i^*, s_j) \mid i, j \in \{1, 2\}, i \neq j, s_j \in (S_j \setminus S_i)\}).$$

F is depicted in Figure 3.6. Now we observe that F has exactly two resolutions, namely $F_1 = (A_F, R_F \setminus \{(s_2^*, s_1^*)\})$ and $F_2 = (A_F, R_F \setminus \{(s_1^*, s_2^*)\})$. Consider F_1. Arguments $S_1 \cap S_2$ as well as argument s_1^* are unattacked, hence contained in $grd(F_1)$. Moreover s_1^* attacks s_2^*, defending all $s \in ((S_1 \setminus S_2) \setminus \{s_1^*\})$. Finally s_1^* attacks all $s_2 \in ((S_2 \setminus S_1) \setminus \{s_2^*\})$.

We conclude that $grd(F_1) = S_1$. Symmetrically, we get that $grd(F_2) = S_2$. Hence, $grd^*(F) = \{S_1, S_2\}$.

(2) We show that $\mathbb{T} = \mathbb{S} \cup \{\emptyset\}$ is conflict-sensitive as long as $|\mathbb{S}| \leq 2$. This trivially holds for $|\mathbb{S}| \leq 1$, since then for all $A, B \in \mathbb{T}$, $A \cup B \in \mathbb{T}$. So let $\mathbb{S} = \{S_1, S_2\}$ with $S_1 \neq S_2$ and w.l.o.g. $S_1, S_2 \neq \emptyset$. If $S_1 \subset S_2$ or $S_2 \subset S_1$ then $S_1 \cup S_2 \in \mathbb{T}$, hence $\mathbb{T}$ is conflict-sensitive. On the other hand, if S_1 and S_2 are incomparable there is an $a \in S_1$ and a $b \in S_2$ such that $(a, b) \notin Pairs_{\mathbb{T}}$, again showing conflict-sensitivity. Therefore $\mathbb{T} \in \Sigma^{adm}_{\mathrm{AF}}$. The result for *com* follows from the fact that $\Sigma^{adm}_{\mathrm{AF}} \subset \Sigma^{com}_{\mathrm{AF}}$ (cf. Theorem 3). □

The claim does not hold for conflict-free sets, as the only extension-sets of size 2 realizable under *cf* are of the form $\{\emptyset, \{a\}\}$, where a is an arbitrary argument.

For extension-sets of size three, the statement as in Proposition 11 does not hold for any of the semantics. This is witnessed by the extension-set $\{\{a, b\}, \{a, c\}, \{b, c\}\}$, as shown in the following example.

Example 17. Consider the extension-set $\mathbb{S} = \{\{a, b\}, \{a, c\}, \{b, c\}\}$ and a semantics σ, and assume that the AF F realizes $\mathbb{S}$ under σ. For any semantics which requires its extensions to be conflict-free (which holds for all semantics considered in this work) F must not have attacks among a, b, and c. Hence $\{a, b, c\} \in cf(F)$. Moreover, each $S \in \mathbb{S}$ defends itself in F, therefore also $\{a, b, c\} \in adm(F)$. Hence $\mathbb{S} \cup \{\emptyset\} \notin \Sigma^{adm}_{\mathrm{AF}}$. Moreover $\mathbb{S} \notin \Sigma^{prf}_{\mathrm{AF}}$ and, by Theorem 2 and Proposition 8, $\mathbb{S} \notin \Sigma^{\sigma}_{\mathrm{AF}}$ for $\sigma \in \{nai, stb, stg, sem, grd^*\}$. Finally, there must be some $T \supseteq \{a, b, c\}$ with $T \in com(F)$, hence also $\mathbb{S} \cup \{\emptyset\} \notin \Sigma^{com}_{\mathrm{AF}}$.

In the remainder of this section, we address the question how many extensions can maximally be achieved by an AF under a semantics σ? Such insights ease checking $\mathbb{S} \in \Sigma^{\sigma}_{\mathrm{AF}}$ whenever the cardinality of $\mathbb{S}$ exceeds a certain number.

Research in this direction has been initiated by Baumann and Strass [26]. They proposed a function giving the maximal number of stable extensions an AF with n arguments can have. In accordance to realizability, we are interested in the number of extensions as a function of a fixed amount of arguments occurring in any extension.

While our results on signatures give insights about the extent of structural diversity a semantics can express, the maximal number of extensions gives, to some degree, an answer to how much quantitative disagreement a semantics can express.

Recall that for any extension-set $\mathbb{S}$, we denote the number of extensions in $\mathbb{S}$ as $|\mathbb{S}|$, and the number of arguments occurring in any extension of $\mathbb{S}$ as $\|\mathbb{S}\|$ (cf. Definition 8).

Definition 42. Given a semantics σ, we define the *diversity function*

$$\Delta_{\sigma}(n) = \max_{F \in AF_{\mathfrak{A}}, \|\sigma(F)\| = n} |\sigma(F)|.$$

Note that we do not restrict the number of arguments in the AF F, but only require that at most n arguments occur in some σ-extension of F.

Theorem 7. *For $\sigma \in \{cf, adm, com\}$, it holds that $\Delta_\sigma(n) = 2^n$*

Proof. Given an extension-set $\mathbb{S}$ with $\|\mathbb{S}\| = n$, the AF $(Args_{\mathbb{S}}, \emptyset)$ has 2^n conflict-free and admissible sets and the AF $(\{a, a' \mid a \in Args_{\mathbb{S}}\}, \{(a, a'), (a', a), (a', a') \mid a \in Args_{\mathbb{S}}\})$ has 2^n complete extensions, namely all subsets of $\mathbb{S}$ in both cases. □

For the other semantics, which are the ones requiring incomparability of extensions, we first give a technical lemma. The intuition of this lemma is that we can make any AF symmetric without losing any preferred extensions. We might get additional ones, but only with arguments already contained in other preferred extensions.

Lemma 8. *For any AF F, there is a symmetric AF F^{sym} with $\|prf(F)\| = \|prf(F^{sym})\|$ and $|prf(F)| \leq |prf(F^{sym})|$.*

Proof. In order to get the symmetric AF F^{sym} we transform $F = (A, R)$ by (1) removing all arguments $a \notin Args_{prf(F)}$ together with adjacent attacks, and (2) adding (b, a) to R if $(a, b) \in R$.

$$F^{sym} = (Args_{prf(F)}, \{(a, b), (b, a) \mid (a, b) \in R_F\} \cap (Args_{prf(F)} \times Args_{prf(F)}))$$

Obviously conflict-freeness and defense is preserved, i.e. any set admissible in F is admissible in F^{sym}. Moreover, as only attacks to and from arguments not occurring in any preferred extension of F are removed, any conflict between two preferred extensions $E_1, E_2 \in prf(F)$ survives the translation, therefore there must be two $E_1', E_2' \in prf(F^{sym})$ with $E_1 \subseteq E_1'$ and $E_2 \subseteq E_2'$. Hence $|prf(F)| \leq |prf(F^{sym})|$. As $Args_{prf(F^{sym})}$ coincides with the arguments of F^{sym} (by symmetry of F^{sym}), it follows that $\|prf(F)\| = \|prf(F^{sym})\|$. □

Baumann and Strass [26] provide a function mapping numbers of arguments n to the maximal number of stable extensions of an AF with n arguments can have. Their main result is as follows.

Proposition 12 ([26])**.** *For any natural number n, it holds that*

$$\max_{F \in AF_{\mathfrak{A}}, |A_F| = n} |stb(F)| = \Lambda(n)$$

with

$$\Lambda(n) = \begin{cases} 1, & \text{if } n = 1 \\ 3^s, & \text{if } n \geq 2 \land n = 3s \\ 4 \cdot 3^{s-1}, & \text{if } n \geq 2 \land n = 3s + 1 \\ 2 \cdot 3^s, & \text{if } n \geq 2 \land n = 3s + 2. \end{cases}$$

In contrast to Δ_σ, Baumann and Strass are interested in the maximal number of stable extensions which can be achieved by an AF with n arguments, no matter how many of these arguments occur in some extension. The AF giving the maximum number of extension is basically composed of connected components of size 3 (or 2) with each component being a clique (see [26] for details). The following result shows that the values of the function Λ carry over to the function Δ_σ for naive, stable, stage, preferred, and semi-stable semantics. Informally, this means that additional arguments do not allow for a greater maximal number of extensions.

Theorem 8. *For $\sigma \in \{nai, stb, stg, prf, sem\}$ and any natural number n, it holds that*

$$\Delta_\sigma(n) = \Lambda(n).$$

Proof. Consider a semantics $\sigma \in \{nai, stb, stg, prf, sem\}$, a natural number n and an AF F with $\|\sigma(F)\| = n$. Moreover assume that F has maximal diversity, i.e. $|\sigma(F)| = \Delta_\sigma(n)$. Since $\Sigma^\sigma_{\mathrm{AF}} \subseteq \Sigma^{prf}_{\mathrm{AF}}$, we can find an AF F' with $prf(F') = \sigma(F)$, therefore $|\sigma(F)| = |prf(F')|$ and $\|\sigma(F)\| = \|prf(F')\|$. Moreover, by Lemma 8, we can find a symmetric AF $F^{sym} = (A^{sym}, R^{sym})$ such that $\|prf(F^{sym})\| = \|prf(F')\|$ and $|prf(F^{sym})| \geq |prf(F')|$. In this symmetric AF it holds that $prf(F^{sym}) = \sigma(F^{sym}) = stb(F^{sym})$. Moreover, each argument occurs in at least one σ-extension, i.e. $\|\sigma(F^{sym})\| = |A^{sym}|$. Therefore it follows by Proposition 12 that $|\sigma(F^{sym})| \leq \Lambda(n)$. Since we assumed F having maximal diversity, it follows that $\Delta_\sigma(n) \leq \Lambda(n)$.

Finally consider the fact that for all AFs $F = (A, R)$ with $|A| = n$ having $|stb(F)| = \Lambda(n)$ according to [26], it holds that each argument occurs in at least one stable extension, i.e. $\|stb(F)\| = |A|$. Moreover, F is symmetric, hence $\sigma(F) = stb(F)$. Therefore F is an AF with $\|\sigma(F)\| = n$ and $|\sigma(F)| = \Lambda(n)$, hence $\Delta_\sigma(n) = \Lambda(n)$. □

The diversity function for resolution-based grounded semantics gives strictly smaller values. This can already seen in the case of $n = 3$ (say $\{a, b, c\}$), where, due to incomparability of extension-sets under grd^*, the only candidates of size $\Delta_\sigma(3) = 3$ are $\{\{a, b\}, \{a, c\}, \{b, c\}\}$ and $\{\{a\}, \{b\}, \{c\}\}$. The former candidate is not conflict-sensitive and therefore not even contained in $\Sigma^{prf}_{\mathrm{AF}}$, the latter is not realizable under grd^* due to Proposition 9. On the other hand we can give the following lower bound:

$$\Delta_{grd^*}(n) \geq 2^{\lfloor \frac{n}{2} \rfloor}$$

This is achieved by building $\lfloor \frac{n}{2} \rfloor$ strongly connected components of size 2. Then each resolution gives rise to a particular extension under grd^*.

3.2.7 Complexity

In this subsection, we consider the computational complexity of checking realizability, i.e. given an extension-set $\mathbb{S}$ and a semantics σ, whether there is an AF F with $\sigma(F) = \mathbb{S}$. This is equivalent with checking membership in the signature for semantics σ, so whether $\mathbb{S} \in \Sigma^\sigma_{\mathrm{AF}}$ holds. For most of the semantics, it is not hard to see that this can be done in polynomial time in the size of $\mathbb{S}$. The only exception is the naive semantics, since the characterization in Theorem 1 makes use of $dcl(\mathbb{S})$ which is not polynomially bounded in the size of $\mathbb{S}$.

We provide an alternative characterization based on the ternary majority operator *maj3*:

Definition 43. Given three sets $S_1, S_2, S_3 \subseteq \mathfrak{A}$, the majority of these sets is defined as $maj_3(S_1, S_2, S_3) = (S_1 \cap S_2) \cup (S_2 \cap S_3) \cup (S_1 \cap S_3)$.

In other words, this means that $s \in maj_3(S_1, S_2, S_3)$ if and only if s appears in at least two of the sets. Using *maj3* we can show an alternative characterization of $\Sigma^{nai}_{\mathrm{AF}}$.

Proposition 13. *For every incomparable extension-set* $\mathbb{S} \subseteq 2^{\mathfrak{A}}$ *it holds that* $dcl(\mathbb{S})$ *is tight iff for all* $S_1, S_2, S_3 \in \mathbb{S}$ *there is an* $S \in \mathbb{S}$*, such that* $maj_3(S_1, S_2, S_3) \subseteq S$.

Proof. Let $\mathbb{S} \subseteq 2^{\mathfrak{A}}$ be an incomparable extension-set.

First suppose that for all $S_1, S_2, S_3 \in \mathbb{S}$ there is some $S \in \mathbb{S}$ such that $maj_3(S_1, S_2, S_3) \subseteq S$. Towards a contradiction assume that the downward-closure of $\mathbb{S}$ is not tight, i.e. there exist $S' \in dcl(\mathbb{S})$ and $a \in Args_{\mathbb{S}} = Args_{dcl(\mathbb{S})}$, such that $(S' \cup \{a\}) \notin dcl(\mathbb{S})$ and for all $s \in S'$, $(a, s) \in Pairs_{\mathbb{S}} = Pairs_{dcl(\mathbb{S})}$. Assume $|S'| = 1$, i.e. $S' = \{s\}$. As $(a, s) \in Pairs_{\mathbb{S}}$ by assumption, there is a $T \in \mathbb{S}$ with $\{a, s\} \subseteq T$, a contradiction to $S' \cup \{a\} \notin dcl(\mathbb{S})$. Hence $|S'| > 1$, i.e. $S' = \{s_1, \dots, s_n\}$ with $n > 1$. By assumption, $\{s_1, \dots, s_n, a\} \notin \mathbb{S}$, but $(a, s_i) \in Pairs_{\mathbb{S}}$ for each $s_i \in S'$. Hence, for each $s_i \in S'$ there is some $S_i \in \mathbb{S}$ with $\{a, s_i\} \subseteq S_i$ for $i = 1 \dots n$. Moreover $maj_3(S_i, S_j, S') \supseteq \{s_i, s_j, a\}$ for each $i, j \in \{1, \dots, n\}$ since $s_i \in S' \cap S_i$, $s_j \in S' \cap S_j$, and $a \in S_i \cap S_j$. Therefore there is some $S_{ij} \in \mathbb{S}$ with $S_{ij} \supseteq \{s_i, s_j, a\}$. Now for some $k \in \{1, \dots, n\}$, we get $S_{ijk} = maj_3(S_{ij}, S_k, S') \supseteq \{s_i, s_j, s_k, a\}$ and $S_{ijk} \in \mathbb{S}$. Following this procedure for all $1 \dots n$ yields a $T \in \mathbb{S}$ with $T \supseteq \{s_1, \dots, s_n, a\}$, a contradiction to $S' \cup \{a\} \notin dcl(\mathbb{S})$.

To show the only-if-direction consider some extension-set $\mathbb{S}$ where $dcl(\mathbb{S})$ is tight and assume, towards a contradiction, sets $S_1, S_2, S_3 \in \mathbb{S}$ such that $maj_3(S_1, S_2, S_3) \not\subseteq S$, for all $S \in \mathbb{S}$. Now, consider the $\subseteq$-maximal $S' \in dcl(\mathbb{S})$ with $S' \subset maj_3(S_1, S_2, S_3)$. It holds that $\exists a \in maj_3(S_1, S_2, S_3) \setminus S'$ such that $S' \cup \{a\} \notin dcl(\mathbb{S})$. Thus for each $s \in S'$ there is some $S_i \in \{S_1, S_2, S_3\}$ with $\{s, a\} \subseteq S_i$. Hence, $(s, a) \in Pairs_{\mathbb{S}}$ for each $s \in S'$, which is, together with the facts that $S' \in dcl(\mathbb{S})$ and $S' \cup \{a\} \notin dcl(\mathbb{S})$, a witness that $dcl(\mathbb{S})$ is not tight, and therefore a contradiction to the assumption. □

Example 18. Consider the extension-set $\mathbb{T} = \{\{a, b', c'\}, \{a', b, c'\}, \{a', b', c\}\}$. We have already argued in Example 15 that $dcl(\mathbb{T})$ is not tight. Now we observe that $maj_3(\{a, b', c'\}, \{a', b, c'\}, \{a', b', c\}) = \{a', b', c'\}$. It turns out that there is no $T \in \mathbb{T}$ such that $\{a', b', c'\} \subseteq T$, confirming Proposition 13. ◊

We obtain the following theorem.

Theorem 9. *For semantics* $\sigma \in \{cf, nai, stb, stg, adm, prf, sem\}$*, given an extension-set* $\mathbb{S} \subseteq 2^{\mathfrak{A}}$*, testing* $\mathbb{S} \in \Sigma^{\sigma}_{AF}$ *is in polynomial time.*

Proof. Given $\mathbb{S} \subseteq 2^{\mathfrak{A}}$, checking $\mathbb{S} \neq \emptyset$ and $\emptyset \in \mathbb{S}$ is clearly in polynomial time. Incomparability of $\mathbb{S}$ can be checked via a double loop over $\mathbb{S}$, in each step checking incomparability of two elements of $\mathbb{S}$. To check whether $\mathbb{S}$ is tight we loop over all $S \in \mathbb{S}$ and in each such loop, we loop over all $a \in \mathit{Args}_{\mathbb{S}} \setminus S$ to find some $(a,s) \notin \mathit{Pairs}_{\mathbb{S}}$ if $S \cup \{a\} \notin \mathbb{S}$ in a final loop over all $s \in S$. Similarly, to check whether $\mathbb{S}$ is conflict-sensitive we loop over all $A, B \in \mathbb{S}$ and all $a, b \in A \cup B$ to test if, given $A \cup B \notin \mathbb{S}$ we find some $(a,b) \notin \mathit{Pairs}_{\mathbb{S}}$. Finally, checking the majority criterion from Proposition 13 can be done in polynomial time by looping over all triples (S_1, S_2, S_3) stemming from $\mathbb{S}$, and testing whether $\mathit{maj3}(S_1, S_2, S_3)$ is contained in some $S \in \mathbb{S}$. Since these properties are, according to Theorem 1, the ones to test when checking $\mathbb{S} \in \Sigma^{\sigma}_{\mathrm{AF}}$, the result follows. □

Of course one difficulty with Theorem 9 is that one may be concerned with deciding realizability of collections of sets, $\mathbb{S}$, with such collections having size superpolynomial in $|\mathit{Args}_{\mathbb{S}}|$. In such cases it would be more realistic to encode $\mathbb{S}$ in a more compact form. We observe that there are a number of ways in which such "compact encodings" may be treated.

1. As the models of a given propositional formula.

2. As the extensions of an AF under another argumentation semantics.

While in the first case the problem gets at most DP-hard for the semantics under consideration, one can show hardness for the second level of the polynomial hierarchy for certain combinations of semantics in the second case. Detailed results on these issues can be found in [102].

3.3 Compact Realizability

In this section we deal with realizability in a subclass of AFs, namely *compact argumentation frameworks* (CAFs). The results covered here were first presented in [30] and later extended and round up in [31].

We will first define compact argumentation frameworks and relate the classes of AFs induced by the respective semantics to each other. Then we will study realizability restricted to CAFs. First, we will show that, for most semantics, additional arguments (leading to non-compact AFs) are necessary to obtain the full expressiveness of the semantics. Then, we will relate the signatures in CAF of the semantics under consideration to each other, which will give a picture which is significantly different compared to general signatures.

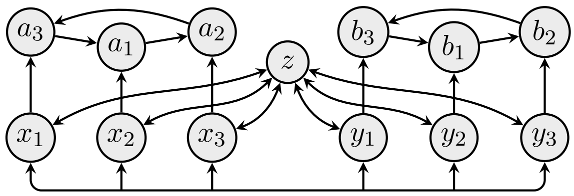


Figure 3.7: AF discussed in Example 19, which is *prf*-compact but neither *sem*-compact nor *stg*-compact.

3.3.1 Compact Argumentation Frameworks

The idea behind compact argumentation frameworks is that each argument appearing in the framework is contained in at least one extension. In other words, they are characterized by the absence of rejected arguments.

Definition 44. Let $F \in AF_{\mathfrak{A}}$ and σ be a semantics. An argument $a \in A_F$ is a *rejected argument* in F under σ if $a \notin \mathit{Args}_{\sigma(F)}$.[2]

It is clear that the question whether a given AF is compact can only be answered with respect to a given semantics. Therefore each semantics σ gives rise to a specific class of argumentation frameworks, namely the ones being compact for σ.

Definition 45. Let σ be a semantics. An AF F is called *compact for* σ (or σ-compact) if $\mathit{Args}_{\sigma(F)} = A_F$. The set of all *compact argumentation frameworks* for σ is denoted by CAF_σ.

The main feature of compact argumentation frameworks is the absence of rejected arguments under a given semantics. The following example illustrates this idea.

Example 19. Let us consider the AF F depicted in Figure 3.7.[3] The preferred extensions of F are $\mathit{prf}(F) = \{\{z\}, \{x_1, a_1\}, \{x_2, a_2\}, \{x_3, a_3\}, \{y_1, b_1\}, \{y_2, b_2\}, \{y_3, b_3\}\}$, meaning that F is *prf*-compact ($F \in \mathrm{CAF}_{\mathit{prf}}$) since each argument occurs in at least one preferred extension and therefore F has no rejected arguments under preferred semantics. On the other hand observe that $\mathit{stb}(F) = \emptyset$, $\mathit{sem}(F) = \mathit{prf}(F) \setminus \{\{z\}\}$, and $\mathit{stg}(F) = \{\{x_i, a_i, b_j\}, \{y_i, b_i, a_j\} \mid 1 \leq i, j \leq 3\}$, i.e. z is not contained in any stable, semi-stable or stage extension. Therefore F is not compact for any semantics among stable, semi-stable and stage, that is $F \notin \mathrm{CAF}_{\mathit{stb}}$, $F \notin \mathrm{CAF}_{\mathit{sem}}$, and $F \notin \mathrm{CAF}_{\mathit{stg}}$. $\Diamond$

As already mentioned before and indicated by Example 19, the contents of CAF_σ differ with respect to the semantics σ. Concerning relations between the classes of compact AFs we start with an easy observation.

[2]Recall that this means that for all $S \in \sigma(F)$ it holds that $a \notin S$.

[3]The construct in the lower part of the figure represents symmetric attacks between each pair of distinct arguments. We will make use of this style in illustrations throughout the paper.

$$
\begin{array}{ccccccc}
 & & \mathrm{CAF}_{grd^*} & & & & \\
 & & & \subseteq & & & \\
 & & \mathrm{CAF}_{sem} & \subset & \mathrm{CAF}_{prf} = \mathrm{CAF}_{adm} = \mathrm{CAF}_{com} & & \\
 & \subset & & & & \subset & \\
\mathrm{CAF}_{stb} & & & & & & \mathrm{CAF}_{nai} = \mathrm{CAF}_{cf} \\
 & & \subset & & & \subset & \\
 & & & & \mathrm{CAF}_{stg} & &
\end{array}
$$

Figure 3.8: Relations between classes of compact AFs (cf. Theorem 10). Unconnected pairs are incomparable.

Lemma 9. *For any two semantics σ and τ such that for each AF F and every $S \in \sigma(F)$ there is some $S' \in \tau(F)$ with $S \subseteq S'$, it holds that $CAF_\sigma \subseteq CAF_\tau$.*

Proof. Suppose $F \in \mathrm{CAF}_\sigma$. By definition, $Args_{\sigma(F)} = A_F$. Now if for each $S \in \sigma(F)$ there is some $S' \in \tau(F)$ with $S \subseteq S'$, we have $Args_{\sigma(F)} \subseteq Args_{\tau(F)}$. Since $Args_{\tau(F)} \subseteq A_F$ by definition, $Args_{\tau(F)} = A_F$ follows. Hence, $F \in \mathrm{CAF}_\tau$. $\square$

Note that the case where $\sigma(F) \subseteq \tau(F)$ holds for each AF F is a special case of the premise of Lemma 9. The next result provides a full picture of the relations between classes of compact AFs for the semantics we consider (see also Figure 3.8).

Theorem 10. *The following relations hold:*

1. $CAF_{stb} \subset CAF_\sigma \subset CAF_{nai}$ *for* $\sigma \in \{prf, sem, stg\}$*;*
2. $CAF_{sem} \subset CAF_{prf}$*;*
3. $CAF_{stg} \not\subseteq CAF_\tau$ *and* $CAF_\tau \not\subseteq CAF_{stg}$ *for* $\tau \in \{prf, sem\}$*;*
4. $CAF_{prf} = CAF_{adm} = CAF_{com}$*;*
5. $CAF_{cf} = CAF_{nai}$*;*
6. $CAF_{grd^*} \subset CAF_{prf}$*;*
7. $CAF_{grd^*} \not\subseteq CAF_\theta$ *and* $CAF_\theta \not\subseteq CAF_{grd^*}$ *for* $\theta \in \{stb, stg, sem\}$*.*

Proof. (1) Let $\sigma \in \{prf, sem, stg\}$. The $\subseteq$-relations are due to Lemma 9 together with following facts: (a) in any AF F, $stb(F) \subseteq \sigma(F)$; (b) each σ-extension E of an AF F is conflict-free in F, thus there exists a naive extension E' of F with $E \subseteq E'$.

$\mathrm{CAF}_\sigma \subset \mathrm{CAF}_{nai}$: The AF $(\{a, b\}, \{(a, b)\})$ is compact for naive semantics but not for σ.

$\mathrm{CAF}_{stb} \subset \mathrm{CAF}_\sigma$: Consider the AF F depicted in Figure 3.9. We have $prf(F) = sem(F) = \{\{x_1, a_1\}, \{x_2, a_2\}, \{x_3, a_3\}, \{y_1, b_1\}, \{y_2, b_2\}, \{y_3, b_3\}\}$, and each of these extensions can be extended to a stage extension (the former three by adding one of the arguments

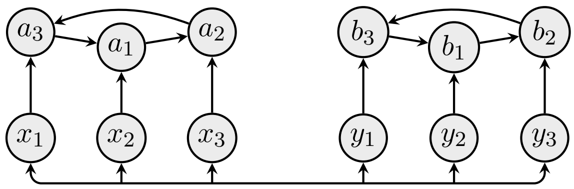


Figure 3.9: AF F' contained in CAF_{prf}, CAF_{sem}, and CAF_{stg} but not in CAF_{stb}.

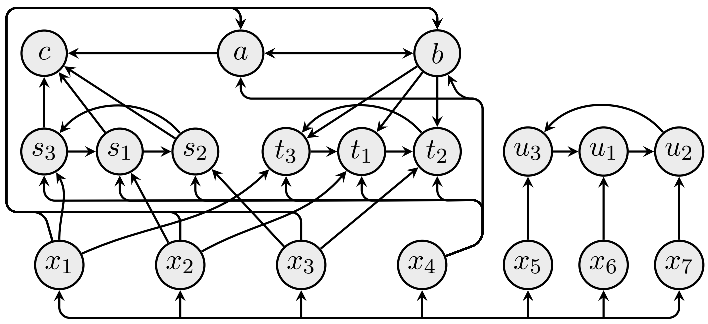


Figure 3.10: AF F'' contained in CAF_{sem} but not in CAF_{stg}.

b_1, b_2, b_3; the latter three by adding one of the arguments a_1, a_2, a_3), but $stb(F) = \emptyset$. Thus $A_F = Args_{\sigma(F)} \neq Args_{stb(F)} = \emptyset$, meaning that $F \in \mathrm{CAF}_\sigma$ but $F \notin \mathrm{CAF}_{stb}$.

(2) $\mathrm{CAF}_{sem} \subseteq \mathrm{CAF}_{prf}$ is by the fact that, in any AF F, $sem(F) \subseteq prf(F)$ (cf. Lemma 9). Properness of the relation is by the AF in Figure 3.7, which is (as discussed in Example 19) *prf*-compact but not *sem*-compact.

(3) First we show $\mathrm{CAF}_{stg} \not\subseteq \mathrm{CAF}_\tau$ for $\tau \in \{prf, sem\}$. To this end, consider the simple AF $F' = (\{a,b,c\}, \{(a,b),(b,c),(c,a)\})$. We have $stg(F') = \{\{a\},\{b\},\{c\}\}$, thus $F' \in \mathrm{CAF}_{stg}$. On the other hand, $sem(F') = prf(F') = \{\emptyset\}$, thus $F' \notin \mathrm{CAF}_\tau$.

$\mathrm{CAF}_{prf} \not\subseteq \mathrm{CAF}_{stg}$ follows by the observations in Example 19.

$\mathrm{CAF}_{sem} \not\subseteq \mathrm{CAF}_{stg}$: Consider the AF F'' in Figure 3.10. One can check that this AF is *sem*-compact, but not *stg*-compact. In fact, one can verify that

$$\begin{aligned} stg(F'') = &\{\{x_i, t_i, s_i, u_j\} \mid i,j \in \{1,2,3\}\} \cup \\ &\{\{x_4, c, u_i\} \mid i \in \{1,2,3\}\} \cup \\ &\{\{x_{i+4}, u_i, s_j, b\} \mid i,j \in \{1,2,3\}\}, \end{aligned}$$

while

$$\begin{aligned} sem(F'') = &\{\{x_i, s_i, t_i\} \mid i \in \{1,2,3\}\} \cup \\ &\{\{x_4, c\}\} \cup \\ &\{\{x_{i+4}, u_i, a\} \mid i \in \{1,2,3\}\} \cup \\ &\{\{x_{i+4}, u_i, b\} \mid i \in \{1,2,3\}\} \end{aligned}$$

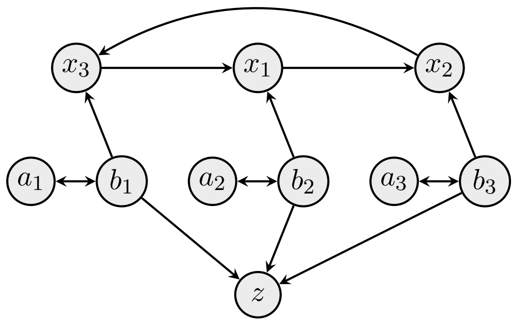


Figure 3.11: AF F contained in CAF_{grd^*}, but not in CAF_{stb}, CAF_{stg}, and CAF_{sem}.

It can be seen that argument a does not occur in any stage extension. Although $\{a, u_1, x_5\}, \{a, u_2, x_6\}, \{a, u_3, x_7\} \in sem(F'')$, the range of any conflict-free set containing a is a proper subset of the range of every stage extension of F''. Hence $\mathrm{CAF}_{sem} \not\subseteq \mathrm{CAF}_{stg}$.

(4) Since preferred extensions of any given AF F are exactly the $\subseteq$-maximal admissible and complete extensions of F it holds that $Args_{prf(F)} = Args_{adm(F)} = Args_{com(F)}$. Therefore $F \in \mathrm{CAF}_{prf}$ iff $F \in \mathrm{CAF}_{adm}$ iff $F \in \mathrm{CAF}_{com}$.

(5) Since naive extensions of any given AF F are exactly the $\subseteq$-maximal conflict-free sets of F it holds that $Args_{nai(F)} = Args_{cf(F)}$. Therefore $F \in \mathrm{CAF}_{nai}$ iff $F \in \mathrm{CAF}_{cf}$.

(6) $\mathrm{CAF}_{grd^*} \subseteq \mathrm{CAF}_{prf}$ is by Lemma 9, the fact that, in any AF F, $grd^*(F) \subseteq com(F)$, and $\mathrm{CAF}_{prf} = \mathrm{CAF}_{com}$ (cf. (4)). Properness of the relation is witnessed by the AF given by the symmetric triangular graph $F = (\{a, b, c\}, \{(a, b), (b, a), (b, c), (c, b), (a, c), (c, a)\})$, which has $prf(F) = \{\{a\}, \{b\}, \{c\}\}$, but $grd^*(F) = \{\emptyset\}$.

(7) Let $\theta \in \{stb, stg, sem\}$. $\mathrm{CAF}_\theta \not\subseteq \mathrm{CAF}_{grd^*}$ follows from the same AF F as in (6), which has $\theta(F) = \{\{a\}, \{b\}, \{c\}\}$, but $grd^*(F) = \{\emptyset\}$. $\mathrm{CAF}_{grd^*} \not\subseteq \mathrm{CAF}_\theta$, on the other hand, is witnessed by the AF F depicted in Figure 3.11. We observe that

$$\begin{aligned} grd^*(F) = \{&\{a_1, a_2, a_3, z\}, \{a_1, a_2, b_3, x_3\}, \{a_1, b_2, a_3, x_2\}, \{b_1, a_2, a_3, x_1\},\\ &\{a_1, b_2, b_3, x_3\}, \{b_1, a_2, b_3, x_1\}, \{b_1, b_2, a_3, x_2\}, \{b_1, b_2, b_3\}\}. \end{aligned}$$

However, $\theta(F) = grd^*(F) \setminus \{\{a_1, a_2, a_3, z\}\}$, since the range of $\{a_1, a_2, a_3, z\}$ misses arguments x_1, x_2, and x_3. Therefore z is rejected w.r.t. θ, hence $F \in \mathrm{CAF}_{grd^*}$, but $F \notin \mathrm{CAF}_\theta$. □

Finally a note on the relation between compact AFs and two well-known syntactic classes of AFs (cf. Definition 22). First observe that any symmetric and self-attack-free (i.e. having an irreflexive attack relation) AF is contained in CAF_{stb}, as already observed in [73, Proposition 6], and therefore also in each CAF_σ for all semantics σ under consideration. But already CAF_{stb} contains strictly more AFs than the class of symmetric and self-attack-free AFs, which is, for instance, indicated by the AF

$(\{a,b,c,d\}, \{(a,b),(b,c),(c,d),(d,a)\})$, i.e. the directed cycle of four arguments, which is clearly not symmetric but compact for the stable semantics. On the other hand, the class of self-attack-free AFs is just CAF_{cf} (and CAF_{nai}), since every argument which is not self-attacking is always contained in a conflict-free (and naive) extension, while a self-attacking argument never is.

3.3.2 Compact Signatures

We now turn to the issue of realizing extension-sets by compact AFs, that is, without the use of rejected arguments. We will see that for most semantics the full expressiveness indeed requires the use of rejected arguments. Moreover, we will show that the relations between semantics in terms of expressiveness are dramatically changed when requiring compactness of realizing AFs.

Definition 46. Let σ be a semantics. An extension-set $\mathbb{S}$ is *compactly realizable* under σ if there is a compact AF $F \in \mathrm{CAF}_\sigma$ with $\sigma(F) = \mathbb{S}$. The *compact signature (c-signature)* $\Sigma^\sigma_{\mathrm{CAF}}$ of σ consists of all extension-sets that are compactly realizable under σ:

$$\Sigma^\sigma_{\mathrm{CAF}} = \{\sigma(F) \mid F \in \mathrm{CAF}_\sigma\}.$$

It is clear that $\Sigma^\sigma_{\mathrm{CAF}} \subseteq \Sigma^\sigma_{\mathrm{AF}}$ holds for any semantics. We will see in the following theorem, which summarizes and extends results from [146], that if an extension-set is realizable under the naive or conflict-free semantics, then it is also compactly realizable under that semantics, i.e. compact and general signatures coincide. This does not hold for the other semantics, where we get a $\subset$-relation between compact and general signatures.

Theorem 11. *It holds that*

1. $\Sigma^\sigma_{CAF} = \Sigma^\sigma_{AF}$ *for* $\sigma \in \{cf, nai\}$, *and*
2. $\Sigma^\tau_{CAF} \subset \Sigma^\tau_{AF}$ *for* $\tau \in \{stb, stg, sem, prf, adm, com, grd^*\}$.

Proof. (1) Consider some $\mathbb{S} \subseteq 2^{\mathfrak{A}}$ such that $\mathbb{S} \in \Sigma^{nai}_{\mathrm{AF}}$ (resp. $\mathbb{S} \in \Sigma^{cf}_{\mathrm{AF}}$) with F being the AF realizing $\mathbb{S}$ under naive (resp. conflict-free) semantics. It holds that an argument is contained in $Args_{\mathbb{S}}$ iff it is not self-attacking. Moreover removing any self-attacking argument together with its associated attacks has no effect on the naive (resp. conflict-free) extensions. Hence the AF F' obtained from removing all self-attacking arguments together with their associated attacks has $nai(F') = \mathbb{S}$ and $F' \in \mathrm{CAF}_{nai}$ (resp. $cf(F') = \mathbb{S}$ and $F' \in \mathrm{CAF}_{cf}$), therefore $\Sigma^{nai}_{\mathrm{CAF}} = \Sigma^{nai}_{\mathrm{AF}}$ (resp. $\Sigma^{cf}_{\mathrm{CAF}} = \Sigma^{cf}_{\mathrm{AF}}$).

(2) By definition we have $\Sigma^\tau_{\mathrm{CAF}} \subseteq \Sigma^\tau_{\mathrm{AF}}$. It remains to show that $\Sigma^\tau_{\mathrm{CAF}} \neq \Sigma^\tau_{\mathrm{AF}}$ for $\tau \in \{stb, stg, sem, prf, adm, com, grd^*\}$.

stb, *stg*: Let $\tau \in \{stb, stg\}$ and consider the extension-set

$$\mathbb{S} = \{\{a,b,c\},\{a,b,c'\},\{a,b',c\},\{a',b,c\},\{a,b',c'\},\{a',b,c'\},\{a',b',c\}\}.$$

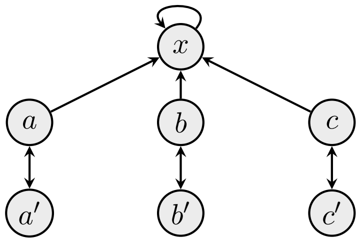


Figure 3.12: AF F such that $\tau(F)$ cannot be compactly realized under $\tau \in \{stb, stg\}$.

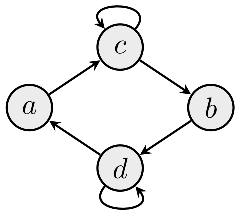


Figure 3.13: AF F such that $\tau(F)$ cannot be compactly realized under $\tau \in \{adm, com\}$.

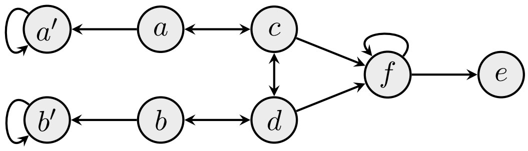


Figure 3.14: AF F such that $\tau(F)$ cannot be compactly realized under $\tau \in \{prf, sem, grd^*\}$.

$\mathbb{S}$ is realized under τ by the AF depicted in Figure 3.12. Assume there is an AF $F = (Args_{\mathbb{S}}, R)$ compactly realizing $\mathbb{S}$ under τ. Inspecting $Pairs_{\mathbb{S}}$ we infer that $R \subseteq \{(a,a'),(a',a),(b,b'),(b',b),(c,c'),(c',c)\}$. Note that, for any remaining choice of R, $stb(F) = stg(F)$. Now for $\{a,b,c\} \in stb(F)$ we need $(a,a'),(b,b'),(c,c') \in R$. On the other hand, for $\{a',b,c\},\{a,b',c\},\{a,b,c'\} \in stb(F)$ we need $(a',a),(b',b),(c',c) \in R$. But then also $\{a',b',c'\} \in stb(F)$. Hence $\mathbb{S} \notin \Sigma^{stb}_{\mathrm{CAF}}$ and also $\mathbb{S} \notin \Sigma^{stg}_{\mathrm{CAF}}$, witnessing $\Sigma^{stb}_{\mathrm{CAF}} \subset \Sigma^{stb}_{\mathrm{AF}}$ and $\Sigma^{stg}_{\mathrm{CAF}} \subset \Sigma^{stg}_{\mathrm{AF}}$.

adm, *com*: Let $\tau \in \{adm, com\}$ and consider the extension-set $\mathbb{S} = \{\emptyset, \{a,b\}\}$. The non-compact AF depicted in Figure 3.13 realizes $\mathbb{S}$ under τ, hence $\mathbb{S} \in \Sigma^{\tau}_{AF}$. Now assume there is a compact AF $F \in \mathrm{CAF}_{\tau}$ with $\tau(F) = \mathbb{S}$. Since a and b must not be in conflict there is only one choice for F, namely $F = (\{a,b\},\emptyset)$, which has $adm(F) = \{\emptyset,\{a\},\{b\},\{a,b\}\}$ and $com(F) = \{\{a,b\}\}$. Hence $\mathbb{S} \notin \Sigma^{\tau}_{\mathrm{CAF}}$.

prf, *sem*, *grd**: Let $\tau \in \{prf, sem, grd^*\}$ and consider the extension-set

$$\mathbb{S} = \{\{a,b\},\{a,d,e\},\{b,c,e\}\}.$$

$\mathbb{S} \in \Sigma^{\tau}_{\mathrm{AF}}$ holds since Figure 3.14 shows an AF (with additional arguments) realizing $\mathbb{S}$

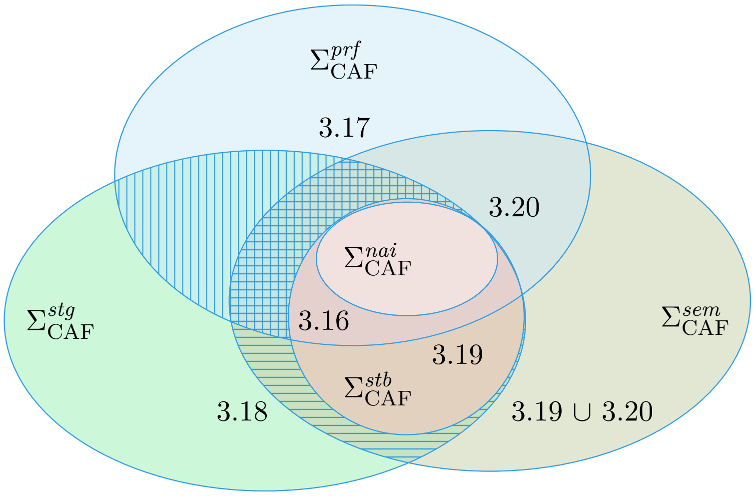


Figure 3.15: A Venn diagram illustrating compact signatures of naive, stable, semi-stable, stage and preferred semantics. The numbers refer to figures showing representative AFs for the respective area. For instance, number X in the area for $\Sigma^{\sigma}_{\mathrm{CAF}} \setminus \Sigma^{\tau}_{\mathrm{CAF}}$ means that for the AF depicted in Figure X, say F, $\sigma(F) \in \Sigma^{\sigma}_{\mathrm{CAF}}$ and $\sigma(F) \notin \Sigma^{\tau}_{\mathrm{CAF}}$. Details are explained in the proof of Theorem 12.

under τ.[4] Now suppose there exists a compact AF $F = (Args_{\mathbb{S}}, R)$ such that $\tau(F) = \mathbb{S}$. Since $\{a, d, e\}, \{b, c, e\} \in \mathbb{S}$, it is clear that R must not contain an edge involving e. But then, e is contained in each $E \in \tau(F)$, hence $\{a, b\} \notin \tau(F)$. It follows that $\tau(F) \neq \mathbb{S}$ and therefore $\mathbb{S} \notin \Sigma^{\tau}_{\mathrm{CAF}}$. □

Note that $\Sigma^{\theta}_{\mathrm{CAF}} = \Sigma^{\theta}_{\mathrm{AF}}$ also holds for single-status semantics such as $\theta \in \{grd, idl\}$, since the canonical realization $(S, \emptyset)$ of extension-sets $\{S\}$ is obviously compact.

In the following we relate the compact signatures of the semantics under consideration to each other. Recall that for general signatures it holds that $\Sigma^{nai}_{\mathrm{AF}} \subset \Sigma^{stg}_{\mathrm{AF}} = (\Sigma^{stb}_{\mathrm{AF}} \setminus \{\emptyset\}) \subset \Sigma^{sem}_{\mathrm{AF}} = \Sigma^{prf}_{\mathrm{AF}}$ (cf. Theorem 2). This picture changes when considering the relationships between compact signatures. Figure 3.15 depicts the relations between compact signatures for naive, stable, stage, semi-stable and preferred semantics, which we will show in the next theorem. The dashed areas represent particular intersections for which the question of existence of extension-sets has to be left open. Also notice that stable semantics cannot realize the empty extension-set within compact AFs.

Theorem 12. *In accordance with Figure 3.15, it holds that:*

1. $\Sigma^{nai}_{CAF} \subset \Sigma^{\sigma}_{CAF}$ *for* $\sigma \in \{stb, stg, sem, prf\}$;
2. $\Sigma^{stb}_{CAF} \subset \Sigma^{\sigma}_{CAF}$ *for* $\sigma \in \{stg, sem\}$;

[4] The self-attacking arguments a' and b' are not needed to realize $\mathbb{S}$ under *prf* or grd^*.

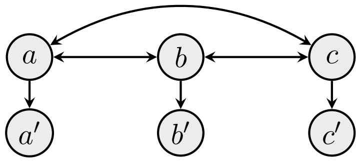


Figure 3.16: AF witnessing $\Sigma_{\text{CAF}}^{nai} \subset \Sigma_{\text{CAF}}^{\sigma}$ for $\sigma \in \{stb, sem, stg, prf\}$.

3. $\Sigma_{CAF}^{prf} \setminus (\Sigma_{CAF}^{stb} \cup \Sigma_{CAF}^{sem} \cup \Sigma_{CAF}^{stg}) \neq \emptyset$;
4. $\Sigma_{CAF}^{stg} \setminus (\Sigma_{CAF}^{stb} \cup \Sigma_{CAF}^{prf} \cup \Sigma_{CAF}^{sem}) \neq \emptyset$;
5. $\Sigma_{CAF}^{stb} \setminus \Sigma_{CAF}^{prf} \neq \emptyset$;
6. $(\Sigma_{CAF}^{prf} \cap \Sigma_{CAF}^{sem}) \setminus (\Sigma_{CAF}^{stb} \cup \Sigma_{CAF}^{stg}) \neq \emptyset$;
7. $\Sigma_{CAF}^{sem} \setminus (\Sigma_{CAF}^{stb} \cup \Sigma_{CAF}^{prf} \cup \Sigma_{CAF}^{stg}) \neq \emptyset$.

Proof. (1) First recall that for a given $\mathbb{S} \in \Sigma_{\text{CAF}}^{nai}$, the canonic AF $F_{cf}(\mathbb{S})$ where $A_F = Args_{\mathbb{S}}$ and $R_F = (Args_{\mathbb{S}} \times Args_{\mathbb{S}}) \setminus Pairs_{\mathbb{S}}$ gives $\mathbb{S} = nai(F_{cf}(\mathbb{S})) = \sigma(F_{cf}(\mathbb{S}))$, and $F_{cf}(\mathbb{S})$ is compact for σ, thus $\Sigma_{\text{CAF}}^{nai} \subseteq \Sigma_{\text{CAF}}^{\sigma}$. Moreover, the AF depicted in Figure 3.16, say F, is compact for σ, since $\sigma(F) = \{\{a, b', c'\}, \{a', b, c'\}, \{a', b', c\}\}$. On the other hand, $\sigma(F)$ cannot be realized under the naive semantics: since $(a', b'), (a', c'), (b', c') \in Pairs_{\sigma(F)}$, any AF F' trying to realize $\sigma(F)$ under nai must also have some $E \in nai(F')$ with $E \supseteq \{a', b', c'\}$. Hence $\Sigma_{\text{CAF}}^{nai} \subset \Sigma_{\text{CAF}}^{\sigma}$.

(2) $\Sigma_{\text{CAF}}^{stb} \subseteq \Sigma_{\text{CAF}}^{\sigma}$ for $\sigma \in \{stg, sem\}$, follows from the fact that $stg(F) = sem(F) = stb(F)$ for every $F \in \text{CAF}_{stb}$ [65]. Properness is by (4) and (7), to be shown in the remainder of this proof.

In the following we provide, as part of the proof, examples witnessing the remaining statements. The general procedure is as follows: Let $\sigma_1, \ldots, \sigma_n$ and $\tau_1, \ldots, \tau_m$ be semantics. To show that $\left(\bigcap_{1 \leq i \leq n} \Sigma_{\text{CAF}}^{\sigma_i}\right) \setminus \left(\bigcup_{1 \leq j \leq m} \Sigma_{\text{CAF}}^{\tau_j}\right) \neq \emptyset$ holds, we fix some extension-set $\mathbb{S} \subseteq 2^{\mathfrak{A}}$, provide an AF $F \in \text{CAF}_{\sigma_i}$ with $\sigma_i(F) = \mathbb{S}$ for all $i \in \{1, \ldots, n\}$,[5] and show that $\mathbb{S}$ is not compactly realizable under any of the semantics $\tau_1, \ldots, \tau_m$.

(3) We begin by showing $\Sigma_{\text{CAF}}^{prf} \setminus (\Sigma_{\text{CAF}}^{stb} \cup \Sigma_{\text{CAF}}^{sem} \cup \Sigma_{\text{CAF}}^{stg}) \neq \emptyset$.

Example 20. Consider the extension-set

$$\mathbb{S} = \{\{a, b\}, \{a, x_i, s_i\}, \{b, y_i, s_i\}, \{x_i, y_i, s_i\} \mid i \in \{1, 2, 3\}\}$$

and observe that the AF F depicted in Figure 3.17 (note that among arguments $\{x_i, y_i \mid i \in \{1, 2, 3\}\}$ all attacks but $\{(x_i, y_i), (y_i, x_i), (x_i, x_i), (y_i, y_i) \mid i \in \{1, 2, 3\}\}$ are present) has exactly $prf(F) = \mathbb{S}$. Since F is compact for prf we have $\mathbb{S} \in \Sigma_{\text{CAF}}^{prf}$. Now let

[5] In fact, we could provide different AFs $F_i \in \text{CAF}_{\sigma_i}$ with $\sigma_i(F_i) = \mathbb{S}$ for each $i \in \{1, \ldots, n\}$.

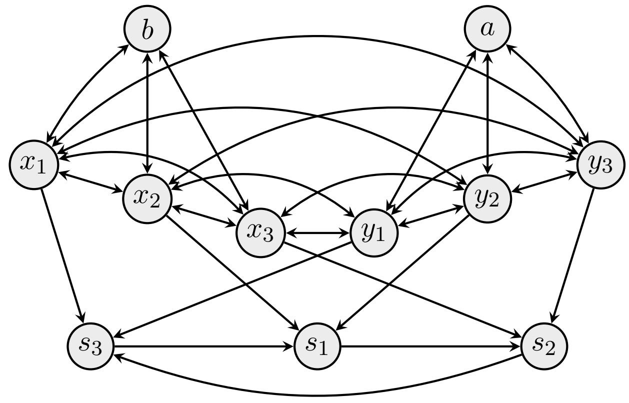


Figure 3.17: AF witnessing $\Sigma^{prf}_{\mathrm{CAF}} \setminus (\Sigma^{stb}_{\mathrm{CAF}} \cup \Sigma^{sem}_{\mathrm{CAF}} \cup \Sigma^{stg}_{\mathrm{CAF}}) \neq \emptyset$.

$\sigma \in \{stb, stg, sem\}$. We show that $\mathbb{S} \notin \Sigma^{\sigma}_{\mathrm{CAF}}$. Towards a contradiction assume that there is an AF G with $A_G = Args_{\mathbb{S}}$ and $\sigma(G) = \mathbb{S}$. First observe that there cannot be any attack between a and b on the one hand and s_1, s_2, and s_3 on the other. For $\sigma = stb$ we have a contradiction to $\sigma(G) = \mathbb{S}$ since $s_1, s_2, s_3 \notin \{a,b\}^+_G$. Also for $\sigma = stg$ we have a contradiction since for each i, $\{a, b, s_i\}$ is conflict-free and $\{a, b, s_i\}^+_G \supset \{a,b\}^+_G$, hence $\{a, b\} \notin stg(G)$. Finally consider $\sigma = sem$. Let $S = \{a, x_1, s_1\}$ and $T = \{x_1, y_1, s_1\}$. If there was no attack between a and y_1 then $S \cup T$ would be conflict-free and admissible and therefore $S, T \notin sem(G)$. Since both T and $\{a, b\}$ must defend themselves, necessarily both $(y_1, a), (a, y_1) \in R_G$. By the symmetric cases we get $\{\langle a, y_i\rangle, \langle b, x_i\rangle \mid i \in \{1, 2, 3\}\} \subseteq R_G$.[6] Now in order to have $\{a, b\} \in sem(G)$, no s_i can be defended by $\{a, b, s_i\}$, hence each s_i must have an attacker that is not attacked by $\{a, b\}$ or s_i. Hence $\{(s_j, s_k), (s_k, s_l), (s_l, s_j)\} \subseteq R_G$ for some $j, k, l \in \{1, 2, 3\}$ with $j \neq k \neq l \neq j$ and no other attacks among $\{s_j, s_k, s_l\}$. W.l.o.g. assume $j = 1$, $k = 2$, and $l = 3$. Now observe that S has to defend s_1 from s_3, therefore $(x_1, s_3) \in R_G$. So far we have $S^+_G \supseteq (Args_{\mathbb{S}} \setminus \{x_2, x_3\})$. S has to attack both x_2 and x_3 since otherwise either S would not defend itself or at least one of $S \cup \{x_2\}$ and $S \cup \{x_3\}$ would be admissible and have greater range than S. But now $S^+_G = Args_{\mathbb{S}} \supset \{a, b\}^+_G$, a contradiction to $\{a, b\} \in sem(G)$. $\Diamond$

(4) We continue with $\Sigma^{stg}_{\mathrm{CAF}} \setminus (\Sigma^{stb}_{\mathrm{CAF}} \cup \Sigma^{prf}_{\mathrm{CAF}} \cup \Sigma^{sem}_{\mathrm{CAF}}) \neq \emptyset$.

Example 21. Let $\oplus$ such that $i \oplus j = (i + j) \mod 9$. Consider the AF $F = (\{a_0, \ldots, a_8\}, \{(a_i, a_{i\oplus 1}) \mid 0 \leq i < 9, \})$, i.e. the directed cycle of nine arguments depicted in Figure 3.18. We get $stg(F) = \{\{a_i, a_{i\oplus 2}, a_{i\oplus 4}, a_{i\oplus 6}\} \mid 0 \leq i < 9\}$. Now assume this extension-set is compactly realizable under stable, preferred or semi-stable semantics, i.e. there is some G with $\sigma(G) = stg(F)$ ($\sigma \in \{stb, prf, sem\}$) and $A_G = A_F$. Since a_i and a_j occur together in some stage extension of F for all i, j with $i \oplus 1 \neq j$ and $i \neq j \oplus 1$, the only possible attacks in G are (a_i, a_j) with $i \oplus 1 = j$ or $i = j \oplus 1$. Now let $S_i = \{a_i, a_{i\oplus 2}, a_{i\oplus 4}, a_{i\oplus 6}\}$.

[6] Recall the notation for symmetric attacks: for arguments $a, b \in \mathfrak{A}$, we use $\langle a, b\rangle$ as shortcut for $(a, b), (b, a)$.

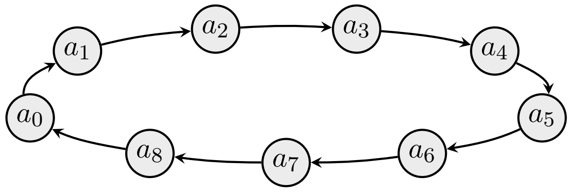


Figure 3.18: AF witnessing $\Sigma^{stg}_{\text{CAF}} \setminus (\Sigma^{stb}_{\text{CAF}} \cup \Sigma^{prf}_{\text{CAF}} \cup \Sigma^{sem}_{\text{CAF}}) \neq \emptyset$.

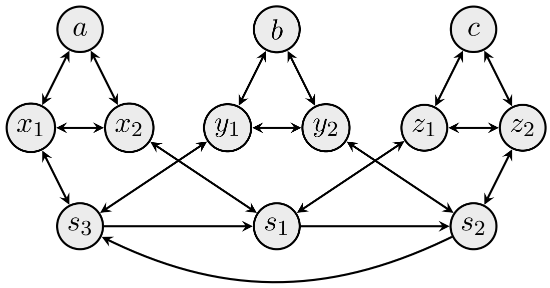


Figure 3.19: AF witnessing $\Sigma^{stb}_{\text{CAF}} \setminus \Sigma^{prf}_{\text{CAF}} \neq \emptyset$.

In order to have $S_i \in \sigma(G)$, a_i has to attack $a_{i\oplus 8}$ and $a_{i\oplus 6}$ has to attack $a_{i\oplus 7}$, first for S_i (resp. $(S_i)^+_G$ for *stb* and *sem*) to be maximal and second to be defended. Hence $R_G = \{\langle a_i, a_{i\oplus 1}\rangle \mid 0 \leq i < 9\}$, i.e. the cycle of length 9 with all attacks being symmetric. Consequently, $\sigma(G) = stg(F) \cup \{a_i, a_{i\oplus 3}, a_{i\oplus 6} \mid 0 \leq i < 3\}$, showing that there is no AF compactly realizing $stg(F)$ under σ. $\diamondsuit$

(5) The following example witnesses that $\Sigma^{stb}_{\text{CAF}} \setminus \Sigma^{prf}_{\text{CAF}} \neq \emptyset$.

Example 22. Consider the AF F depicted in Figure 3.19 and observe that

$$\begin{aligned}
\mathbb{S} = stb(F) = \{&\{a, b, z_1, s_2\}, \{a, b, z_2, s_3\},\\
&\{a, c, y_1, s_1\}, \{a, c, y_2, s_3\},\\
&\{b, c, x_1, s_1\}, \{b, c, x_2, s_2\},\\
&\{a, y_1, z_1, s_2\}, \{a, y_1, z_2, s_1\}, \{a, y_2, z_1, s_3\}, \{a, y_2, z_2, s_3\},\\
&\{b, x_1, z_1, s_2\}, \{b, x_1, z_2, s_1\}, \{b, x_2, z_1, s_2\}, \{b, x_2, z_2, s_3\},\\
&\{c, x_1, y_1, s_1\}, \{c, x_1, y_2, s_1\}, \{c, x_2, y_1, s_2\}, \{c, x_2, y_2, s_3\},\\
&\{x_1, y_1, z_1, s_2\}, \{x_1, y_1, z_2, s_1\}, \{x_1, y_2, z_1\}, \{x_1, y_2, z_2, s_1\},\\
&\{x_2, y_1, z_1, s_2\}, \{x_2, y_1, z_2\}, \{x_2, y_2, z_1, s_3\}, \{x_2, y_2, z_2, s_3\}\}.
\end{aligned}$$

Note that $\{a, b, c\}$ does not have full range and thus is not a stable extension of F. Assume there exists some AF G compactly realizing $\mathbb{S}$ under preferred semantics, i.e. $prf(G) = \mathbb{S}$ and $A_G = Args_{\mathbb{S}}$. One can check that every pair of arguments in F which does not feature an attack actually occurs in some stable extension of F (we call such an

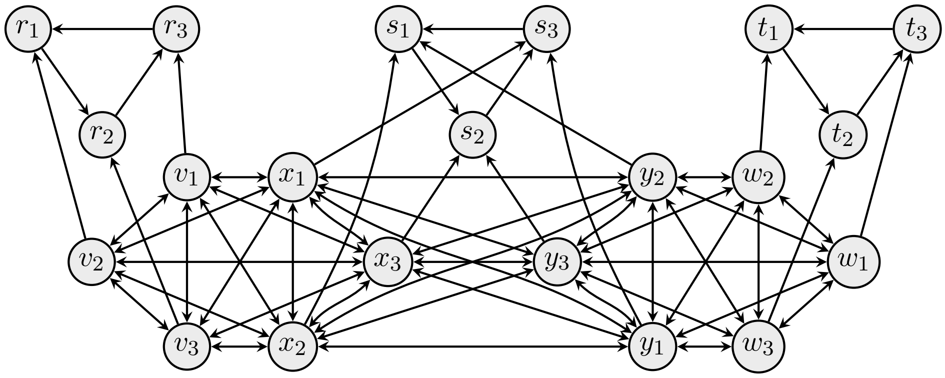


Figure 3.20: AF showing $(\Sigma_{\mathrm{CAF}}^{prf} \cap \Sigma_{\mathrm{CAF}}^{sem}) \setminus (\Sigma_{\mathrm{CAF}}^{stb} \cup \Sigma_{\mathrm{CAF}}^{stg}) \neq \emptyset$.

F analytic in [31]). That means that for the AF G there can only be attacks between arguments being linked in Figure 3.19.

Now consider the extension $S = \{b, c, x_1, s_1\} \in \mathbb{S}$. For $S \in prf(G)$ there are two possible reasons for $a \notin S$. Either a is attacking or attacked by S, or a is not defended by $S \cup \{a\}$. Assume a not to be defended by $S \cup \{a\}$. Then $(x_2, a) \in R_G$ and $(x_1, x_2), (s_1, x_2) \notin R_G$. But then $x_2 \notin S$ defends itself, hence S cannot be a maximal admissible set in G. It follows that a is in conflict with S, the only possibility being a conflict with x_1, hence $(x_1, a) \in R_G$ ($(a, x_1) \in R_G$ is not sufficient since no other argument in S can defend x_1 against a). Considering $\{a, y_1, z_1, s_2\} \in \mathbb{S}$, none of y_1, z_1, and s_2 can defend a against x_1, hence also $(a, x_1) \in R_G$.

In the very same manner, one can justify the existence of symmetric attacks between a and x_2, b and y_i, and c and z_i ($i \in \{1, 2\}$), respectively. Therefore the set $\{a, b, c\}$ is admissible in G. Hence there must be some $S' \in prf(G)$ with $S' \supseteq \{a, b, c\}$, a contradiction to $\mathbb{S}$ being compactly realizable under the preferred semantics. $\Diamond$

(6) We proceed with an example showing that $(\Sigma_{\mathrm{CAF}}^{prf} \cap \Sigma_{\mathrm{CAF}}^{sem}) \setminus (\Sigma_{\mathrm{CAF}}^{stb} \cup \Sigma_{\mathrm{CAF}}^{stg}) \neq \emptyset$.

Example 23. Consider the AF F from Figure 3.20. We have

$$\mathbb{S} = sem(F) = prf(F) = \{\{v_i, y_j, r_i, s_j\}, \{w_i, x_j, t_i, s_j\}, \{v_i, w_j, r_i, t_j\} \mid 1 \leq i, j \leq 3\}.$$

Let $\sigma \in \{stb, stg\}$ and assume there is an AF G with $\sigma(G) = \mathbb{S}$ and and $A_G = Args_{\mathbb{S}}$. First note that for all $i, j \in \{1, 2, 3\}$ each pair $\{v_i, s_j\}, \{w_i, s_j\}, \{r_i, s_j\}, \{t_i, s_j\}$ is contained in some element of $\mathbb{S}$, hence there cannot be an attack between any of these pairs in G. Now let $S = \{v_i, w_j, r_i, t_j\}$ for some $i, j \in \{1, 2, 3\}$. We have $S_G^+ \subseteq A_G \setminus \{s_1, s_2, s_3\}$, hence S cannot be a stable extension of G. Moreover, since G must be self-loop-free, $S \cup \{s_k\}$ with $1 \leq k \leq 3$ is conflict-free and obviously has a bigger range than S. Therefore S cannot be a stage extension in G. It follows that $\mathbb{S}$ cannot be compactly realized under $\sigma \in \{stb, stg\}$. $\Diamond$

(7) For $\Sigma_{\text{CAF}}^{sem} \setminus (\Sigma_{\text{CAF}}^{stb} \cup \Sigma_{\text{CAF}}^{prf} \cup \Sigma_{\text{CAF}}^{stg}) \neq \emptyset$ we will make use of the following lemma, which might be of interest on its own.

Lemma 10. *Let $\sigma, \tau \in \{stb, prf, sem, stg\}$ and $F_1, F_2 \in CAF_\tau$ such that $\tau(F_1) \notin \Sigma_{CAF}^{\sigma}$ and $A_{F_1} \cap A_{F_2} = \emptyset$. It holds that $\tau(F_1 \cup F_2) \notin \Sigma_{CAF}^{\sigma}$.*

Proof. Assume, towards a contradiction, there is some AF $G \in \text{CAF}_\sigma$ such that $\sigma(G) = \tau(F_1 \cup F_2)$. Since $A_{F_1} \cap A_{F_2} = \emptyset$, it follows that $\tau(F_1 \cup F_2) = \{E_1 \cup E_2 \mid E_1 \in \tau(F_1), E_2 \in \tau(F_2)\}$.[7] Due to compactness of F_1 and F_2, every argument $a \in A_{F_1}$ occurs together with every argument $b \in A_{F_2}$ in some τ-extension of $F_1 \cup F_2$, meaning that G cannot contain any attack between a and b. Hence $G = G_1 \cup G_2$ with $A_{G_1} = A_{F_1}$ and $A_{G_2} = A_{F_2}$ and, consequently, $\sigma(G) = \{E_1 \cup E_2 \mid E_1 \in \sigma(F_1), E_2 \in \sigma(F_2)\}$. Therefore it must hold that $\sigma(G_1) = \tau(F_1)$, a contradiction to the assumption that $\tau(F_1) \notin \Sigma_{\text{CAF}}^{\sigma}$. $\square$

Now we get $\Sigma_{\text{CAF}}^{sem} \setminus (\Sigma_{\text{CAF}}^{stb} \cup \Sigma_{\text{CAF}}^{prf} \cup \Sigma_{\text{CAF}}^{stg}) \neq \emptyset$ as follows: Let $F = F_1 \cup F_2$ where F_1 is the AF in Figure 3.19 and F_2 is the AF in Figure 3.20 (observe that for $A_{F_1} \cap A_{F_2} = \emptyset$ some renaming is necessary). From $F_1, F_2 \in \text{CAF}_{sem}$ and $sem(F_1) \notin \Sigma_{\text{CAF}}^{prf}$ (see Example 22) we get $sem(F) \notin \Sigma_{\text{CAF}}^{prf}$ by Lemma 10. In the same way $sem(F) \notin \Sigma_{\text{CAF}}^{stb} \cup \Sigma_{\text{CAF}}^{stg}$ follows from $F_1, F_2 \in \text{CAF}_{sem}$ and $sem(F_2) \notin \Sigma_{\text{CAF}}^{stb} \cup \Sigma_{\text{CAF}}^{stg}$ (see Example 23).

This concludes the proof of Theorem 12. $\square$

The compact signature for single-status semantics such as *grd* and *idl* is $\{\{S\} \mid S \subseteq \mathfrak{A}\}$, with S being finite, and therefore contained in all signatures covered by Theorem 12. It is also contained in the compact signatures of resolution-based grounded semantics, of which we also know that $\Sigma_{\text{CAF}}^{nai} \not\subseteq \Sigma_{\text{CAF}}^{grd^*}$. This is already witnessed by the extension-set $\{\{a\}, \{b\}, \{c\}\}$, which is not (compactly) realizable under grd^* (cf. Proposition 9), but compactly realizable under *nai*.

The following theorem shows the relations between the compact signature for *cf*, *adm*, and *com*, i.e. the semantics for which extension-sets are not incomparable (except $\{\emptyset\}$ or extension-set of size 1 for *com*).

Theorem 13. *It holds that*

1. $\Sigma_{CAF}^{cf} \subset \Sigma_{CAF}^{adm}$ *and*
2. $\Sigma_{CAF}^{com} \not\subseteq \Sigma_{CAF}^{\sigma}$ *and* $\Sigma_{CAF}^{\sigma} \not\subseteq \Sigma_{CAF}^{com}$ *for* $\sigma \in \{cf, adm\}$.

Proof. (1) Given an arbitrary AF F it holds that $cf(F) = adm(F^{sym})$, where F^{sym} is the AF obtained from making all attacks of F symmetric and removing all self-attacking arguments, hence $\Sigma_{\text{CAF}}^{cf} \subseteq \Sigma_{\text{CAF}}^{adm}$. Properness of the relation is by the AF

[7] This property has been called well-definedness by Spanring [182] and holds for all semantics under consideration.

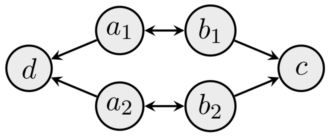


Figure 3.21: AF F compactly realizing an extension-set $\mathbb{S} \notin \Sigma_{\text{CAF}}^{adm} \cup \Sigma_{\text{CAF}}^{cf}$ under *com*.

$G = (\{a, b, c, d\}, \{(a, b), (b, c), (c, d), (d, a)\})$, that is the directed cycle of four arguments, having $adm(G) = \{\emptyset, \{a, c\}, \{b, d\}\}$, which is an extension-set not (compactly) realizable under *cf*. This is by the fact that if $\{a, c\}$ is conflict-free in some AF then clearly also $\{a\}$ and $\{c\}$ must be conflict-free. Hence $\Sigma_{\text{CAF}}^{cf} \subset \Sigma_{\text{CAF}}^{adm}$.

(2) $\Sigma_{\text{CAF}}^{com} \not\subseteq \Sigma_{\text{CAF}}^{\sigma}$: Any extension-set $\mathbb{S}$ containing exactly one non-empty set of arguments S is compactly realizable under *com* by the AF $(S, \emptyset)$, but not under *cf* and *adm* since $\emptyset$ is not contained in $\mathbb{S}$. The following example shows that these trivial cases are not the only AFs in $\Sigma_{\text{CAF}}^{com} \setminus \Sigma_{\text{CAF}}^{\sigma}$. To this end consider the AF F depicted in Figure 3.21. We have $com(F) = \{\emptyset, \{a_1\}, \{a_2\}, \{b_1\}, \{b_2\}, \{a_1, b_2\}, \{a_2, b_1\}, \{a_1, a_2, c\}, \{b_1, b_2, d\}\}$. On the one hand it is easy to see that F is compact for complete semantics, on the other hand observe that both $\{a_1\}, \{a_2\} \in com(F)$, $(a_1, a_2) \in Pairs_{com(F)}$, but $\{a_1, a_2\} \notin com(F)$. So $com(F)$ is not conflict-sensitive. Hence $com(F) \notin \Sigma_{\text{AF}}^{\sigma}$ and therefore also $com(F) \notin \Sigma_{\text{CAF}}^{\sigma}$.

$\Sigma_{\text{CAF}}^{\sigma} \not\subseteq \Sigma_{\text{CAF}}^{com}$: Let $F = (\{a, b, c\}, \{\langle a, b\rangle, \langle b, c\rangle\})$ and observe that $cf(F) = adm(F) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, c\}\}$. Now assume there is an AF $G \in \text{CAF}_{com}$ with $com(G) = cf(F)$. Clearly $A_G = \{a, b, c\}$ and $R_G \subseteq \{(a, b), (b, a), (b, c), (c, b)\}$. Now for $\emptyset \in com(G)$ each argument must be attacked and, moreover, the singletons $\{a\}$, $\{b\}$ and $\{c\}$ must defend themselves. Hence it must be that $G = F$ which means $com(G) = \{\emptyset, \{b\}, \{a, c\}\}$, a contradiction. □

Comparing the insights obtained from Theorems 12 and 13 with the results on general realizability presented in Theorem 2, we observe notable differences depending on whether rejected arguments are allowed or not. When allowing rejected arguments (as in general realizability), preferred and semi-stable semantics are equally expressive and at the same time strictly more expressive than stable and stage semantics. As we have seen, this does not carry over to the compact setting where, with the exception of $\Sigma_{\text{CAF}}^{stb} \subset \Sigma_{\text{CAF}}^{sem}$ and $\Sigma_{\text{CAF}}^{stb} \subset \Sigma_{\text{CAF}}^{stg}$, signatures among *stb*, *prf*, *sem*, and *stg* are incomparable. Moreover, while complete semantics is strictly more expressive than admissible and conflict-free semantics in general, the compact signatures are incomparable.

What remains an open issue is the existence of extension-sets lying in the intersection of $\Sigma_{\text{CAF}}^{prf}$ (resp. $\Sigma_{\text{CAF}}^{sem}$) and $\Sigma_{\text{CAF}}^{stg}$ but outside of $\Sigma_{\text{CAF}}^{stb}$ (see the dashed areas in the Venn diagram in Figure 3.15). Due to considerations involving implicit conflicts we conjecture in [31] that both intersections are empty, i.e. $\Sigma_{\text{CAF}}^{prf} \cap \Sigma_{\text{CAF}}^{stg} \subset \Sigma_{\text{CAF}}^{stb}$ and $\Sigma_{\text{CAF}}^{sem} \cap \Sigma_{\text{CAF}}^{stg} = \Sigma_{\text{CAF}}^{stb}$.

3.4 Input-Output Realizability

In this section we consider realizability in the context of the decomposability of AFs. This is based on the idea that an AF can be considered as a composition of modules and their interconnections. Such a module is itself an AF with designated input and output arguments, i.e. external arguments affecting resp. being affected by the module. Its behaviour in the interplay with other modules is described by a function which maps each acceptance status (either extensions or labellings) of the input arguments into the set of acceptance statuses of the output arguments. Independently of the concrete module, this behaviour will be described by what we will call I/O-specifications. The question of interest is, given a semantics σ and an I/O-specification $\mathfrak{f}$, whether there is a module (later called I/O-module) realizing $\mathfrak{f}$ under σ, i.e. showing, when evaluated under σ, exactly the same input-output behaviour as prescribed by $\mathfrak{f}$. The set of I/O-specifications realizable by a semantics constitutes its I/O-signature and reveals the degree of functional completeness of the semantics.

Recalling the example from the introduction, the sub-framework including the 4-length chain realizes the mapping where $\{a_1\}$ is mapped to $\emptyset$ (i.e. if a_1 belongs to an extension then a_4 does not belong to it) and $\emptyset$ is mapped to $\{a_4\}$ (i.e. if a_1 does not belong to an extension then a_4 belongs to it): we call this kind of mapping a two-valued I/O-specification. On the other hand, one may want to distinguish between "out" arguments and "undecided" arguments. Considering the AF given by the 4-lengh chain, if a_1 is accepted then a_4 is out, if a_1 is out then a_4 is in, if a_1 is undecided then a_4 is undecided too. We call this kind of mapping a three-valued I/O-specification. As it will be shown in the following, not all three-valued I/O-specifications are realizable, e.g. there is no sub-framework realizing the variant of the mapping above where a_1 undecided yields a_4 accepted.

The remainder of this section is organized as follows. We will first study I/O-realizability in the extension-based (i.e. two-valued) setting. After introducing the necessary formal machinery we will exactly characterize all realizable two-valued I/O-specifications for the stable, preferred, semi-stable, stage, complete, ideal, and grounded semantics. Then we will deal with labelling-based (i.e. three-valued) I/O-realizability, giving exact characterizations for the preferred and grounded semantics. Finally we will consider partial specifications where for some inputs the output is not specified.

3.4.1 Two-valued I/O-realizability

So far we were interested in the realizability of extension-sets, i.e. finding an AF such that the evaluation under a semantics gives a certain result. When regarding AFs as modules within a larger system, we are rather interested in the effect of input arguments on output arguments. To this end we introduce the notion of a two-valued I/O-specification, which describes a desired input-output behaviour by assigning to each set of input arguments a set of sets of output arguments. The underlying idea is that a certain subset of input arguments being accepted can cause several outcomes, each described by a set of accepted output arguments.

Definition 47. A *(two-valued) I/O-specification* consists of two sets $I, O \subseteq \mathfrak{A}$ and a total[8] function $\mathfrak{f} : 2^I \mapsto 2^{2^O}$.

In order to realize given I/O-specifications we require AFs with dedicated input and output argument. An I/O-module represents a (partial) AF where two sets of arguments are identified as input and output arguments, respectively, with the restriction that input arguments do not have any ingoing attacks.

Remark 2. Differently from an I/O-module, the notion of *argumentation multipole* in [17] assumes a fixed set of incoming and outgoing attacks rather than of input and output arguments. However, for the purposes of this work the two notions are equivalent insofar as input (output) arguments of an I/O-module are identified with the sources (destinations) of incoming (outgoing) attacks of the corresponding argumentation multipole.

Definition 48. Given a set of input arguments $I \subseteq \mathfrak{A}$ and a set of output arguments $O \subseteq \mathfrak{A}$ with $I \cap O = \emptyset$, an *I/O-module* is an AF $F = (A, R)$ such that $I, O \subseteq A$ and $I_F^- = \emptyset$.

In order to evaluate the effect of input arguments on output arguments we introduce the notion of an injection, which intuitively assigns a certain status to the input arguments. The injection of a set $J \subseteq I$ to an I/O-module F simulates the input J in the way that all arguments in J are accepted (none of them has ingoing attacks since F is an I/O-module) and all arguments in $(I \setminus J)$ are rejected (each of them is attacked by the newly introduced argument z, which has no ingoing attacks).

Definition 49. Given an I/O-module $F = (A, R)$ and a set of arguments $J \subseteq I$, the *(two-valued) injection* of J to F is the AF

$$\triangleright(F, J) = (A \cup \{z\}, R \cup \{(z, i) \mid i \in (I \setminus J)\}),$$

where z is a newly introduced argument.

Now we have all the tools at hand to define what it means to realize a given I/O-specification. In order for an I/O-module F to realize $\mathfrak{f}$ under a semantics σ, the injection of each $J \subseteq I$ to F must have $\mathfrak{f}(J)$ as its σ-extensions restricted to the output arguments. So, informally, with input J assigned the set of outputs under σ must be exactly $\mathfrak{f}(J)$.

Definition 50. Given $I, O \subseteq \mathfrak{A}$, a semantics σ and an I/O-specification $\mathfrak{f}$, the I/O-module F *realizes* $\mathfrak{f}$ under σ iff

$$\forall J \subseteq I : \sigma(\triangleright(F, J))|_O = \mathfrak{f}(J).$$

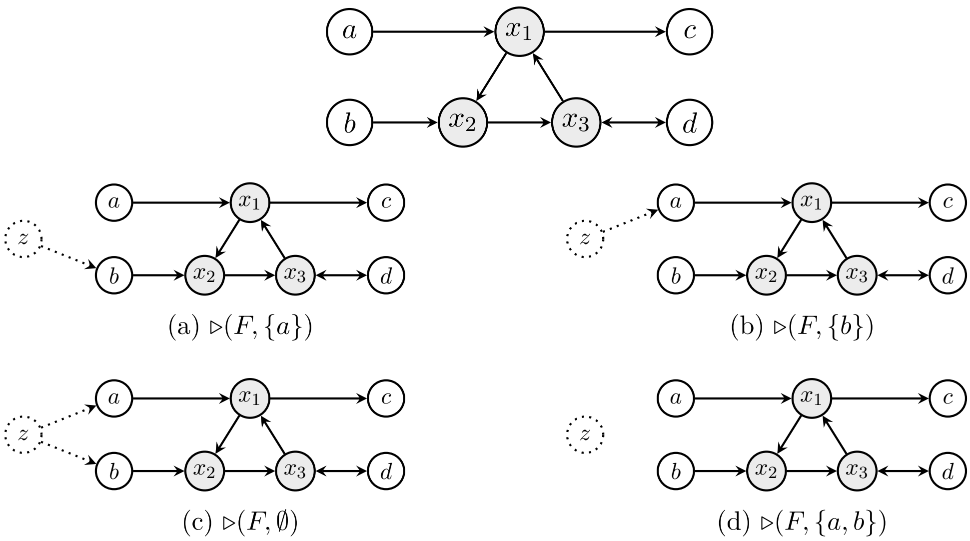


Figure 3.22: I/O-module F with $I = \{a, b\}$ and $O = \{c, d\}$ realizing the I/O-specification given in Example 24 under $\{\mathit{stb}, \mathit{prf}, \mathit{sem}, \mathit{stg}\}$ on top and the injections of all possible input assignments in (a) – (d).

The following example illustrates these basic concepts.

Example 24. Consider the sets $I = \{a, b\}$ and $O = \{c, d\}$. A exemplary I/O-specification is given by the function $\mathfrak{f} : 2^I \mapsto 2^{2^O}$ such that

$$\begin{aligned} \mathfrak{f}(\emptyset) &= \{\{d\}\} \\ \mathfrak{f}(\{a\}) &= \{\{c, d\}\} \\ \mathfrak{f}(\{b\}) &= \{\{c\}, \{d\}\} \\ \mathfrak{f}(\{a, b\}) &= \{\{c, d\}, \{c\}\} \end{aligned}$$

Considering, for instance, the case of input $\{b\}$, the intended meaning of $\mathfrak{f}$ is that if b is accepted and a is not, then either c or d, but not both, should be accepted. On the other hand, in the case of input $\{a\}$, i.e. a is accepted and b is not, both c and d should be accepted. The AF F in Figure 3.22, represents an I/O-module with dedicated input arguments $\{a, b\}$ and output arguments $\{c, d\}$. It turns out that F realizes $\mathfrak{f}$ under stable semantics. In order to show this we have to check, for each $J \subseteq I$, whether the injection of J to F, $\triangleright(F, J)$, has exactly $\mathfrak{f}(J)$ as stable extensions restricted to O, i.e. $\mathit{stb}(\triangleright(F, J))|_O = \mathfrak{f}(J)$. Considering $J = \emptyset$ (cf. Figure 3.22c), we have that $\triangleright(F, \emptyset)$ is F together with a new argument z attacking both a and b. Hence we have $\mathit{stb}(\triangleright(F, \emptyset)) = \{\{z, x_1, d\}\}$, meaning that $\mathit{stb}(\triangleright(F, \emptyset))|_O = \{\{d\}\}$, which is as specified by $\mathfrak{f}$. Moreover we get $\mathit{stb}(\triangleright(F, \{a\})) = \{\{z, a, x_2, c, d\}\}$ (cf. Figure 3.22a), $\mathit{stb}(\triangleright(F, \{b\})) = \{\{z, b, x_3, c\}, \{z, b, x_1, d\}\}$ (cf. Figure 3.22b), and $\mathit{stb}(\triangleright(F, \{a, b\})) =$

[8] We will study the case of partial functions on its own in Section 3.4.3.

$\{\{z, a, b, c, d\}, \{z, a, b, x_3, c\}\}$ (cf. Figure 3.22d). This shows that for all possible inputs, the extensions restricted to the output arguments are as specified by $\mathfrak{f}$, hence F realizes $\mathfrak{f}$ under stable semantics.

While it is easy to verify that F also realizes $\mathfrak{f}$ under preferred, semi-stable and stage semantics, it does not realize $\mathfrak{f}$ under grounded, ideal and complete semantics. For $J = \{b\}$ we have that $grd(\triangleright(F, \{b\}))|_O = idl(\triangleright(F, \{b\}))|_O = \{\emptyset\}$ and $com(\triangleright(F, \{b\}))|_O = \{\emptyset, \{c\}, \{d\}\}$, being not in line with $\mathfrak{f}$ (recall that $\mathfrak{f}(b) = \{\{c\}, \{d\}\}$). $\diamondsuit$

The question we want to address is the following: which conditions must $\mathfrak{f}$ fulfill in order to be realizable by some I/O-module and how can such an I/O-module be constructed? The following generic AF will be the key concept for the forthcoming characterization results.

Definition 51. Given an I/O-specification $\mathfrak{f}$, let $Y = \{y_i \mid i \in I\}$ and $X = \{x_J^S \mid J \subseteq I, S \in \mathfrak{f}(J)\}$. The *canonical I/O-module* (for $\mathfrak{f}$) is defined as

$$\begin{aligned}
\mathfrak{C}(\mathfrak{f}) = (&I \cup O \cup Y \cup X \cup \{w\}, \\
&\{(i, y_i) \mid i \in I\} \cup \\
&\{(y_i, x_J^S) \mid x_J^S \in X, i \in J\} \cup \{(i, x_J^S) \mid x_J^S \in X, i \in (I \setminus J)\} \cup \\
&\{(x, x') \mid x, x' \in X, x \neq x'\} \cup \{(x, w) \mid x \in X\} \cup \{(w, w)\} \cup \\
&\{(x_J^S, o) \mid x_J^S \in X, o \in (O \setminus S)\}).
\end{aligned}$$

Besides the dedicated input and ouput argumentss, $\mathfrak{C}(\mathfrak{f})$ consist of a copy of each input argument, an argument for each combination of input and output given by $\mathfrak{f}$, as well as the argument w. Intuitively, the argument x_J^S shall enforce output S for input J. It does so by attacking all other arguments in X and all output arguments except S. Moreover, w ensures that any stable extension of (an injection to) $\mathfrak{C}(\mathfrak{f})$ must contain at least one argument of X.

The following theorem shows that any I/O-specification is realizable under stable semantics.

Theorem 14. *Every I/O-specification $\mathfrak{f}$ is realized by $\mathfrak{C}(\mathfrak{f})$ under stb.*

Proof. Let $I, O \subseteq \mathfrak{A}$ and $\mathfrak{f}$ be an arbitrary I/O-specification. We have to show that $stb(\triangleright(\mathfrak{C}(\mathfrak{f}), J))|_O = \mathfrak{f}(J)$ holds for any $J \subseteq I$. Consider such a $J \subseteq I$.

First let $S \in \mathfrak{f}(J)$. We show that $E = \{z\}$[9]$\cup J \cup \{y_i \mid i \in (I \setminus J)\} \cup \{x_J^S\} \cup S \in stb(\triangleright(\mathfrak{C}(\mathfrak{f}), J))$, thus $S \in stb(\triangleright(\mathfrak{C}(\mathfrak{f}), J))|_O$. E is conflict-free in $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$ since z is only in conflict with the arguments $(I \setminus J)$, a y_i with $i \in (I \setminus J)$ is only in conflict with $i \notin E$, x_J^S is only in conflict with other $x \in X$, $i \in (I \setminus J)$ and y_j with $j \in J$, and arguments in S are only in conflict with arguments from X but not from x_J^S. E is stable in $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$ since x_J^S

[9]Recall that the argument z is introduced by the injection.

attacks w, all other $x \in X$ and all $o \in (O \setminus S)$; z attacks all $i \in (I \setminus J)$; each y_j with $j \in J$ is attacked by j.

It remains to show that there is no $S' \in stb(\triangleright(\mathfrak{C}(\mathfrak{f}), J))|_O$ with $S' \notin \mathfrak{f}(J)$. Towards a contradiction assume there is some $S' \in stb(\triangleright(\mathfrak{C}(\mathfrak{f}), J))|_O$ with $S' \notin \mathfrak{f}(J)$. Hence there must be some $E' \in stb(\triangleright(\mathfrak{C}(\mathfrak{f}), J))$ with $S' \subset E'$. Since w attacks itself, $w \notin E'$, thus by construction of $\mathfrak{C}(\mathfrak{f})$ there must be some $x_{J'}^{S'} \in (X \cap E')$ attacking w, and $x_{J'}^{S'}$ must attack all $o \in (O \setminus S')$. Since $S' \notin \mathfrak{f}(J)$ by assumption, it must hold that $J' \neq J$. Now note that $z \in E'$ and $j \in E'$ for all $j \in J$, since they are not attacked by construction of $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$. Now if $J' \subset J$ then there is some $j \in (J \setminus J')$ attacking $x_{J'}^{S'}$, a contradiction to conflict-freeness of E'. On the other hand if $J' \not\subseteq J$ there is some $j' \in (J' \setminus J)$ which is attacked by z. Therefore also $y_{j'} \in E'$, which attacks $x_{J'}^{S'}$, again a contradiction. $\square$

As to preferred, semi-stable and stage semantics, any I/O-specification is realizable, provided that a (possibly empty) output is prescribed for any input.

Proposition 14. *Every I/O-specification $\mathfrak{f}$ such that $\forall J \subseteq I : \mathfrak{f}(J) \neq \emptyset$ is realized by $\mathfrak{C}(\mathfrak{f})$ under $\sigma \in \{prf, sem, stg\}$.*

Proof. We show that for all $J \subseteq I$ stable, preferred, stage and semi-stable extensions coincide in $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$, thus the result follows from Theorem 14.

First, according to the hypothesis and Theorem 14, there exists a stable extension of $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$ for each $J \subseteq I$, thus stable, stage and semi-stable extensions coincide. As to preferred semantics, we know that any stable extension is also preferred, and we show that the reverse also holds in $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$. By construction of $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$, it is easy to see that for any preferred extension E it holds that $E = \{z\} \cup J \cup \{y_i \mid i \in (I \setminus J)\} \cup \{x_J^S\} \cup S$, where S is a set among $\mathfrak{f}(J)$ and exists by the hypothesis. E is stable, since x_J^S attacks w, all other $x \in X$ and all $o \in (O \setminus S)$, z attacks all $i \in (I \setminus J)$, and each y_j with $j \in J$ is attacked by j. $\square$

Theorem 15. *An I/O-specification $\mathfrak{f}$ is realizable under $\sigma \in \{prf, sem, stg\}$ iff $\forall J \subseteq I$: $\mathfrak{f}(J) \neq \emptyset$.*

Proof. The if-direction is a direct consequence of Proposition 14.

The only-if-direction follows directly by the fact that in every AF, particularly in any injection of some set of arguments to an I/O-module, a σ-extension exists. $\square$

Example 25. Consider the I/O-specification $\mathfrak{f}$ with $I = \{a, b\}$ and $O = \{c, d, e\}$ defined as follows:

$$\begin{aligned}
\mathfrak{f}(\emptyset) &= \{\emptyset\} \\
\mathfrak{f}(\{a\}) &= \{\{c, e\}\} \\
\mathfrak{f}(\{b\}) &= \{\{c, d, e\}, \{d, e\}\} \\
\mathfrak{f}(\{a, b\}) &= \{\{c, d, e\}, \{c, d\}\}
\end{aligned}$$

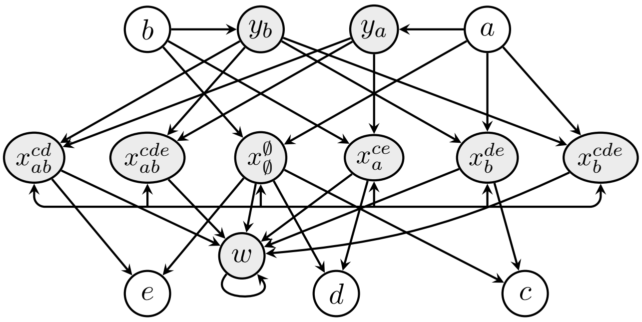


Figure 3.23: I/O-module realizing the I/O-specification given in Example 25 under $\{stb, prf, sem, stg\}$.

The canonical I/O-module $\mathfrak{C}(\mathfrak{f})$ is depicted in Figure 3.23[10]. Let σ be a semantics among $\{stb, prf, sem, stg\}$. One can verify that for every possible input $J \subseteq I$, the injection of J to $\mathfrak{C}(\mathfrak{f})$ has exactly $\mathfrak{f}(J)$ as σ-extensions restricted to O. As an example let $J = \{b\}$. $\triangleright(\mathfrak{C}(\mathfrak{f}), \{b\})$ adds to $\mathfrak{C}(\mathfrak{f})$ the argument z attacking a. Now

$$\sigma(\triangleright(\mathfrak{C}(\mathfrak{f}), \{b\})) = \{\{z, b, y_a, x^{\{d,e\}}_{\{b\}}, d, e\}, \{z, b, y_a, x^{\{c,d,e\}}_{\{b\}}, c, d, e\}\},$$

hence $\sigma(\triangleright(\mathfrak{C}(\mathfrak{f}), \{b\}))|_O = \{\{d, e\}, \{c, d, e\}\} = \mathfrak{f}(\{b\})$. $\Diamond$

Also for complete, grounded and ideal semantics we are able to identify a necessary and sufficient condition for realizability. While we show sufficiency of these conditions in more detail, their necessity is by the well-known facts that the intersection of all complete extensions is always a complete extension too and ideal and grounded semantics always yield exactly one extension. We define the former property for I/O-specifications:

Definition 52. An I/O-specification $\mathfrak{f}$ is *closed* iff for each $J \subseteq I$ it holds that $\mathfrak{f}(J) \neq \emptyset$ and $\bigcap \mathfrak{f}(J) \in \mathfrak{f}(J)$.

Example 26. Again considering the I/O-specification $\mathfrak{f}$ given in Example 25, we observe that $\mathfrak{f}$ is closed, since $\bigcap \mathfrak{f}(J) \in \mathfrak{f}(J)$ for each $J \subseteq \{a, b\}$. For instance, $\bigcap \mathfrak{f}(\{b\}) = \{d, e\}$ and, indeed, $\{d, e\} \in \mathfrak{f}(\{b\})$. $\Diamond$

Proposition 15. *Every closed I/O-specification $\mathfrak{f}$ is realized by $\mathfrak{C}(\mathfrak{f})$ under com.*

Proof. Let $J \subseteq I$. By construction of $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$, $E^* = \{z\} \cup J \cup \{y_i \mid i \in (I \setminus J)\}$ is contained in all complete extensions, while the elements of $(I \setminus J) \cup \{y_i \mid i \in J\}$ are attacked by E^* and thus by all complete extensions. All $x^{S'}_{J'}$ with $J' \neq J$ are attacked by J or some y_i with $i \in (I \setminus J)$, thus they are attacked by E^*, while all x^S_J with $S \in \mathfrak{f}(J)$ attack each other, and the other attacks they receive come from elements attacked by E^*. Two cases can then be distinguished. If $|\mathfrak{f}(J)| = 1$ then by construction of $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$

[10] Argument names such as $x^{\{c,d\}}_{\{a,b\}}$ are abbreviated by x^{cd}_{ab}.

there is just one x_J^S defended by E^*, thus the only complete extension is $E^* \cup \{x_J^S\} \cup S$. If, on the other hand, $|\mathfrak{f}(J)| > 1$, any x_J^S with $S \in \mathfrak{f}(J)$ can be included, giving rise to the complete extension $E^* \cup \{x_J^S\} \cup S$, or none of x_J^S can be included, giving rise to the complete extension $E^* \cup \bigcap \mathfrak{f}(J)$ since an x_J^S attacks all $o \in (O \setminus S)$. Taking into account that $\bigcap \mathfrak{f}(J) \in \mathfrak{f}(J)$, in both cases we have that $com(\triangleright(\mathfrak{C}(\mathfrak{f}), J))|_O = \mathfrak{f}(J)$. □

Remark 3. The attentive reader might have already noticed that, for an I/O-specification $\mathfrak{f}$ and an input $J \subseteq I$, there is not necessarily a one-to-one correspondence between $com(\triangleright(\mathfrak{C}(\mathfrak{f}), J))$ and $\mathfrak{f}(J)$. There can be more than one complete extensions of the injection of J to $\mathfrak{C}(\mathfrak{f})$ corresponding to a single output given by $\mathfrak{f}(J)$. In particular, for $S = \bigcap \mathfrak{f}(J)$, there are two distinct complete extensions $\{z\} \cup J \cup \{y_i \mid i \in (I \setminus J)\} \cup S$ and $\{z\} \cup J \cup \{y_i \mid i \in (I \setminus J)\} \cup \{x_J^S\} \cup S$. The restriction of the complete extensions to the output arguments takes care of the one-to-one correspondence to $\mathfrak{f}(J)$.

Theorem 16. *An I/O-specification $\mathfrak{f}$ is realizable under com iff $\mathfrak{f}$ is closed.*

Proof. The if-direction is a consequence of Proposition 15.

The only-if-direction follows directly by the fact that in any AF, particularly in any injection of some extension to an I/O-module, the intersection of all complete extensions is always a complete extension too. □

Proposition 16. *Every I/O-specification $\mathfrak{f}$ with $|\mathfrak{f}(J)| = 1$ for each $J \subseteq I$ is realized by $\mathfrak{C}(\mathfrak{f})$ under grd and idl.*

Proof. Let $J \subseteq I$. By construction of $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$, $E^* = \{z\} \cup J \cup \{y_i \mid i \in (I \setminus J)\}$ is contained in all complete extensions, while the elements of $(I \setminus J) \cup \{y_i \mid i \in J)\}$ are attacked by E^* and thus by all complete extensions. All $x_{J'}^{S'}$ with $J' \neq J$ are attacked by J or some y_i with $i \in (I \setminus J)$, thus they are attacked by E^*, while the unique x_J^S with $S \in \mathfrak{f}(J)$ is defended by E^*. As a consequence, there is only one complete extension, i.e. $E^* \cup \{x_J^S\} \cup S$, which is consequently also the grounded and ideal extension. The result directly follows. □

Theorem 17. *An I/O-specification $\mathfrak{f}$ is realizable under grd and idl iff $|\mathfrak{f}(J)| = 1$ for each $J \subseteq I$.*

Proof. The if-direction is a consequence of Proposition 16.

The only-if-direction follows directly by the fact that in any AF, particularly in any injection of some extension to an I/O-module, the grounded and ideal extension are uniquely defined. □

In order to compare the expressiveness of semantics in terms of I/O-realizability, we define the notion of an I/O-signature.

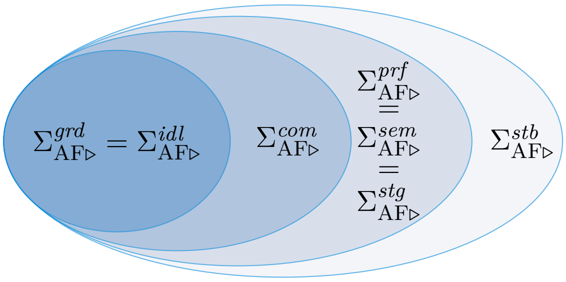


Figure 3.24: A Venn diagram illustrating the I/O-signatures of grounded, ideal, complete, semi-stable, stage, preferred, and stable semantics.

Definition 53. Let σ be a semantics. The (two-valued) I/O-signature of σ consists of all I/O-specifications that are realizable under σ:

$$\Sigma^{\sigma}_{\mathrm{AF}\triangleright} = \{\sigma_{\triangleright}(F) \mid F \text{ is an } I/O\text{-module}\},$$

where $\sigma_{\triangleright}$ is the I/O-version of σ, defined as a function mapping I/O-modules to I/O-specifications such that, given an I/O-module F, $\sigma_{\triangleright}(F) = J \subseteq I \mapsto \sigma(\triangleright(F, J))|_O$.

We summarize the presented results on I/O-realizability in the following theorem.

Theorem 18. *In accordance with Figure 3.24 it holds that*

$$\Sigma^{grd}_{AF\triangleright} = \Sigma^{idl}_{AF\triangleright} \subset \Sigma^{com}_{AF\triangleright} \subset \Sigma^{prf}_{AF\triangleright} = \Sigma^{sem}_{AF\triangleright} = \Sigma^{stg}_{AF\triangleright} \subset \Sigma^{stb}_{AF\triangleright}$$

Proof. The relations follow from Theorems 14, 15, 16, and 17. □

Note that we have disregarded I/O-realizability of admissible semantics. This is because the concept of an injection enforcing a certain acceptance status of input arguments is not applicable to admissible semantics.

3.4.2 Three-valued I/O-realizability

Until now we have dealt with realizing I/O-specifications mapping extensions to sets of extensions. As explained in Section 2.2, there are two reasons why an argument does not belong to an extension, namely either because it is attacked by the extension (i.e. it is assigned **f** in the three-valued version of the semantics) or because it is undecided due to insufficient justification (i.e. it is assigned **u**). This distinction impacts on the justification status of arguments, since attacks from undecided arguments can prevent attacked arguments from belonging to an extension, while attacks from arguments labelled **f** are ineffective. Therefore, a full description of the behaviour of a module's interaction within a larger AF has to take into account this distinction. In order to do so, we first provide a three-valued counterpart of the notions introduced in Definitions 47, 49, and 50.

First, 3-valued I/O-specifications maps labellings of input arguments to set of labellings of output arguments.

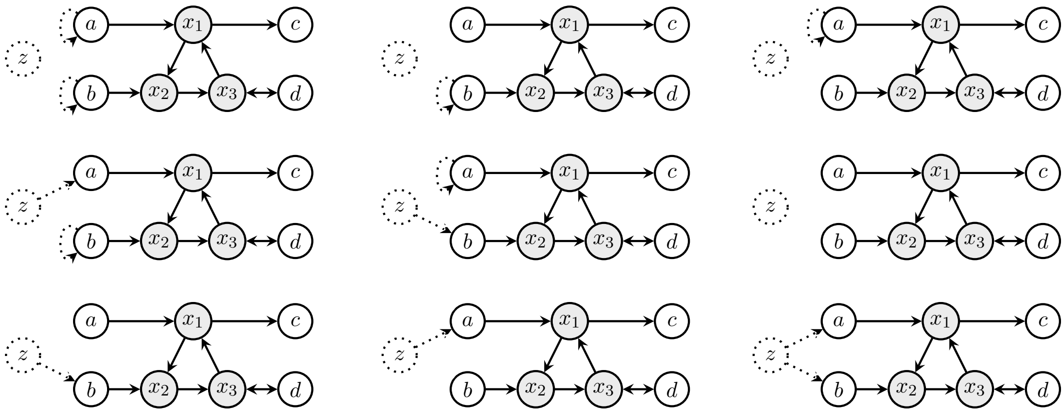


Figure 3.25: 3-valued injections to the I/O-module from Figure 3.22, as explained in Example 27.

Definition 54. A 3-*valued* I/O*-specification* consists of two sets $I, O \subseteq \mathfrak{A}$ and a total function $\mathfrak{f} : \mathcal{V}(I) \mapsto 2^{\mathcal{V}(O)}$.

The 3-valued injection now distinguishes between an input argument being **f** or **u**. In the first case, it is attacked by the new argument z and in the second case, it attacks itself and remains otherwise unattacked.

Definition 55. Given an I/O-module $F = (A, R)$ and an interpretation v over I, the 3-*valued injection* of v to F is the AF

$$\blacktriangleright(F, v) = (A \cup \{z\}, R \cup \{(z, a) \mid v(a) = \mathbf{f}\} \cup \{(b, b) \mid v(b) = \mathbf{u}\}),$$

where z is a newly introduced argument.

Definition 56. Given $I, O \subseteq \mathfrak{A}$, a semantics σ_3 and a 3-valued I/O-specification $\mathfrak{f}$, the I/O-module F *realizes* $\mathfrak{f}$ under σ_3 iff

$$\forall v \in \mathcal{V}(I) : \sigma_3(\blacktriangleright(F, v))|_O = \mathfrak{f}(v).$$

The following example illustrates these concepts.

Example 27. A possible 3-valued I/O-specification for $I = \{a, b\}$ and $O = \{c, d\}$ is the function $\mathfrak{f} : \mathcal{V}(I) \mapsto 2^{\mathcal{V}(O)}$ such that:

$$\begin{array}{lll}
\mathfrak{f}(\mathbf{uu}) = \{\mathbf{ut}\} & \mathfrak{f}(\mathbf{tu}) = \{\mathbf{tt}\} & \mathfrak{f}(\mathbf{ut}) = \{\mathbf{ut}, \mathbf{tf}\} \\
\mathfrak{f}(\mathbf{fu}) = \{\mathbf{ft}\} & \mathfrak{f}(\mathbf{uf}) = \{\mathbf{ut}\} & \mathfrak{f}(\mathbf{tt}) = \{\mathbf{tt}, \mathbf{tf}\} \\
\mathfrak{f}(\mathbf{tf}) = \{\mathbf{tt}\} & \mathfrak{f}(\mathbf{ft}) = \{\mathbf{tf}, \mathbf{ft}\} & \mathfrak{f}(\mathbf{ff}) = \{\mathbf{ft}\}
\end{array}$$

Recall our notation for interpretations: a sequence of truth values denotes the interpretation mapping the ith argument to the ith value in the sequence. That is, the interpretation $\mathbf{fu}$ for I is short for $\{a \mapsto \mathbf{f}, b \mapsto \mathbf{u}\}$.

Inspecting $\mathfrak{f}$ we observe that, for instance, setting a to $\mathbf{f}$ and b to $\mathbf{u}$ shall have the effect that c evaluates to $\mathbf{f}$ and d evaluates to $\mathbf{t}$. Setting both input arguments to $\mathbf{t}$ shall have two possible outputs, namely one where both output arguments are accepted and one with c accepted and d rejected. It turns out that the I/O-module F depicted in Figure 3.22 realizes $\mathfrak{f}$ under preferred semantics. The 3-valued injections to F are depicted in Figure 3.25. Consider, for instance, the injection of $\mathbf{fu}$, i.e. $\blacktriangleright(F, \mathbf{fu})$ (second row, left). Since $\mathit{prf}(\blacktriangleright(F, \mathbf{fu})) = \{\{z, x_1, d\}\}$ it indeed holds that $\mathit{prf}_3(\blacktriangleright(F, \mathbf{fu}))|_O = \{\mathbf{ft}\} = \mathfrak{f}(\mathbf{fu})$. $\diamondsuit$

By definition of the stable semantics it is clear that in order to be realized under stb_3, a 3-valued I/O-specification must have empty output for all inputs including an argument assigned to $\mathbf{u}$ and, as can be derived from the two-valued case, no output argument assigned to $\mathbf{u}$ for inputs with each argument assigned to $\mathbf{t}$ or $\mathbf{f}$.

Theorem 19. *A 3-valued I/O-specification $\mathfrak{f}$ is realizable under stb_3 iff for each $v \in \mathcal{V}(I)$ it holds that*

- *if $\exists i \in I : v(i) = \mathbf{u}$ then $\mathfrak{f}(v) = \emptyset$, and*
- *otherwise $w(o) \neq \mathbf{u}$ for all $w \in \mathfrak{f}(v)$ and $o \in O$.*

Proof. For the only-if-direction consider the case where $\exists i \in I : v(i) = \mathbf{u}$. Then for any I/O-module F, $\blacktriangleright(F, v)$ contains a self-attacking argument otherwise unattacked, hence $\mathit{stb}_3(\blacktriangleright(F, v)) = \emptyset$. In the other case, by definition of the stable semantics it is clear that each $o \in O$ must be assigned either $\mathbf{t}$ or $\mathbf{f}$ by any stable interpretation.

For the if-direction we get that if $\exists i \in I : v(i) = \mathbf{u}$ then $\mathit{stb}_3(\blacktriangleright(\mathfrak{C}(\mathfrak{f}), v)) = \emptyset$. Otherwise the 3-valued injection coincides with the injection from the two-valued case and the result follows from Theorem 14. $\square$

In order to characterize those 3-valued I/O-specifications which are realizable under the other semantics we need the following concept of monotonicity.

Definition 57. A 3-valued I/O-specification $\mathfrak{f}$ is *monotonic* if for all v_1 and v_2 such that $v_1 \leq_i v_2$ it holds that

$$\forall w_1 \in \mathfrak{f}(v_1)\, \exists w_2 \in \mathfrak{f}(v_2) : w_1 \leq_i w_2.$$

The intuitive meaning of monotonicity is the following: if w_1 is an output for input v_1, then for every input which is more committed than v_1 there must be an output more committed than w_1.

Example 28. Consider again the 3-valued I/O-specification $\mathfrak{f}$ from Example 27. We check whether $\mathfrak{f}$ is monotonic. Note that is suffices to check the condition for the direct successor wrt. $\leq_i$.

- For $\mathbf{ut} \in \mathfrak{f}(\mathbf{uu})$ there is $\mathbf{tt} \in \mathfrak{f}(\mathbf{tu})$, $\mathbf{ut} \in \mathfrak{f}(\mathbf{ut})$, $\mathbf{ft} \in \mathfrak{f}(\mathbf{fu})$, $\mathbf{ut} \in \mathfrak{f}(\mathbf{uf})$.
- For $\mathbf{tt} \in \mathfrak{f}(\mathbf{tu})$ there is $\mathbf{tt} \in \mathfrak{f}(\mathbf{tt})$, $\mathbf{tt} \in \mathfrak{f}(\mathbf{tf})$.
- For $\mathbf{ut} \in \mathfrak{f}(\mathbf{ut})$ there is $\mathbf{tt} \in \mathfrak{f}(\mathbf{tt})$, $\mathbf{ft} \in \mathfrak{f}(\mathbf{ft})$.
- For $\mathbf{tf} \in \mathfrak{f}(\mathbf{ut})$ there is $\mathbf{tf} \in \mathfrak{f}(\mathbf{tt})$, $\mathbf{tf} \in \mathfrak{f}(\mathbf{ft})$.
- For $\mathbf{ft} \in \mathfrak{f}(\mathbf{fu})$ there is $\mathbf{tf} \in \mathfrak{f}(\mathbf{ft})$, $\mathbf{ft} \in \mathfrak{f}(\mathbf{ff})$.
- For $\mathbf{ut} \in \mathfrak{f}(\mathbf{uf})$ there is $\mathbf{tt} \in \mathfrak{f}(\mathbf{tf})$, $\mathbf{ft} \in \mathfrak{f}(\mathbf{ff})$.

Therefore we conclude that $\mathfrak{f}$ is monotonic. ◊

Coming to necessary conditions for 3-valued I/O-specifications we start with rather obvious observations:

Proposition 17. *For every* 3*-valued I/O-specification $\mathfrak{f}$ which is realizable under grd_3, $|\mathfrak{f}(v)| = 1$ for all $v \in \mathcal{V}(I)$.*

Proof. This is immediate by the fact that $|grd_3(F)| = 1$ for every AF F. □

Proposition 18. *For every* 3*-valued I/O-specification $\mathfrak{f}$ which is realizable under prf_3, $|\mathfrak{f}(v)| \geq 1$ for all $v \in \mathcal{V}(I)$.*

Proof. This is immediate by the fact that $|prf_3(F)| \geq 1$ for every AF F. □

Monotonicity is a necessary condition for grounded and preferred semantics

Proposition 19. *Every* 3*-valued I/O-specification which is realizable under grd_3 or prf_3 is monotonic.*

Proof. Let $\mathfrak{f}$ be a 3-valued I/O-specification and suppose it is realized by the I/O-module F under grd_3. Moreover let $v_1 \leq_i v_2$ be interpretations over I.

grd_3: $\forall w_2 \in com_3(\blacktriangleright(F, v_2))|_O \; \exists w_1 \in com_3(\blacktriangleright(F, v_1))|_O : w_1 \leq_i w_2$ was shown in [17, Proposition 7]. From this and the fact that $|\mathfrak{f}(v_1)| = |\mathfrak{f}(v_2)| = 1$ (cf. Proposition 17) monotonicity for grd_3 follows.

prf_3: We know, again from [17, Proposition 7], that $\forall w_1 \in com_3(\blacktriangleright(F, v_1))|_O \; \exists w_2 \in com_3(\blacktriangleright(F, v_2))|_O : w_1 \leq_i w_2$. Now observe that each preferred interpretation is also complete and for each complete interpretation there exists preferred interpretation which is greater wrt. $\leq_i$. Hence the result for prf_3 follows. □

In Propositions 17 and 19 we have given necessary conditions for 3-valued I/O-specifications to be realizable under grd_3 and prf_3. In the following we show that these conditions are also sufficient in the sense that we can find a realizing I/O-module. The constructions of these I/O-modules will depend on the given 3-valued I/O-specification and on the semantics, but they will share the same input and output part. The semantics-specific parts, denoted by $X^\sigma_\mathfrak{f}$ and $R^\sigma_\mathfrak{f}$ in the following definition, will be given later.

Definition 58. Given a 3-valued I/O-specification $\mathfrak{f}$ we define $I' = \{i' \mid i \in I\}$, $O' = \{o' \mid o \in O\}$, $R_I = \{(i, i') \mid i \in I\}$ and $R_O = \{(o', o'), (o', o) \mid o \in O\}$. The 3-*valued canonical I/O-module* for semantics σ_3 and the 3-valued I/O-specification $\mathfrak{f}$ is defined as

$$\mathfrak{D}^\sigma_\mathfrak{f} = (I \cup I' \cup X^\sigma_\mathfrak{f} \cup O' \cup O, R_I \cup R^\sigma_\mathfrak{f} \cup R_O).$$

with $R^\sigma_\mathfrak{f} \subseteq ((I \cup I') \times X^\sigma_\mathfrak{f}) \cup (X^\sigma_\mathfrak{f} \times X^\sigma_\mathfrak{f}) \cup (X^\sigma_\mathfrak{f} \times (O' \cup O))$.

The semantics-independent part of $\mathfrak{D}^\sigma_\mathfrak{f}$ guarantees that the labelling of I coincides with the injected labelling and the labelling of I' is just the negation.

Lemma 11. *Given an arbitrary* 3*-valued I/O-specification* $\mathfrak{f}$ *and a semantics* $\sigma_3 \in \{grd_3, idl_3, com_3, prf_3, sem_3, stg_3\}$ *it holds for every* $v \in \mathcal{V}(I)$ *that*

$$\sigma_3(\blacktriangleright(\mathfrak{D}^\sigma_\mathfrak{f}, v))|_I = \{v\}, \mathit{and}$$
$$\sigma_3(\blacktriangleright(\mathfrak{D}^\sigma_\mathfrak{f}, v))|_{I'} = \{\neg v\}.$$

Proof. By the fact that arguments in $I \cup I'$ are not allowed to be attacked by the semantics-specific arguments $X^\sigma_\mathfrak{f}$, it follows that, in $\blacktriangleright(\mathfrak{D}^\sigma_\mathfrak{f}, v)$, an argument $a \in I$ is unattacked if $v(a) = \mathbf{t}$, attacked by the unattacked argument z if $v(a) = \mathbf{f}$, and self-attacking and otherwise unattacked if $v(a) = \mathbf{u}$. Hence the result for I follows. The result for I' is then immediate by the fact that each $a' \in I'$ is only attacked by $a \in I$ and therefore has the negated interpretation of a. $\square$

Now we turn to the semantics-specific constructions. For grounded semantics we need the concept of determining input interpretations. An interpretation v over the input arguments is determining for output argument o if v is a minimal (w.r.t. $\leq_i$) input interpretations where o gets a concrete value ($\mathbf{t}$ or $\mathbf{f}$) according to $\mathfrak{f}$.

With abuse of notation, in the following we may identify a set including a single interpretation with the interpretation itself.

Definition 59. Given a 3-valued I/O-specification $\mathfrak{f}$ with $|\mathfrak{f}(v)| = 1$ for all $v \in \mathcal{V}(I)$ and an argument $o \in O$, an interpretation v over I is *determining for o* (in $\mathfrak{f}$), if $\mathfrak{f}(v)(o) \neq \mathbf{u}$ and $\forall v' <_i v : \mathfrak{f}(v')(o) = \mathbf{u}$. We denote the set of interpretation which are determining for o (in $\mathfrak{f}$) as $\mathfrak{d}_\mathfrak{f}(o)$.

Note that for 3-valued I/O-specifications which are monotonic, two different interpretations which are determining for a certain output argument cannot be comparable. The following example illustrates the concept of determining interpretations.

Example 29. Let $\mathfrak{f}$ be the following 3-valued I/O-specification with $I = \{a, b\}$ and $O = \{c, d\}$:

$$\begin{array}{ccc} \mathfrak{f}(\mathbf{uu}) = \{\mathbf{uu}\} & \mathfrak{f}(\mathbf{tu}) = \{\mathbf{tu}\} & \mathfrak{f}(\mathbf{ut}) = \{\mathbf{ut}\} \\ \mathfrak{f}(\mathbf{uf}) = \{\mathbf{uf}\} & \mathfrak{f}(\mathbf{fu}) = \{\mathbf{uu}\} & \mathfrak{f}(\mathbf{tt}) = \{\mathbf{tt}\} \\ \mathfrak{f}(\mathbf{tf}) = \{\mathbf{tf}\} & \mathfrak{f}(\mathbf{ft}) = \{\mathbf{ut}\} & \mathfrak{f}(\mathbf{ff}) = \{\mathbf{tf}\} \end{array}$$

We have the following sets of determining interpretations: $\mathfrak{d}_\mathfrak{f}(c) = \{\mathbf{tu}, \mathbf{ff}\}$ and $\mathfrak{d}_\mathfrak{f}(d) = \{\mathbf{ut}, \mathbf{uf}\}$. Consider, for instance, the input interpretation $\mathbf{ff}$. We have $\mathfrak{f}(\mathbf{ff}) = \mathbf{tf}$. In order to check if $\mathbf{ff}$ is determining for c we have to look at all input interpretation being less committed than $\mathbf{ff}$. Now we observe $\mathfrak{f}(\mathbf{uf}) = \mathbf{uf}$, $\mathfrak{f}(\mathbf{fu}) = \mathfrak{f}(\mathbf{uu}) = \mathbf{uu}$. In all of these desired output interpretation c has value $\mathbf{u}$, so $\mathbf{ff}$ is determining for c. On the other hand $\mathbf{ff}$ is not determining for d, since $\mathfrak{f}(\mathbf{uf})(d) = \mathbf{f}$. $\Diamond$

The semantics-specific construction for grounded semantics uses the concept of determining interpretation and is defined as follows.

Definition 60. Given a 3-valued I/O-specification $\mathfrak{f}$ with $|\mathfrak{f}(v)| = 1$ for all $v \in \mathcal{V}(I)$, the *grd-specific part* of $\mathfrak{D}_\mathfrak{f}^{grd}$ is given by

$$\begin{aligned} X_\mathfrak{f}^{grd} =& \{x_o^v \mid o \in O, v \in \mathfrak{d}_\mathfrak{f}(o)\}, \text{ and} \\ R_\mathfrak{f}^{grd} =& \{(i, x_o^v) \mid x_o^v \in X_\mathfrak{f}^{grd}, v(i) = \mathbf{f}\} \cup \{(i', x_o^v) \mid x_o^v \in X_\mathfrak{f}^{grd}, v(i) = \mathbf{t}\} \cup \\ & \{(x_o^v, o') \mid x_o^v \in X_\mathfrak{f}^{grd}, \mathfrak{f}(v)(o) = \mathbf{t}\} \cup \{(x_o^v, o) \mid x_o^v \in X_\mathfrak{f}^{grd}, \mathfrak{f}(v)(o) = \mathbf{f}\}. \end{aligned}$$

For every $o \in O$ and each input interpretation v which is determining for o, there is the argument x_o^v. This argument can be assigned $\mathbf{t}$ if v is the labelling of I (recall Lemma 11) and intuitively enforces the labelling of o to be as given by $\mathfrak{f}(v)$.

Example 30. Again consider the 3-valued I/O-specification $\mathfrak{f}$ from Example 29. We have seen the determining interpretation there. The I/O-module $\mathfrak{D}_\mathfrak{f}^{grd}$ is depicted in Figure 3.26. It can be seen that, for every output argument o, each determining interpretation $v \in \mathfrak{d}_\mathfrak{f}(o)$ has a corresponding argument x_o^v in $\mathfrak{D}_\mathfrak{f}^{grd}$. Depending on the value of $\mathfrak{f}(v)(o)$, x_o^v either attacks argument o or o'. $\Diamond$

The next results, requiring two preliminary lemmas, characterize realizability of grounded semantics.

Lemma 12. *Let $\mathfrak{f}$ be a 3-valued I/O-specification which is monotonic and s.t. $|\mathfrak{f}(v)| = 1$ for each $v \in \mathcal{V}(I)$. Moreover let $o \in O$ and $v_1, v_2 \in \mathcal{V}(I)$ be such that $\mathfrak{f}(v_1)(o) = \mathbf{t}$ and $\mathfrak{f}(v_2)(o) = \mathbf{f}$. Then v_1 and v_2 are not compatible.*

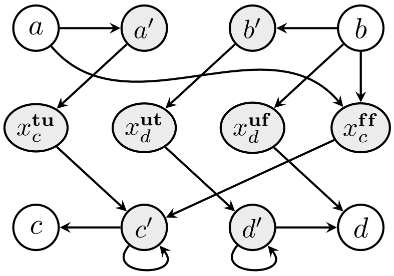


Figure 3.26: Canonical I/O-module $\mathfrak{D}_{\mathfrak{f}}^{grd}$ for the 3-valued I/O-specification $\mathfrak{f}$ given in Example 29. As discussed in Example 31, it indeed realizes $\mathfrak{f}$ under grd_3.

Proof. Towards a contradiction assume that v_1 and v_2 are compatible, and let $V \in \mathcal{V}(I)$ such that for each $i \in I$, $V(i) = \mathbf{t}$ iff $v_1(i) = \mathbf{t} \vee v_2(i) = \mathbf{t}$, $V(i) = \mathbf{f}$ iff $v_1(i) = \mathbf{f} \vee v_2(i) = \mathbf{f}$, $V(i) = \mathbf{u}$ iff $v_1(i) = v_2(i) = \mathbf{u}$. Note that V is well-defined since $v_1^{\mathbf{t}} \cap v_2^{\mathbf{f}} = v_1^{\mathbf{f}} \cap v_2^{\mathbf{t}} = \emptyset$. It holds that $v_1 \leq_i V$, thus $\mathfrak{f}(V)(o) = \mathbf{t}$ since $\mathfrak{f}$ is monotonic. However, it also holds that $v_2 \leq_i V$, thus $\mathfrak{f}(V)(o) = \mathbf{f}$, a contradiction. □

Lemma 13. *Given a* 3*-valued* I/O*-specification* $\mathfrak{f}$ *which is monotonic and s.t.* $|\mathfrak{f}(v)| = 1$ *for each* $v \in \mathcal{V}(I)$*, let* $o \in O$ *and* $v_1, v_2 \in \mathcal{V}(I)$ *be such that* v_1 *is determining for* o*. Then* $grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v_2))(x_o^{v_1})$ *is*

1. $\mathbf{t}$ *if* $v_1 \leq_i v_2$*;*
2. $\mathbf{f}$ *if* v_1 *and* v_2 *are not compatible; and*
3. $\mathbf{u}$ *if* v_1 *and* v_2 *are compatible but* $v_1 \not\leq_i v_2$*.*

Proof. Let $g = grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v_2))$ and note that, by Lemma 11, we know that $g|_I = v_2$ and $g|_{I'} = \neg v_2$.

(1) If $v_1 \leq_i v_2$ then, by construction of $\mathfrak{D}_{\mathfrak{f}}^{grd}$ and Lemma 11, all attackers of $x_o^{v_1}$ are $\mathbf{f}$ in g, hence $g(x_o^{v_1}) = \mathbf{t}$.

(2) If v_1 and v_2 are not compatible then there is some $i \in I$ such that either $v_1(i) = \mathbf{t}$ and $v_2(i) = \mathbf{f}$ or $v_1(i) = \mathbf{f}$ and $v_2(i) = \mathbf{t}$. In the first case $x_o^{v_1}$ is attacked by i' and $g(i') = \mathbf{t}$, in the second case $x_o^{v_1}$ is attacked by i and $g(i) = \mathbf{t}$, both entailing $g(x_o^{v_1}) = \mathbf{f}$.

(3) If v_1 and v_2 are compatible then, by construction of $\mathfrak{D}_{\mathfrak{f}}^{grd}$ and Lemma 11, all attackers of $x_o^{v_1}$ are either $\mathbf{f}$ or $\mathbf{u}$. Moreover, since $v_1 \not\leq_i v_2$ there is some $i \in I$ with $v_2(i) = \mathbf{u}$ and $v_1(i) \neq \mathbf{u}$. But then $g(i) = g(i') = \mathbf{u}$ and $x_o^{v_1}$ is attacked by either i or i', hence $g(x_o^{v_1}) = \mathbf{u}$. □

Proposition 20. *Every* 3*-valued* I/O*-specification* $\mathfrak{f}$ *which is monotonic and s.t.* $|\mathfrak{f}(v)| = 1$ *for each* $v \in \mathcal{V}(I)$*, is realized by* $\mathfrak{D}_{\mathfrak{f}}^{grd}$ *under* grd_3*.*

Proof. Consider some input interpretation v. We have to show $grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v))|_O = \mathfrak{f}(v)$. To this end let $o \in O$.

Assume $\mathfrak{f}(v)(o) = \mathbf{u}$. Then, since $\mathfrak{f}$ is monotonic, $\mathfrak{f}(v')(o) = \mathbf{u}$ for all $v' \leq_i v$. Therefore, there is no $v' \leq_i v$ with $v' \in \mathfrak{d}_{\mathfrak{f}}(o)$. By Lemma 13 we therefore get that for all $v'' \in \mathfrak{d}_{\mathfrak{f}}(o)$ it holds that $grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v))(x_o^{v''}) \neq \mathbf{t}$. Since, by construction of $\mathfrak{D}_{\mathfrak{f}}^{grd}$, such $x_o^{v''}$ with $v'' \in \mathfrak{d}_{\mathfrak{f}}(o)$ are the only attackers of o and o', $grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v))(o) = \mathbf{u}$.

Next assume $\mathfrak{f}(v)(o) = \mathbf{t}$. Then there is some $v' \leq_i v$ with $v' \in \mathfrak{d}_{\mathfrak{f}}(o)$ and $\mathfrak{f}(v')(o) = \mathbf{t}$. By Lemma 13 we get $grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v))(x_o^{v'}) = \mathbf{t}$. Moreover, $x_o^{v'}$ attacks o', hence $grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v))(o') = \mathbf{f}$. Towards a contradiction assume there is some $x_o^{v''}$ attacking o with $grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v))(x_o^{v''}) \in \{\mathbf{t}, \mathbf{u}\}$. Then, by Lemma 13, v'' and v are compatible and as $v' \leq_i v$ also v'' and v' are compatible. However, $x_o^{v''}$ attacking o and $x_o^{v'}$ attacking o' means, by construction of $\mathfrak{D}_{\mathfrak{f}}^{grd}$, $\mathfrak{f}(v'')(o) = \mathbf{f}$ and $\mathfrak{f}(v')(o) = \mathbf{t}$, respectively. But then, by Lemma 12, v'' and v' are not compatible, a contradiction. Therefore all attackers of o are labelled $\mathbf{f}$ by $grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v))$, hence $grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v))(o) = \mathbf{t}$.

Finally assume $\mathfrak{f}(v)(o) = \mathbf{f}$. Then there is some $v' \leq_i v$ with $v' \in \mathfrak{d}_{\mathfrak{f}}(o)$ and $\mathfrak{f}(v')(o) = \mathbf{f}$. By Lemma 13 we get $grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v))(x_o^{v'}) = \mathbf{t}$. Moreover, $x_o^{v'}$ attacks o, hence $grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v))(o) = \mathbf{f}$. □

Example 31. Again consider the 3-valued I/O-specification $\mathfrak{f}$ from Example 29. and the corresponding I/O-module $\mathfrak{D}_{\mathfrak{f}}^{grd}$ depicted in Figure 3.26. Consider, for instance, the 3-valued injection of $\mathbf{fu}$ to $\mathfrak{D}_{\mathfrak{f}}^{grd}$, which adds the additional arguments z attacking a as well as a self-attack of b to $\mathfrak{D}_{\mathfrak{f}}^{grd}$. We get $grd(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, \mathbf{fu})) = \{z, a'\}$, hence $grd_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, \mathbf{fu}))|_O = \mathbf{uu}$, being in line with $\mathfrak{f}$. One can check that this holds for all possible 3-valued injections, hence $\mathfrak{D}_{\mathfrak{f}}^{grd}$ realizes $\mathfrak{f}$ under the grounded semantics. ◊

Theorem 20. *A 3-valued I/O-specification $\mathfrak{f}$ is realizable under grd_3 iff $\mathfrak{f}$ is monotonic and for each $v \in \mathcal{V}(I)$, $|\mathfrak{f}(v)| = 1$.*

Proof. The if-direction is a direct consequence of Proposition 20. The only-if-direction follows by Propositions 17 and 19. □

Now we present the part of the 3-valued canonical I/O-module which is specific to the preferred semantics.

Definition 61. Given a 3-valued I/O-specification $\mathfrak{f}$, the *prf-specific part* of $\mathfrak{D}_{\mathfrak{f}}^{prf}$ is given by

$$\begin{aligned}
X_{\mathfrak{f}}^{prf} =& \{x_w^v \mid v \in \mathcal{V}(I), w \in \mathfrak{f}(v)\}, \text{ and} \\
R_{\mathfrak{f}}^{prf} =& \{(i, x_w^v) \mid x_w^v \in X_{\mathfrak{f}}^{prf}, v(i) = \mathbf{f}\} \cup \{(i', x_w^v) \mid x_w^v \in X_{\mathfrak{f}}^{prf}, v(i) = \mathbf{t}\} \cup \\
& \{(x_w^v, o') \mid x_w^v \in X_{\mathfrak{f}}^{prf}, w(o) = \mathbf{t}\} \cup \{(x_w^v, o) \mid x_w^v \in X_{\mathfrak{f}}^{prf}, w(o) = \mathbf{f}\} \cup \\
& \{(x_w^v, x_{w'}^{v'}) \mid \neg(v <_i v' \wedge w \leq_i w') \wedge \neg(v' <_i v \wedge w' \leq_i w)\}.
\end{aligned}$$

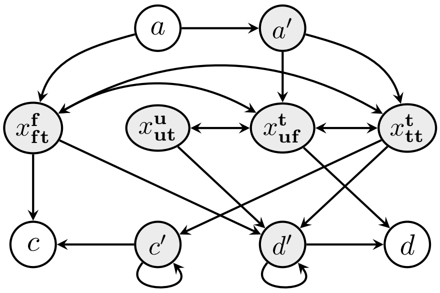


Figure 3.27: 3-valued canonical I/O-module $\mathfrak{D}^{prf}_{\mathfrak{f}}$ for the 3-valued I/O-specification $\mathfrak{f}$ given in Example 32.

Every combination of an input interpretation v and a corresponding output interpretation w is represented by an argument x^v_w in $\mathfrak{D}^{prf}_{\mathfrak{f}}$. The way the input arguments are linked to x^v_w makes sure that, with input interpretation v injected, x^v_w is not attacked by any argument among $I \cup I'$ which can be $\mathbf{t}$ in a preferred interpretation (cf. Lemma 11). Therefore each such argument x^v_w can act as a representative for a preferred interpretation, enforcing output interpretation w. The attacks among arguments $X^{prf}_{\mathfrak{f}}$ are symmetric such that x^v_w attacks all $x^{v'}_{w'}$ except those where either $v <_i v'$ and $w \leq_i w'$ or $v' <_i v$ and $w' \leq_i w$.

Example 32. Consider the 3-valued I/O-specification for $I = \{a\}$ and $O = \{c, d\}$ given by

$$\mathfrak{f}(\mathbf{u}) = \{\mathbf{ut}\} \qquad \mathfrak{f}(\mathbf{t}) = \{\mathbf{tt}, \mathbf{uf}\} \qquad \mathfrak{f}(\mathbf{f}) = \{\mathbf{ft}\}.$$

It is easy to see that $\mathfrak{f}$ is monotonic, since $\mathbf{ut} \in \mathfrak{f}(\mathbf{u})$ has a successor wrt. $\leq_i$ in both $\mathfrak{f}(\mathbf{t})$ (namely $\mathbf{ut}$) and $\mathfrak{f}(\mathbf{f})$ (namely $\mathbf{ft}$).

The AF in Figure 3.27 depicts the 3-valued canonical I/O-module $\mathfrak{D}^{prf}_{\mathfrak{f}}$. Observe the symmetric attacks between arguments $x^v_w, x^{v'}_{w'} \in X^{prf}_{\mathfrak{f}}$ whenever $v = v'$ ($x^{\mathbf{t}}_{\mathbf{uf}}$ and $x^{\mathbf{t}}_{\mathbf{tt}}$), v and v' are not comparable (e.g. $x^{\mathbf{f}}_{\mathbf{ft}}$ and $x^{\mathbf{t}}_{\mathbf{uf}}$), or $v <_i v'$ but $w \not\leq_i w'$ ($x^{\mathbf{u}}_{\mathbf{ut}}$ and $x^{\mathbf{t}}_{\mathbf{uf}}$). However, there is no attack if both $v <_i v'$ and $w \leq_i w'$ holds, as for instance between $x^{\mathbf{u}}_{\mathbf{ut}}$ and $x^{\mathbf{t}}_{\mathbf{tt}}$. To see the motivation behind this consider the injection of $\mathbf{t}$ to $\mathfrak{D}^{prf}_{\mathfrak{f}}$. We get $\mathit{prf}(\blacktriangleright(\mathfrak{D}^{prf}_{\mathfrak{f}}, \mathbf{t})) = \{\{z, a, x^{\mathbf{u}}_{\mathbf{ut}}, x^{\mathbf{t}}_{\mathbf{tt}}, c, d\}, \{z, a, x^{\mathbf{t}}_{\mathbf{uf}}\}\}$, giving rise to $\mathit{prf}_3(\blacktriangleright(\mathfrak{D}^{prf}_{\mathfrak{f}}, \mathbf{t}))|_O = \{\mathbf{tt}, \mathbf{uf}\}$, therefore realizing $\mathfrak{f}$ under prf_3. A (symmetric) attack between $x^{\mathbf{u}}_{\mathbf{ut}}$ and $x^{\mathbf{t}}_{\mathbf{tt}}$ would give $\mathbf{ut}$ as output interpretation, which is not as specified by $\mathfrak{f}(\mathbf{t})$. $\Diamond$

In the following we formally show that $\mathfrak{D}^{prf}_{\mathfrak{f}}$ realizes $\mathfrak{f}$ under preferred semantics, given that $\mathfrak{f}$ is monotonic and there is at least one output interpretation for each input interpretation. We begin with a technical lemma, giving sufficient conditions on the status of the arguments in $X^{prf}_{\mathfrak{f}}$ to get the desired labelling of the output arguments.

Lemma 14. *Given a 3-valued I/O-specification $\mathfrak{f}$ and an input interpretation $v \in \mathcal{V}(I)$, it holds for each preferred interpretation $p \in prf_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{prf}, v))$ that $p|_O = w$ for some $w \in \mathfrak{f}(v)$ if*

- *$p(x_w^v) = \mathbf{t}$ and*
- *for all $x_{w'}^{v'} \in X_{\mathfrak{f}}^{prf}$,*
  - *$w <_i w'$ implies $p(x_{w'}^{v'}) \neq \mathbf{t}$ and*
  - *w and w' being not comparable implies $p(x_{w'}^{v'}) = \mathbf{f}$.*

Proof. Consider some $w \in \mathfrak{f}(v)$ and an arbitrary $o \in O$. We show that $p(o) = w(o)$.

First assume $w(o) = \mathbf{u}$. By the hypothesis $p(x_{w'}^{v'}) \neq \mathbf{t}$ for all $w' \not\leq_i w$. Moreover, for all $w' \leq_i w$ we have that $w'(o) = \mathbf{u}$ since $w(o) = \mathbf{u}$, thus by construction of $\mathfrak{D}_{\mathfrak{f}}^{prf}$, $x_{w'}^{v'}$ attacks neither o nor o'. Summing up, neither o nor o' is attacked by an argument which is $\mathbf{t}$ in p. Hence $p(o) = \mathbf{u}$.

Next let $w(o) = \mathbf{t}$. Since $p(x_w^v) = \mathbf{t}$ we must have that $p(o') = \mathbf{f}$. Besides that, o is attacked by all $x_{w'}^{v'}$ with $w'(o) = \mathbf{f}$. But this means that w and w' are not comparable, hence $p(x_{w'}^{v'}) = \mathbf{f}$ by assumption. Now we know that all attackers of o are $\mathbf{f}$ in p, therefore $p(o) = \mathbf{t}$.

Finally let $w(o) = \mathbf{f}$. Since $p(x_w^v) = \mathbf{t}$ and x_w^v attacks o we get that $p(o) = \mathbf{f}$. □

We proceed by showing that every monotonic function $\mathfrak{f}$ assigning at least one output interpretation to each input interpretation is realized by $\mathfrak{D}_{\mathfrak{f}}^{prf}$ under the preferred semantics.

Proposition 21. *Every 3-valued I/O-specification $\mathfrak{f}$ which is monotonic and s.t. $|\mathfrak{f}(v)| \geq 1$ for each $v \in \mathcal{V}(I)$ is realized by $\mathfrak{D}_{\mathfrak{f}}^{prf}$ under prf_3.*

Proof. Consider an arbitrary input interpretation $v \in \mathcal{V}(I)$. In the following we show that $prf_3(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{prf}, v))|_O = \mathfrak{f}(v)$.

By construction of $\mathfrak{D}_{\mathfrak{f}}^{prf}$, those $x_{w'}^{v'} \in X_{\mathfrak{f}}^{prf}$ with $v' \leq_i v$ are the only arguments in $X_{\mathfrak{f}}^{prf}$ which can be $\mathbf{t}$ in a preferred interpretation of $\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{prf}, v)$, since their attackers in $I \cup I'$ are all $\mathbf{f}$, while the other arguments in $X_{\mathfrak{f}}^{prf}$ are attacked by an argument $\mathbf{t}$ of $I \cup I'$ (recall also Lemma 11). The arguments x_w^v with $w \in \mathfrak{f}(v)$ (there is at least one such argument by the hypothesis) form a clique in $\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{prf}, v)$. Moreover each of these x_w^v defends itself from all other $x_{w'}^{v'}$, hence there is a preferred interpretation of $\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{prf}, v)$ for each $w \in \mathfrak{f}(v)$ identified by x_w^v. Let p_w be the preferred interpretation with $p_w(x_w^v) = \mathbf{t}$ where $w \in \mathfrak{f}(v)$. All $x_{w'}^{v'}$ with $w' \not\leq_i w \wedge w \not\leq_i w'$ are attacked by x_v^w, hence $p_w(x_{w'}^{v'}) = \mathbf{f}$. Assume $w <_i w'$. If $v \not<_i v'$, then $x_{w'}^{v'}$ is again attacked by x_v^w and $p_w(x_{w'}^{v'}) = \mathbf{f}$. If $v <_i v'$, $p_w(x_{w'}^{v'}) \neq \mathbf{t}$

since it is attacked by some argument among $I \cup I'$ which is $\mathbf{u}$ in p_w. Therefore, by Lemma 14, $p_w|_O = w$.

It remains to show that there is no other preferred interpretation besides these p_w with $w \in \mathfrak{f}(v)$. Towards a contradiction, assume that there is a preferred interpretation p' where no x^v_w with $w \in \mathfrak{f}(v)$ is $\mathbf{t}$. By our initial considerations, those $x^{v'}_{w'}$ with $v' <_i v$ are the only arguments of $X^{prf}_{\mathfrak{f}}$ which can be $\mathbf{t}$ in p'. It cannot be the case that none of them is $\mathbf{t}$, since p' would not be incomparable to p_w with $w \in \mathfrak{f}(v)$, of which there exists at least one. Therefore there is at least one $x^{v'}_{w'}$ which is $\mathbf{t}$ in p', with $v' <_i v$, and without loss of generality we can assume that there is no $x^{v''}_{w''}$ which is $\mathbf{t}$ and $v' <_i v''$. Now, since $\mathfrak{f}$ is monotonic there has to be a $w \in \mathfrak{f}(v)$ such that $w' \leq_i w$. We prove that no argument in $X^{prf}_{\mathfrak{f}}$ attacking x^v_w is $\mathbf{t}$ in p'.

First, the only arguments in $X^{prf}_{\mathfrak{f}}$ that can be $\mathbf{t}$ are those $x^{v''}_{w''}$ with $v'' <_i v$. Note that, according to Definition 61, $x^{v'}_{w'}$ does not attack x^v_w, since $v' <_i v$ and $w' \leq_i w$. If an attacker $x^{v''}_{w''}$ is attacked in turn by $x^{v'}_{w'}$ then it is $\mathbf{f}$, otherwise either $v'' <_i v' \wedge w'' \leq_i w'$ or $v' <_i v'' \wedge w' \leq_i w''$. The first case is impossible, since we would have $v'' <_i v \wedge w'' \leq_i w$, entailing that $x^{v''}_{w''}$ does not attack x^v_w. In the other case, by the assumption on $x^{v'}_{w'}$ it holds that $x^{v''}_{w''}$ is not $\mathbf{t}$.

Now, x^v_w defends itself against all arguments in $X^{prf}_{\mathfrak{f}}$ and none of them is $\mathbf{t}$, moreover by construction of $\blacktriangleright(\mathfrak{D}^{prf}_{\mathfrak{f}}, v)$, all attackers from I and I' are $\mathbf{f}$. But then, consider the interpretation p'' obtained from p' by assigning to x^v_w the label $\mathbf{t}$, and by assigning to all the attackers of x^v_w the label $\mathbf{f}$. p'' is admissible and $p' \leq_i p''$, contradicting the maximality of p'. □

Theorem 21. *A 3-valued I/O-specification $\mathfrak{f}$ is realizable under prf_3 iff $\mathfrak{f}$ is monotonic.*

Proof. The if-direction was shown in Proposition 21 while the only-if-direction directly follows from Propositions 18 and 19. □

Comparing the capabilities of stable and preferred semantics in realizing 3-valued I/O-specifications we find that they are incomparable. While we can realize 3-valued I/O-specifications which demand the empty set of output interpretations for certain input interpretation under stb_3 but not under prf_3, we cannot realize anything involving output labels $\mathbf{u}$ under stb_3.

For the remaining semantics we have to leave the exact characterizations of realizable 3-valued I/O-specifications open. However, we can show that realizability of a 3-valued I/O-specification under grd_3 is a sufficient condition for realizability under idl_3, as well as that realizability under prf_3 is sufficient for sem_3. First we show that Proposition 20 also applies to ideal semantics since for each interpretation v the grounded interpretation of $\blacktriangleright(\mathfrak{D}^{grd}_{\mathfrak{f}}, v)$ coincides with the ideal interpretation.

Proposition 22. *Every 3-valued I/O-specification $\mathfrak{f}$ which is monotonic and s.t. $|\mathfrak{f}(v)| = 1$ for each $v \in \mathcal{V}(I)$, is realized by $\mathfrak{D}^{grd}_{\mathfrak{f}}$ under idl_3.*

Proof. Let $\mathfrak{f}$ be an 3-valued I/O-specification which is monotonic and s.t. $|\mathfrak{f}(v)| = 1$ for each $v \in \mathcal{V}(I)$. Consider some input interpretation v, and let $F = \blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{grd}, v)$. We show that $idl(F) = grd(F)$.

To this end assume, towards a contradiction, that there is some $E \in adm(F)$ such that $E \supset grd(F)$, i.e. $E \setminus grd(F) \neq \emptyset$. Let $E' = E \setminus grd(F)$. First of all, $z \in grd(F)$, hence $z \notin E'$. Next, consider some $i \in I$. If $v(i) = \mathbf{t}$ then $i \in grd(F)$, if $v(i) = \mathbf{f}$ then $i' \in grd(F)$ (cf. Lemma 11); either way, $i, i' \in grd(F)^+$, hence $i, i' \notin E'$. If $v(i) = \mathbf{u}$ then i is self-attacking and not attacked otherwise in F, hence neither i nor i' can be included in an admissible set, i.e. $i, i' \notin E'$. Now consider $x_o^{v'} \in X_{\mathfrak{f}}^{grd}$. By Lemma 13 it follows that if $v' \leq_i v$ or v and v' are not compatible, then $x_o^{v'} \in grd(F)^+$, hence $x_o^{v'} \notin E'$. If v and v' are compatible but $v' \not\leq_i v$ then $x_o^{v'} \notin grd(F)^+$, but there is some $i \in I$ with $v'(i) \neq \mathbf{u}$ and $v(i) = \mathbf{u}$. Consequently, $x_o^{v'}$ is attacked by either i or i', which, as discussed before, can both not be defended in F, hence also $x_o^{v'} \notin E'$. Finally, consider $o \in O$ and assume $o \notin grd(F)^+$. For $o \in E'$, there must be some $x \in X_{\mathfrak{f}}^{grd}$ with $x \in E'$. But this cannot be the case, as just shown.

We have shown that there is no $E \in adm(F)$ with $E \supset grd(F)$. Hence $idl(F) = grd(F)$. Since, by Proposition 20, $\blacktriangleright(grd_3(\mathfrak{D}_{\mathfrak{f}}^{grd}), v)\Big|_O = \mathfrak{f}(v)$, it follows that also $\blacktriangleright(idl_3(\mathfrak{D}_{\mathfrak{f}}^{grd}), v)\Big|_O = \mathfrak{f}(v)$. Therefore the result follows. $\square$

We cannot apply the result from Proposition 21 directly to semi-stable semantics, since it is not guaranteed that each preferred extension of the 3-valued injection of some input interpretation to $\mathfrak{D}_{\mathfrak{f}}^{prf}$ has $\subseteq$-maximal range, hence we might "lose" some elements of $\mathfrak{f}(v)$ under sem_3. To the rescue there is the well-known translation of arbitrary AFs from preferred to semi-stable semantics due to [109], which makes sure that the semi-stable extensions of the AF obtained from the translation coincide with the preferred extensions of the original AF. We use the idea of this translation to define the semi-stable specific part of the 3-valued canonical I/O-module.

Definition 62. Given a 3-valued I/O-specification $\mathfrak{f}$, the *sem*-specific part of $\mathfrak{D}_{\mathfrak{f}}^{sem}$ is given by

$$X_{\mathfrak{f}}^{sem} = \{x, x' \mid x \in X_{\mathfrak{f}}^{prf}\}, \text{ and}$$
$$R_{\mathfrak{f}}^{sem} = R_{\mathfrak{f}}^{prf} \cup \{(x, x'), (x', x), (x', x') \mid x \in X_{\mathfrak{f}}^{prf}\}$$

The idea of the translation from preferred to semi-stable semantics is to make the range of every argument incomparable to the range of every other argument, hence making also the range of preferred extensions pairwise incomparable. This is achieved by adding, for each argument a, a self-attacking argument a' which is in symmetric attack with a. Since any preferred extension of an injection to $\mathfrak{D}_{\mathfrak{f}}^{prf}$ can be uniquely identified by some $x \in X_{\mathfrak{f}}^{prf}$, it suffices to add the primed arguments for each element of $X_{\mathfrak{f}}^{prf}$.

Proposition 23. *Every 3-valued I/O-specification $\mathfrak{f}$ which is monotonic and s.t. $|\mathfrak{f}(v)| \geq 1$ for each $v \in \mathcal{V}(I)$ is realized by $\mathfrak{D}_{\mathfrak{f}}^{sem}$ under sem_3.*

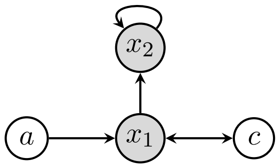


Figure 3.28: I/O-module realizing a 3-valued I/O-specification under semi-stable (resp. ideal) semantics which is not realizable under preferred (resp. grounded) semantics.

Proof. Let $\mathfrak{f}$ be a monotonic 3-valued I/O-specification. Consider an arbitrary input interpretation $v \in \mathcal{V}(I)$. Inspecting the proof of Proposition 21 we observe that $\mathit{prf}_3(\mathfrak{D}_{\mathfrak{f}}^{\mathit{prf}}) = \{p_w \mid w \in \mathfrak{f}(v)\}$, where each p_w contains x_w^v and no other $x_{w'}^v$ with $w' \neq w$. Further observe that $\mathit{prf}(\mathfrak{D}_{\mathfrak{f}}^{\mathit{prf}}) = \mathit{prf}(\mathfrak{D}_{\mathfrak{f}}^{\mathit{sem}})$. By the construction of $\mathfrak{D}_{\mathfrak{f}}^{\mathit{sem}}$ now every p_w has incomparable range by uniquely attacking ${x_w^v}'$. Therefore $\mathit{prf}(\mathfrak{D}_{\mathfrak{f}}^{\mathit{sem}}) = \mathit{sem}(\mathfrak{D}_{\mathfrak{f}}^{\mathit{sem}})$.

Having established $\mathit{prf}(\mathfrak{D}_{\mathfrak{f}}^{\mathit{prf}}) = \mathit{sem}(\mathfrak{D}_{\mathfrak{f}}^{\mathit{sem}})$ we follow from $\blacktriangleright(\mathit{prf}_3(\mathfrak{D}_{\mathfrak{f}}^{\mathit{prf}}), v)\Big|_O = \mathfrak{f}(v)$ (cf. Proposition 21) that $\blacktriangleright(\mathit{sem}_3(\mathfrak{D}_{\mathfrak{f}}^{\mathit{sem}}), v)\Big|_O = \mathfrak{f}(v)$. Therefore the result follows. $\square$

While Propositions 22 and 23 give sufficient conditions for 3-valued I/O-realizability under ideal and semi-stable semantics, respectively, they do not allow us to derive complete characterizations, since there are 3-valued I/O-specifications, which are not monotonic but realizable under ideal and semi-stable semantics, respectively. The following example illustrates this fact.

Example 33. Consider the AF F in Figure 3.28 which represents an I/O-module with $I = \{a\}$ and $O = \{c\}$. The semi-stable and ideal extensions of the various 3-valued injections are $\mathit{sem}(\blacktriangleright(F, \mathbf{u})) = \mathit{idl}(\blacktriangleright(F, \mathbf{u})) = \{\{c\}\}$, $\mathit{sem}(\blacktriangleright(F, \mathbf{t})) = \mathit{idl}(\blacktriangleright(F, \mathbf{t})) = \{\{a, c\}\}$, and $\mathit{sem}(\blacktriangleright(F, \mathbf{f})) = \{\{z, x_1\}\} \neq \mathit{idl}(\blacktriangleright(F, \mathbf{f})) = \{\{z\}\}$. This means that F realizes the 3-valued I/O-specification given by $\mathbf{u} \mapsto \{\mathbf{t}\}$, $\mathbf{t} \mapsto \{\mathbf{t}\}$, and $\mathbf{f} \mapsto \{\mathbf{u}\}$ under ideal semantics, and the one given by $\mathbf{u} \mapsto \{\mathbf{t}\}$, $\mathbf{t} \mapsto \{\mathbf{t}\}$, and $\mathbf{f} \mapsto \{\mathbf{f}\}$ under semi-stable semantics which are both clearly not monotonic. $\diamondsuit$

An exact characterization of 3-valued I/O-specifications which are realizable under semi-stable and ideal semantics, respectively, therefore requires weaker notions of monotonicity. Complete semantics, on the other hand, imposes necessary conditions which are more restrictive. The following is a direct consequence of [17, Proposition 7].

Proposition 24. *Every 3-valued I/O-specification $\mathfrak{f}$ which is realizable under com_3 is monotonic and for all v_1 and v_2 such that $v_1 \leq_i v_2$ it holds that $\forall w_2 \in \mathfrak{f}(v_2) \exists w_1 \in \mathfrak{f}(v_1) : w_1 \leq_i w_2$.*

Example 34. Once more consider the 3-valued I/O-specification $\mathfrak{f}$ from Example 27. We have seen that $\mathfrak{f}$ is monotonic and therefore realizable under prf_3 in Example 28. It is, however, not realizable under complete semantics. To see this let $v_1 = \mathbf{tu}$ and $v_2 = \mathbf{tt}$.

We have $w_2 = \mathbf{tf} \in \mathfrak{f}(v_2)$ but there is no $w_1 \in \mathfrak{f}(v_2)$ such that $w_1 \leq_i w_2$. Therefore the condition given in Proposition 24 is violated and $\mathfrak{f}$ is not realizable under com_3. $\Diamond$

Exact characterizations of realizable 3-valued I/O-specifications for complete, semi-stable and ideal semantics are subject of future work. With the following definition of the three-valued I/O-signature we can state some relations between semantics.

Definition 63. Let σ be a semantics. The three-valued I/O-signature of σ_3 consists of all 3-valued I/O-specifications that are realizable under σ_3:

$$\Sigma^{\sigma_3}_{\mathrm{AF}_\blacktriangleright} = \{\sigma_\blacktriangleright(F) \mid F \text{ is an } I/O\text{-module}\},$$

where $\sigma_\blacktriangleright$ is the three-valued I/O-version of σ, defined as a function mapping I/O-modules to 3-valued I/O-specifications such that, given an I/O-module F, $\sigma_\blacktriangleright(F) = v \in \mathcal{V}(I) \mapsto \sigma_3(\blacktriangleright(F, v))|_O$.

Theorem 22. *The following relations hold:*

1. $\Sigma^{grd_3}_{AF_\blacktriangleright} \subset \Sigma^{com_3}_{AF_\blacktriangleright} \subset \Sigma^{prf_3}_{AF_\blacktriangleright} \subset \Sigma^{sem_3}_{AF_\blacktriangleright}$*;*
2. $\Sigma^{grd_3}_{AF_\blacktriangleright} \subset \Sigma^{idl_3}_{AF_\blacktriangleright}$*;*
3. $\Sigma^{idl_3}_{AF_\blacktriangleright} \setminus \Sigma^{prf_3}_{AF_\blacktriangleright} \neq \emptyset$ *and* $\Sigma^{prf_3}_{AF_\blacktriangleright} \setminus \Sigma^{idl_3}_{AF_\blacktriangleright} \neq \emptyset$*.*

Proof. (1) $\Sigma^{grd_3}_{\mathrm{AF}_\blacktriangleright} \subseteq \Sigma^{com_3}_{\mathrm{AF}_\blacktriangleright}$ holds by the observation that, a 3-valued I/O-specification $\mathfrak{f}$ which is monotonic and has $|\mathfrak{f}(v)| = 1$ is realized by $\mathfrak{D}^{grd}_{\mathfrak{f}}$ also under complete semantics, since $grd_3(\blacktriangleright(\mathfrak{D}^{grd}_{\mathfrak{f}}, v)) = com_3(\blacktriangleright(\mathfrak{D}^{grd}_{\mathfrak{f}}, v))$. This can be read off the proof of Proposition 22, where it is shown that there is no $E \in adm(\blacktriangleright(\mathfrak{D}^{grd}_{\mathfrak{f}}, v))$ with $E \supset grd(\blacktriangleright(\mathfrak{D}^{grd}_{\mathfrak{f}}, v))$. Properness is, for instance, by the I/O-module $F = (\{a, b, c\}, \{(a, b), (b, a), (b, b), (b, c)\})$ with $I = \{a\}$ and $O = \{c\}$, which has $com_3(\blacktriangleright(F, \mathbf{u}))|_O = \{\mathbf{u}\}$, $com_3(\blacktriangleright(F, \mathbf{t}))|_O = \{\mathbf{u}, \mathbf{t}\}$, and $com_3(\blacktriangleright(F, \mathbf{f}))|_O = \{\mathbf{u}\}$.

$\Sigma^{com_3}_{\mathrm{AF}_\blacktriangleright} \subseteq \Sigma^{prf_3}_{\mathrm{AF}_\blacktriangleright}$ is by Theorem 21 and Proposition 24. Properness of the relation was discussed in Example 34.

Finally, $\Sigma^{prf_3}_{\mathrm{AF}_\blacktriangleright} \subseteq \Sigma^{sem_3}_{\mathrm{AF}_\blacktriangleright}$ is by Theorem 21 and Proposition 23, while properness was shown in Example 33.

(2) $\Sigma^{grd_3}_{\mathrm{AF}_\blacktriangleright} \subseteq \Sigma^{idl_3}_{\mathrm{AF}_\blacktriangleright}$ follows from Theorem 20 and Proposition 22. Properness of the relation was shown in Example 33.

(3) $\Sigma^{idl_3}_{\mathrm{AF}_\blacktriangleright} \setminus \Sigma^{prf_3}_{\mathrm{AF}_\blacktriangleright} \neq \emptyset$ is by the I/O-module in Figure 3.28, which realizes a 3-valued I/O-specification under idl_3 which is not monotonic and therefore not realizable under prf_3 (cf. Example 33). $\Sigma^{prf_3}_{\mathrm{AF}_\blacktriangleright} \setminus \Sigma^{idl_3}_{\mathrm{AF}_\blacktriangleright} \neq \emptyset$ is, for instance, by the 3-valued I/O-specification discussed in Example 32, which has more than one output interpretation assigned to the input interpretation $\mathbf{t}$. $\square$

3.4.3 Partial I/O-specifications

Until now we have restricted our considerations to total I/O-specifications, where the output is defined for each input. It is however natural to think of situations where we do not care about the output for some inputs, i.e. where we are only interested in realizability of a partial function.

Definition 64. A *partial* 2*-valued (resp.* 3*-valued) I/O-specification* consists of two sets $I, O \subseteq \mathfrak{A}$ and a partial function $\mathfrak{f} : 2^I \mapsto 2^{2^O}$ (resp. $\mathfrak{f} : \mathcal{V}(I) \mapsto 2^{\mathcal{V}(O)}$).

An I/O-module F realizes $\mathfrak{f}$ under a semantics σ iff for all $J \subseteq I$ (resp. $v \in \mathcal{V}(I)$) such that $\mathfrak{f}$ is defined for J (resp. v), $\sigma(\triangleright(F, J))|_O = \mathfrak{f}(J)$ (resp. $\sigma_3(\blacktriangleright(F, v))|_O = \mathfrak{f}(v)$).

The results provided in Theorems 14, 15, 16, and 17 for the 2-valued case can be directly exploited to handle partial I/O-specifications in the 2-valued case: "don't care"-outputs can be assigned arbitrarily, provided that at least one extension is assigned for *prf*, *sem* and *stg*, a single extension for *grd* and *idl*, and the specification is closed for *com*. This is because conditions for realizability contain no dependencies between outputs for different inputs. Furthermore, all the proofs also work with a partial I/O-specification by considering only the specified inputs in the definition of the canonical I/O-module, i.e. neglecting the inputs with undefined output, yielding a considerable simplification.

Corollary 4. *A partial* 2*-valued I/O-specification is realizable iff for each $J \subseteq I$ such that $\mathfrak{f}$ is defined for J it holds that*

- *stb:* $\top$*;*
- *prf, sem, stg:* $\mathfrak{f}(J) \neq \emptyset$*;*
- *com:* $\mathfrak{f}(J) \neq \emptyset$ *and* $\bigcap \mathfrak{f}(J) \in \mathfrak{f}(J)$*;*
- *grd, idl:* $|\mathfrak{f}(J)| = 1$.

On the other hand, the following example shows some difficulties in the three-valued case.

Example 35. Consider the partial 3-valued I/O-specification $\mathfrak{f}$ for $I = \{a, b\}$ and $O = \{c\}$ with $\mathfrak{f}(\mathbf{uu}) = \{\mathbf{u}\}$, $\mathfrak{f}(\mathbf{tu}) = \{\mathbf{t}\}$ and undefined, i.e. "don't care", for all other inputs. Clearly, $\mathfrak{f}$ is realized by $\mathfrak{D}_{\mathfrak{f}}^{grd}$ or, more easily, by the simple I/O-module $(\{a, b, x, c\}, \{(a, x), (x, c)\})$. Now note that $\mathfrak{f}$ is not monotonic according to Definition 57, since there is no $w \in \mathfrak{f}(\mathbf{tt})$ with $\mathbf{t} \leq_i w$. It is monotonic for those inputs for which it is defined though.

The same can be observed if we consider $\mathfrak{f}'$ which coincides with $\mathfrak{f}$ on inputs $\mathbf{uu}$ and $\mathbf{tu}$ but also defines $\mathfrak{f}'(\mathbf{ut}) = \{\mathbf{f}\}$. Now one can check that there is no I/O-module realizing $\mathfrak{f}'$ under the grounded semantics. The reason for this is that $\mathfrak{f}$ cannot be extended to a total 3-valued I/O-specification $\mathfrak{f}''$ which is still monotonic. In order to be monotonic $\mathfrak{f}''$ extending $\mathfrak{f}'$ would have to fulfill both $\mathbf{t} \leq_i w$ and $\mathbf{f} \leq_i w$ for the unique output $w \in \mathfrak{f}''(\mathbf{tt})$, which is obviously not possible. $\diamondsuit$

This already leads us to the condition for realizability of partial functions, which we state after formally defining what it means to extend a 3-valued I/O-specification.

Definition 65. Given two (partial) 3-valued I/O-specifications $\mathfrak{f}$ and $\mathfrak{f}'$, we say that $\mathfrak{f}'$ *extends* $\mathfrak{f}$ iff for all $v \in \mathcal{V}(I)$ such that $\mathfrak{f}(v)$ is defined, $\mathfrak{f}'(v) = \mathfrak{f}(v)$.

Theorem 23. *A (partial) 3-valued I/O-specification $\mathfrak{f}$ is realizable under semantics σ_3 iff there is a total function $\mathfrak{f}'$ extending $\mathfrak{f}$ which is realizable under σ_3.*

Proof. If $\mathfrak{f}'$ is realized by some I/O-module F then F also realizes $\mathfrak{f}$ since $\mathfrak{f}'(v) = \mathfrak{f}(v)$ for all $v \in \mathcal{V}(I)$ such that $\mathfrak{f}(v)$ is defined. If, on the other hand, $\mathfrak{f}$ is realized by some I/O-module F' then F' obviously also realizes some total 3-valued I/O-specification $\mathfrak{f}'$ which coincides with $\mathfrak{f}$ on those input interpretations which are defined by $\mathfrak{f}$, i.e. $\mathfrak{f}'$ extends $\mathfrak{f}$. $\square$

It may be noted that the extension-based case can be viewed as a particular case of 3-valued partial specification where also the output is partially specified, i.e. for those inputs without undecided arguments we specify a set of extensions (i.e. without distinguishing between $\mathbf{f}$ and $\mathbf{u}$ arguments).

Another relation can be drawn between three-valued I/O-realizability of stable and preferred semantics. In order to be realizable under stable semantics, a 3-valued I/O-specification $\mathfrak{f}$ has to have $\mathfrak{f}(v) = \emptyset$ for each $v \in \mathcal{V}(I)$ where $\exists i \in I : v(i) = \mathbf{u}$. Interpreting the desired output $\emptyset$ as "don't care", we can realize any 3-valued I/O-specification realizable under stable semantics also under preferred semantics.

Proposition 25. *Given a 3-valued I/O-specification $\mathfrak{f}$ which is realizable under stb_3, let $\mathfrak{f}'$ be the partial 3-valued I/O-specification with $\mathfrak{f}'(v) = \mathfrak{f}(v)$ if $\mathfrak{f}(v) \neq \emptyset$ and $\mathfrak{f}'(v)$ undefined if $\mathfrak{f}(v) = \emptyset$. It holds that $\mathfrak{f}'$ is realizable under prf_3.*

Proof. First observe that $\mathfrak{f}(v) \neq \emptyset$, and therefore $\mathfrak{f}'(v)$ defined, only if $\nexists i \in I : v(i) = \mathbf{u}$. Let $\mathfrak{f}''$ be the 3-valued I/O-specification which has $\mathfrak{f}''(v) = \mathfrak{f}'(v)$ whenever $\mathfrak{f}'(v)$ is defined and $\mathfrak{f}''(v) = v_{\mathbf{u}}$, i.e. the interpretation mapping all arguments to $\mathbf{u}$, otherwise. Consider $v_1, v_2 \in \mathcal{V}(I)$ with $v_1 \leq_i v_2$. From the first observation it follows that $\mathfrak{f}''(v_1) = \{v_{\mathbf{u}}\}$, hence every $w_2 \in \mathfrak{f}''(v_2)$ it holds that $w_1 \leq_i w_2$ for all $w_1 \in \mathfrak{f}''(v_1)$ (i.e. for $v_{\mathbf{u}}$). Since, by definition, $\mathfrak{f}''(v) \neq \emptyset$ for each $v \in \mathcal{V}(I)$, monotonicity of $\mathfrak{f}''$ follows. Hence $\mathfrak{f}''$ is realizable under prf_3 and, by Theorem 23, also $\mathfrak{f}'$ is realizable under prf_3. $\square$

3.5 Realizability in ADFs

In this section we present results on realizability in ADFs. We do so by reviewing the works of Strass [187, 188] and Pührer [175] and subsequently combining their results to establish the relation between semantics in terms of expressiveness. Due to the fact that the majority of ADF semantics is three-valued,[11] we also take a three-valued viewpoint and consider sets of interpretations, i.e. subsets of the collection of all three-valued interpretation $\mathcal{V}$, as the objects of interest.

We begin by reviewing some properties of interpretation-sets defined in [175]. The first defines what it means for an interpretation to be adm-induced by a given set of interpretations.

Definition 66. Let $V \subseteq \mathcal{V}$ be a set of interpretations and $v \in \mathcal{V}$ an interpretation. v is *adm-induced* by V if for every $a \in (v^{\mathbf{t}} \cup v^{\mathbf{f}})$ and every $v_2 \in [v]_2$ there is some $v' \in V$ such that $v' \leq_i v_2$ and $v'(a) = v(a)$. We denote the set of all interpretations which are adm-induced by V by $cl(V)$.

The intuition behind an interpretation v being adm-induced by an interpretation-set V is that in order to make all interpretations in V admissible in an ADF, also v has to be admissible in that ADF. It always holds that every interpretation $v \in V$ is adm-induced by V, i.e. $V \subseteq cl(V)$.

Example 36. Let the interpretation-set $V = \{\mathbf{tu}, \mathbf{uf}\}$ be given over arguments $\{a, b\}$. First consider the interpretation $v = \mathbf{uu}$. It is easy to see that any condition holds for every argument in $v^{\mathbf{t}} \cup v^{\mathbf{f}}$ (quantification over the empty set), hence $\mathbf{uu}$ is adm-induced by V. Next consider $v = \mathbf{tf}$. We have to check for every argument in $s \in (v^{\mathbf{t}} \cup v^{\mathbf{f}}) = \{a, b\}$ and every two-valued interpretation $v_2 \in [v]_2 = \{v\}$ that there exists a $v' \in V$ with $v' \leq_i v_2$ and $v'(s) = v(s)$. Indeed, for a we have $v' = \mathbf{tu}$ with $v' \leq_i v$ and $v'(a) = v(a) = \mathbf{t}$ and for b we have $v' = \mathbf{uf}$ with $v' \leq_i v$ and $v'(b) = v(b) = \mathbf{f}$. Hence also $\mathbf{tf}$ is adm-induced by V. As it turns out, no further interpretations are *adm*-induced by V. We conclude that $cl(V) = \{\mathbf{uu}, \mathbf{tu}, \mathbf{uf}, \mathbf{tf}\}$ ◊

The following concept of a *com*-characterization, also introduced in [175], is used, as the name suggests, to characterize realizability under complete semantics. It will be generalized and used in Section 3.6.

Definition 67. Let $V \subseteq \mathcal{V}$ be a set of interpretations. A *com-characterization* of V is a function $f : \mathcal{V}_2 \mapsto \mathcal{V}_2$ such that: for each $v \in \mathcal{V}$ we have $v \in V$ iff for each $a \in A$:

- $v(a) \neq \mathbf{u}$ implies $f(v_2)(a) = v(a)$ for all $v_2 \in [v]_2$ and
- $v(a) = \mathbf{u}$ implies $f(v_2')(a) = \mathbf{t}$ and $f(v_2'')(a) = \mathbf{f}$ for some $v_2', v_2'' \in [v]_2$.

[11] Various two-valued versions have been studied by Polberg [167, 168].

Intuitively, a *com*-characterization f assigns the Boolean value $f(v)(a)$ to an argument a that the acceptance condition of a would evaluate to under interpretation v in an ADF that has V as its complete interpretations. The following example illustrates *com*-characterizations.

Example 37. Consider the set of interpretations $V = \{\mathbf{uu}, \mathbf{uf}, \mathbf{tt}\}$ over arguments $\{a, b\}$ and the following function $f : \mathcal{V}_2 \mapsto \mathcal{V}_2$.

v	$\mathbf{tt}$	$\mathbf{tf}$	$\mathbf{ft}$	$\mathbf{ff}$
$f(v)$	$\mathbf{tt}$	$\mathbf{ff}$	$\mathbf{tf}$	$\mathbf{tf}$

We argue that f is indeed a *com*-characterization of V. A two-valued interpretation like $\mathbf{tt}$ being contained in V immediately determines the value of f, i.e. $f(\mathbf{tt}) = \mathbf{tt}$. If, on the other hand, a two-valued interpretation v_2 is not contained in V, then $f(v_2)$ must be different from v_2 for at least one argument, which is the case for all $v_2 \in \{\mathbf{tf}, \mathbf{ft}, \mathbf{ff}\}$. Now consider the interpretation $v = \mathbf{uf} \in V$. By $v(a) = \mathbf{u}$ we must have $f(\mathbf{tf})(a) \neq f(\mathbf{ff})(a)$, which is indeed the case as $f(\mathbf{tf})(a) = \mathbf{f}$ and $f(\mathbf{ff})(a) = \mathbf{t}$. Moreover $v(b) = \mathbf{f}$ gives rise to $f(\mathbf{tf})(b) = f(\mathbf{ff})(b) = \mathbf{f}$. As an example for an interpretation which is neither contained in V nor two-valued, consider $v' = \mathbf{tu} \notin V$. At least one of the implications of Definition 67 must be violated for v'. In our case, the first one fails since $f(\mathbf{tf})(a) = \mathbf{f}$, while the second one is actually fulfilled by $f(\mathbf{tt})(b) = \mathbf{t}$ and $f(\mathbf{tf})(b) = \mathbf{f}$. One of the implications is also violated for $\mathbf{ut}$ and $\mathbf{fu}$, respectively. Finally, neither all $f(v_2)(a)$ nor all $f(v_2)(b)$ coincide for every $v_2 \in \mathcal{V}_2$, hence f is in line with the fact that $\mathbf{uu} \in V$. $\Diamond$

As it turns out, the existence of a *com*-characterization is necessary and sufficient for realizability of a given set of interpretations. We recall the canonical ADF from [175], which uses a *com*-characterization for realizing a set of interpretations under the complete semantics. We will make use of this construction in the unifying approach presented in Section 3.6.

Definition 68. Given a function $f : \mathcal{V}_2 \mapsto \mathcal{V}_2$, we define the ADF $D_f = \{\langle a, \varphi_a^f \rangle \mid a \in A\}$ where the acceptance condition for each $a \in A$ is given as

$$\varphi_a^f = \bigvee_{w \in \mathcal{V}_2,\, f(w)(a)=\mathbf{t}} \phi_w \quad \text{with} \quad \phi_w = \bigwedge_{w(a')=\mathbf{t}} a' \wedge \bigwedge_{w(a')=\mathbf{f}} \neg a'$$

The idea of the construction is that whenever a is true in the two-valued interpretation assigned by f to the two-valued interpretation w, i.e. $f(w)(a) = \mathbf{t}$, then φ_a^f shall evaluate to true under w. This is achieved by putting the sub-formula ϕ_w in the big disjunction of φ_a^f. If the function f is a *com*-characterization of a given set of interpretations $V \subseteq \mathcal{V}$ according to Definition 67, then it holds that $com_3(D_f) = V$. Moreover, each ADF gives rise to a *com*-characterization.

The following result states the correspondence between *com*-characterizations and realizability more formally.

Proposition 26 ([175])**.** *Let $V \subseteq \mathcal{V}$ be a set of interpretations.*

1. *For each ADF D with $com_3(D) = V$, there is a com-characterization f_D of V.*
2. *For each com-characterization $f : \mathcal{V}_2 \mapsto \mathcal{V}_2$ of V it holds that $com_3(D_f) = V$.*

The use of *com*-characterizations in the canonical construction given in Definition 68 is illustrated in the following example.

Example 38. Again consider the function f given in Example 37. The fact that V is realizable under the complete semantics is witnessed by f being a *com*-characterization of V. The realizing ADF D_f is now given by the acceptance conditions

$$\varphi_a^f = (a \wedge b) \vee (\neg a \wedge b) \vee (\neg a \wedge \neg b) \equiv \neg a \vee b, \text{ and}$$
$$\varphi_b^f = a \wedge b$$

Now one can check that, in accordance with Proposition 26, it holds that $com_3(D_f) = V$ since f is a *com*-characterization of V. ♢

We are now ready to characterize the signatures of ADF semantics.

Theorem 24 ([188, 175])**.** *The following holds:*

$$\begin{aligned}
\Sigma_{ADF}^{mod_3} &= \{V \subseteq \mathcal{V}_2\} \\
\Sigma_{ADF}^{stb_3} &= \{V \subseteq \mathcal{V}_2 \mid V^{\mathbf{t}} \textit{ is incomparable}\} \\
\Sigma_{ADF}^{adm_3} &= \{V \subseteq \mathcal{V} \mid cl(V) = V\} \\
\Sigma_{ADF}^{com_3} &= \{V \subseteq \mathcal{V} \mid \textit{there is a com-characterization of } V\} \\
\Sigma_{ADF}^{prf_3} &= \{V \subseteq \mathcal{V} \mid V \neq \emptyset,\ V \textit{ is incompatible}\} \\
\Sigma_{ADF}^{grd_3} &= \{V \subseteq \mathcal{V} \mid |V| = 1\}
\end{aligned}$$

Note that the condition that $V \neq \emptyset$ is only stated explicitly for prf_3, but also holds implicitly for adm_3, com_3, and grd_3. It does not hold for mod_3 and stb_3, as these semantics can realize the empty set of interpretations.

Strass [188] and Pührer [175] also provide, sometimes implicitly, ways of constructing concrete ADFs realizing a given set of interpretations contained in the signature. Besides the concrete realization under complete semantics given in Definition 68, we use $\rho_{\sigma_3}^{\text{ADF}}$ to refer to an arbitrary realization function for ADFs.

Definition 69. Given a semantics σ_3, the ADF realizing function $\rho_{\sigma_3}^{\text{ADF}} : \mathcal{V} \mapsto AF_{\mathfrak{A}}$ maps interpretation-sets to ADFs such that

- $\rho_{\sigma_3}^{\text{ADF}}(V) = D$ with $\sigma_3(D) = V$ if $V \in \Sigma_{\text{ADF}}^{\sigma_3}$, and

- $\rho^{\mathrm{ADF}}_{\sigma_3}(V) = D$ with $\sigma_3(D) = \{v_{\mathbf{u}}\}$ otherwise.

The characterizations of signatures allow us to state some of their relations to each other.

Theorem 25. *The following relations hold:*

1. $\Sigma^{stb_3}_{ADF} \subset \Sigma^{mod_3}_{ADF} \subset (\Sigma^{prf_3}_{ADF} \cup \{\emptyset\})$;
2. $\Sigma^{adm_3}_{ADF} \cap \Sigma^{\sigma_3}_{ADF} = \emptyset$ *for* $\sigma_3 \in \{mod_3, stb_3\}$;
3. $\Sigma^{com_3}_{ADF} \cap \Sigma^{\sigma_3}_{ADF} = \{V \subseteq \mathcal{V}_2 \mid |V| = 1\}$ *for* $\sigma_3 \in \{mod_3, stb_3\}$;
4. $\Sigma^{adm_3}_{ADF} \cap \Sigma^{prf_3}_{ADF} = \{\{v_{\mathbf{u}}\}\}$;
5. $\Sigma^{com_3}_{ADF} \cap \Sigma^{prf_3}_{ADF} = \Sigma^{grd_3}_{ADF}$.

Proof. (1) $\Sigma^{stb_3}_{\mathrm{ADF}} \subseteq \Sigma^{mod_3}_{\mathrm{ADF}}$ is immediate. Properness is by the interpretation-set $\{\mathbf{t}, \mathbf{f}\}$ which is contained in $\Sigma^{mod_3}_{\mathrm{ADF}}$ (the realizing ADF is $\{\langle a, a\rangle\}$, i.e. the ADF with one self-supporting argument) but not in $\Sigma^{stb_3}_{\mathrm{ADF}}$, as $\{\mathbf{t}, \mathbf{f}\}^{\mathbf{t}} = \{\{a\}, \emptyset\}$ is not incomparable. Moreover, $\Sigma^{mod_3}_{\mathrm{ADF}} \subset (\Sigma^{prf_3}_{\mathrm{ADF}} \cup \{\emptyset\})$ is by the observation that a set of two-valued interpretations $V \subseteq \mathcal{V}_2$ is always incompatible (cf. Lemma 1) and the fact that any set containing an interpretation which is not two-valued cannot be contained in $\Sigma^{mod_3}_{\mathrm{ADF}}$.

(2) The result follows from the fact that $v_{\mathbf{u}} \in V$ for every $V \in \Sigma^{adm_3}_{\mathrm{ADF}}$, which is clearly not two-valued.

(3) Every $V \subseteq \mathcal{V}_2$ with $|V| = 1$ is contained in $\Sigma^{\sigma_3}_{\mathrm{ADF}}$, since $V^{\mathbf{t}}$ is incomparable, and also contained in $\Sigma^{com_3}_{\mathrm{ADF}}$, since there exists some *com*-characterization of V by [175, Lemma 2]. Assume some $V \subseteq \mathcal{V}_2$ with $|V| > 1$ and $V^{\mathbf{t}}$ being incomparable (hence contained in $\Sigma^{\sigma_3}_{\mathrm{ADF}}$). Let $v_1, v_2 \in V$ with $v_1 \neq v_2$. Further assume $V \in \Sigma^{com_3}_{\mathrm{ADF}}$, i.e. there is some ADF D with $com_3(D) = V$. Since the complete interpretations of V form a complete meet-semilattice (cf. [55, Theorem 1]) there must be some $v \in V$ with $v \leq_i (v_1 \sqcap v_2)$, a contradiction to $V \subseteq \mathcal{V}_2$. Hence $V \notin \Sigma^{com_3}_{\mathrm{ADF}}$ and the result follows.

(4) The interpretation $v_{\mathbf{u}}$ is induced by every set of interpretations, hence $v_{\mathbf{u}} \in V$ for every $V \in \Sigma^{adm_3}_{\mathrm{ADF}}$. Since $v_{\mathbf{u}} \leq_i v$ for every other $v \in \mathcal{V}$, $v \notin V$ in order to have $V \in \Sigma^{prf_3}_{\mathrm{ADF}}$. Hence the result follows.

(5) Every $V \subseteq \mathcal{V}$ with $|V| = 1$ is incompatible and there exists some *com*-characterization of V, hence the $\supseteq$-direction holds. To show the $\subseteq$-direction, assume some $V \subseteq \mathcal{V}$ with $|V| > 1$ which is incompatible (hence contained in $\Sigma^{prf_3}_{\mathrm{ADF}}$). Let $v_1, v_2 \in V$ with $v_1 \neq v_2$. Further assume, towards a contradiction, that $V \in \Sigma^{com_3}_{\mathrm{ADF}}$, i.e. there is some ADF D with $com_3(D) = V$. Since the complete interpretations of V form a complete meet-semilattice (cf. [55, Theorem 1]) there must be some $v \in V$ with $v \leq_i (v_1 \sqcap v_2)$, a contradiction to V being incompatible. Hence $V \notin \Sigma^{com_3}_{\mathrm{ADF}}$ and the result follows. □

We will gain a few more insights, in particular about the relation between $\Sigma_{\mathrm{ADF}}^{adm_3}$ and $\Sigma_{\mathrm{ADF}}^{com_3}$, in Section 3.6.

Also for ADFs we are interested in the closure of signatures under subset and intersection. Some of the following results will be of use in Section 4.3.

Proposition 27. *Given an arbitrary ADF D it holds that*

- *for any $V \subseteq prf_3(D)$ with $V \neq \emptyset$ there exists an ADF D' such that $prf_3(D') = V$;*
- *for any $V \subseteq \sigma_3(D)$ there exists an ADF D' such that $\sigma_3(D') = V$ if $\sigma_3 \in \{mod_3, stb_3\}$.*

Proof. Consider some $V \subseteq \mathcal{V}$ and let $V' \subseteq V$. If $V \subseteq \mathcal{V}_2$ then also $V' \subseteq \mathcal{V}_2$. Since $V^{\mathbf{t}} \subseteq V'^{\mathbf{t}}$ it follows that if $V^{\mathbf{t}}$ is incomparable then $V'^{\mathbf{t}}$ is incomparable. Finally, if V is incompatible, then all pairs of elements $v_1, v_2 \in V$ are incompatible, hence all pairs of elements $v_1, v_2 \in V'$ are incompatible, i.e. V' is incompatible. The result for prf_3, mod_3, and stb_3 now follows. $\square$

Admissible and complete semantics are not closed under subset. This can be seen by the AF $D = \{\langle a, a\rangle\}$, which has $com_3(D) = adm_3(D) = \{\mathbf{u}, \mathbf{t}, \mathbf{f}\}$, but $\{\mathbf{t}, \mathbf{f}\} \subseteq com_3(D)$ is realizable neither under admissible nor under complete semantics in ADFs. Moreover, the question is only of limited interest for grounded semantics, as the only non-empty subset of an interpretation-set obtained from grounded semantics is the interpretation-set itself.

Closedness under intersection is now a direct consequence for prf_3, mod_3, and stb_3 and was shown for adm_3 in [175].

Proposition 28. *Given arbitrary ADFs D_1 and D_2 it holds that*

- *there exists an ADF D such that $\sigma_3(D) = \sigma_3(D_1) \cap \sigma_3(D_2)$ if $\sigma_3(D_1) \cap \sigma_3(D_2) \neq \emptyset$ for $\sigma_3 \in \{adm_3, prf_3\}$;*
- *there exists an ADF D such that $\sigma_3(D) = \sigma_3(D_1) \cap \sigma_3(D_2)$ for $\sigma_3 \in \{mod_3, stb_3\}$.*

Proof. The results for prf_3, mod_3, and stb_3 follow from Proposition 27. The result for adm_3 can be found in [175]. $\square$

Pührer [175] gives a counterexample showing that the signature of complete semantics in ADFs is not closed under intersection and argues that the intersection of grounded interpretation-sets is only realizable if they coincide.

Table 3.2: Closure of ADF semantics. $\subseteq$: given ADF D, whether any $V \subseteq \sigma_3(F)$ is realizable. $\cap$: given ADFs D and D', whether $V = \sigma_3(F) \cap \sigma_3(F')$ is realizable. $y^\dagger$ expresses the restriction that $V \neq \emptyset$.

σ_3	adm_3	prf_3	com_3	mod_3	stb_3	grd_3
$\subseteq$	n	$y^\dagger$	n	y	y	$y^\dagger$
$\cap$	$y^\dagger$	$y^\dagger$	n	y	y	$y^\dagger$

3.6 Unifying Approach

In Sections 3.2, 3.3, and 3.4 we have studied several variants of realizability under two-valued semantics of AFs, while in Section 3.5 we have reviewed results on realizability under three-valued semantics of ADFs. Both is in line with the more common usage in the literature. Approaches to characterize three-valued semantics of AFs turned out to be either overly intricate or not very insightful when considering a more relaxed setting [112]. In this section, we present an alternative approach to characterize realizability under three-valued semantics of ADFs, which we can apply to AFs using the fact that AFs constitute a certain subclass of ADFs. In the course of this, we also capture two further classes lying between AFs and ADFs: SETAFs and bipolar ADFs. Compared to characterizations in the previous sections, the approach presented here is more algorithmic, with a concrete implementation in answer set programming presented in Section 3.6.2.

We consider realizability under three-valued semantics in AFs, SETAFs, BADFs, and ADFs for admissible, complete, preferred and two-valued model (reps. stable for AFs and SETAFs) semantics. In the interest of uniformity we will, throughout this section, refer to the stable semantics of AFs (resp. SETAFs) as two-valued model semantics and denote the stable labellings of an AF (resp. SETAF) F by $mod_3(F)$.[12]

3.6.1 A General Framework for Realizability

The underlying idea of the framework presented in this subsection is that all abstract argumentation formalisms considered (that is, AFs, SETAFs, bipolar ADFs and ADFs) can be viewed as subclasses of ADFs. This is clear for ADFs themselves and for BADFs by definition; for SETAFs and AFs this can be seen by Propositions 3 and 5, respectively. However, knowing that these formalisms can be recast as ADFs is not enough. To employ this knowledge for realizability, we must be able to precisely characterize the subclasses in terms of restrictions to the corresponding ADFs' acceptance functions. Fortunately, this is also possible and paves the way for the framework we present in this section. Most importantly, we will make use of the fact that different formalisms and different semantics can be characterized modularly, that is, independently of each other. The characterization

[12] Note that this is just an issue of notation, since model and stable semantics coincide for ADFs representing AFs and SETAFs (cf. Propositions 3 and 5).

of the signature of a particular semantics in a particular formalism is then obtained by combining the semantics characterization with the formalism characterization.

Towards a uniform account of realizability in ADFs under different semantics, we generalize the concept of *com*-characterizations (cf. Definition 67) by defining characterizing properties of functions $f : \mathcal{V}_2 \mapsto \mathcal{V}_2$ also for the other semantics. We start with a new characterization of realizability under admissible semantics in ADFs by means of an *adm*-characterization.

Definition 70. Let $V \subseteq \mathcal{V}$ be a set of interpretations. An *adm-characterization* of V is a function $f : \mathcal{V}_2 \mapsto \mathcal{V}_2$ such that: for each $v \in \mathcal{V}$ we have $v \in V$ iff for every $a \in A$:

- $v(a) \neq \mathbf{u}$ implies $f(v_2)(a) = v(a)$ for all $v_2 \in [v]_2$.

Similar to a *com*-characterization, an *adm*-characterization f assigns to a two-valued interpretation v and an argument a the value $f(v)(a)$ that the acceptance condition of a should evaluate to under v in an ADF that has V as its admissible semantics. Note that the only difference to Definition 67 is dropping the second condition related to arguments with truth value $\mathbf{u}$. However notice that a *com*-characterization is not necessarily an *adm*-characterization. While the two conditions in Definition 67 capture the relation $\Gamma_{D_f}(v) = v$, the remaining one in Definition 70 boils down to $v \leq_i \Gamma_{D_f}(v)$ that defines the admissible semantics (recall Definition 69 for D_f).

We show that the existence of an *adm*-characterization is necessary and sufficient for the existence of a realizing ADF.

Proposition 29. *Let $V \subseteq \mathcal{V}$ be a set of interpretations.*

1. *For each ADF D such that $adm_3(D) = V$, there is an adm-characterization f_D of V.*

2. *For each adm-characterization $f : \mathcal{V}_2 \mapsto \mathcal{V}_2$ of V we have $adm_3(D_f) = V$.*

Proof. (1) We define the function $f_D : \mathcal{V}_2 \mapsto \mathcal{V}_2$ as $f_D(v_2)(a) = v_2(\varphi_a)$ for every $v_2 \in \mathcal{V}_2$ and $a \in A$ where φ_a is the acceptance condition of a in D. We show that f_D is an *adm*-characterization of $V = adm_3(D)$. Let v be an interpretation and consider the case $v \in adm_3(D)$ and $v(a) \neq \mathbf{u}$ for some $a \in A$ and some $v_2 \in [v]_2$. From $v \leq_i \Gamma_D(v)$ we get $v_2(\varphi_a) = v(a)$. By definition of f_D it follows that $f_D(v_2)(a) = v(a)$. Now assume $v \notin adm_3(D)$ and consequently $v \not\leq_i \Gamma_D(v)$. There must be some $a \in A$ such that $v(a) \neq \mathbf{u}$ and $v(a) \neq \Gamma_D(v)(a)$. Hence, there is some $v_2 \in [v]_2$ with $v_2(\varphi_a) \neq v(a)$ and $f_D(v_2)(a) \neq v(a)$ by definition of f_D. Thus, f_D is an *adm*-characterization of V.

(2) First observe that for every two-valued interpretation v_2 and every $a \in A$ we have $f(v_2)(a) = v_2(\varphi_a^f)$.

($\subseteq$) Let $v \in adm_3(D_f)$ be an interpretation and $a \in A$ an argument such that $v(a) \neq \mathbf{u}$. Let v_2 be a two-valued interpretation with $v_2 \in [v]_2$. Since $v \leq_i \Gamma_{D_f}(v)$ we have $v(a) = v_2(\varphi_a^f)$. Therefore, by our observation it must also hold that $f(v_2)(a) = v(a)$. Thus, by Definition 70, $v \in V$.

($\supseteq$) Consider an interpretation v such that $v \notin adm_3(D_f)$. We show that $v \notin V$. From $v \notin adm_3(D_f)$ we get $v \not\leq_i \Gamma_{D_f}(v)$. There must be some $a \in A$ such that $v(a) \neq \mathbf{u}$ and $v(a) \neq \Gamma_{D_f}(v)(a)$. Hence, there is some $v_2 \in [v]_2$ with $v_2(\varphi_a^f) \neq v(a)$ and consequently $f(v_2)(a) \neq v(a)$. Thus, by Definition 70, we have $v \notin V$.

□

Example 39. Consider the sets $V_1 = \{\mathbf{uuu}, \mathbf{tff}, \mathbf{ftu}\}$ and $V_2 = \{\mathbf{tff}, \mathbf{ftu}\}$ of interpretations over $A = \{a, b, c\}$. The following mapping f is an *adm*-characterization of V_1:

v	**ttt**	**ttf**	**tft**	**tff**	**ftt**	**ftf**	**fft**	**fff**
$f(v)$	**ftt**	**tft**	**ttt**	**tff**	**ftf**	**ftt**	**ttf**	**ftf**

To see this, we have to check, for instance, that all $v_2 \in [\mathbf{ftu}]_2 = \{\mathbf{ftt}, \mathbf{ftf}\}$ have $f(v_2)(a) = v(a) = \mathbf{f}$ and $f(v_2)(b) = v(b) = \mathbf{t}$, which is indeed the case. We can verify that the condition given in Definition 70 is fulfilled by all $v \in V_1$ and violated for all $v \in (\mathcal{V} \setminus V_1)$. Thus, the ADF D_f has V_1 as its admissible interpretations. Indeed, the realizing ADF is given by the following acceptance conditions:

$$\begin{aligned}
\varphi_a^f &= (a \wedge b \wedge \neg c) \vee (a \wedge \neg b) \vee (\neg a \wedge \neg b \wedge c) \\
\varphi_b^f &= (a \wedge c) \vee (\neg a \wedge b) \vee (\neg a \wedge \neg b \wedge \neg c) \\
\varphi_c^f &= (a \wedge b) \vee (\neg a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge c)
\end{aligned}$$

Further note that f is also a *com*-characterization of V_1, hence also $com_3(D_f) = V_1$.

For V_2, on the other hand, no *adm*-characterization exists because $\mathbf{uuu} \notin V_2$, but the implication of Definition 70 trivially holds for a, b, and c. ◊

We have seen that the construction D_f for realizing interpretation-sets under complete semantics can also be used for realizing a set V of interpretations under admissible semantics. The only difference is that we here require f to be an *adm*-characterization instead of a *com*-characterization of V. Note that admissible semantics can be characterized by properties that are easier to check than existence of an *adm*-characterization (see the characterization given in Theorem 24 and further shortcuts presented by Pührer [175]). However, using the same type of characterizations for different semantics allows us to present a unified approach for checking realizability and constructing a realizing ADF in case one exists.

For realizing under two-valued models, we can likewise present an adjusted version of *com*-characterizations.

Definition 71. Let $V \subseteq \mathcal{V}$ be a set of interpretations. A *mod-characterization* of V is a function $f : \mathcal{V}_2 \mapsto \mathcal{V}_2$ such that:

1. f is defined on V (that is, $V \subseteq \mathcal{V}_2$) and
2. for each $v \in \mathcal{V}_2$, we have $v \in V$ iff $f(v) = v$.

As we can show, there is a one-to-one correspondence between *mod*-characterizations and ADF realizations.

Proposition 30. *Let $V \subseteq \mathcal{V}$ be a set of interpretations.*

1. *For each ADF D such that $mod_3(D) = V$, there is a mod-characterization f_D of V.*
2. *For each mod-characterization $f : \mathcal{V}_2 \mapsto \mathcal{V}_2$ of V we find $mod_3(D_f) = V$.*

Proof. (1) Let D be an ADF with $mod_3(D) = V$. It immediately follows that $V \subseteq \mathcal{V}_2$. We define the function $f_D : \mathcal{V}_2 \mapsto \mathcal{V}_2$ as $f_D(v_2)(a) = v_2(\varphi_a)$ for every $v_2 \in \mathcal{V}_2$ and $a \in A$ just as in the proof of Proposition 29. It follows directly that for any $v \in \mathcal{V}_2$, we find $f_D(v) = v$ iff $v \in V$. Thus f_D is a *mod*-characterization of V.

(2) Let $V \subseteq \mathcal{V}$ and $f : \mathcal{V}_2 \mapsto \mathcal{V}_2$ be a *mod*-characterization of V. By property 1 of f (cf. Definition 71) we follow that $V \subseteq \mathcal{V}_2$. Further for any $v \in \mathcal{V}_2$ we have:

$$\begin{aligned}
v \in V &\iff v = f(v) \\
&\iff \forall a \in A : (v(a) = f(v)(a)) \\
&\iff \forall a \in A : (v(a) = \mathbf{t} \leftrightarrow f(v)(a) = \mathbf{t}) \\
&\iff \forall a \in A : (v(a) = \mathbf{t} \ \leftrightarrow (\exists w \in \mathcal{V}_2 : f(w)(a) = \mathbf{t} \wedge v = w)) \\
&\iff \forall a \in A : (v(a) = \mathbf{t} \ \leftrightarrow (\exists w \in \mathcal{V}_2 : f(w)(a) = \mathbf{t} \wedge v(\phi_w) = \mathbf{t})) \\
&\iff \forall a \in A : \left(v(a) = \mathbf{t} \ \leftrightarrow v\left(\bigvee_{w \in \mathcal{V}_2, f(w)(a) = \mathbf{t}} \phi_w \right) = \mathbf{t} \right) \\
&\iff \forall a \in A : v(a) = v\left(\bigvee_{w \in \mathcal{V}_2, f(w)(a) = \mathbf{t}} \phi_w \right) \\
&\iff \forall a \in A : v(a) = v(\varphi_a^f) \\
&\iff v \in mod_3(D_f)
\end{aligned}$$

□

A related result was given by Strass [188, Proposition 10]. There it was shown that for every $V \subseteq \mathcal{V}_2$, there is a one-to-one-correspondence between the sets $\mathcal{D}_V = \{D = \{\langle a, \varphi_a \rangle \mid a \in A\} \mid mod_3(D) = V\}$ of realizations and $\mathcal{Y}_V = \{\{Y_a\}_{a \in A} \mid \forall a \in A : Y_a \subseteq$

$\mathcal{V}_2, \bigcap_{a\in A} Y_a = V\}$ of alternative characterizations. The characterization we presented here, however, fits into the general framework of this work and will be directly usable for our realizability algorithm.

The next result summarizes how ADF realizability can be captured by different types of characterizations for the semantics we considered so far.

Theorem 26. *Let $V \subseteq \mathcal{V}$ be a set of interpretations and $\sigma \in \{adm, com, mod\}$. There is an ADF D such that $\sigma_3(D) = V$ if and only if there is a σ-characterization of V.*

The preferred semantics of an ADF D is closely related to admissible semantics as, by definition, the preferred interpretations of D are its $\leq_i$-maximal admissible interpretations.

As a consequence we can also describe preferred realizability in terms of *adm*-characterizations. Recall the lattice-theoretic standard notation $\max_{\leq_i} V$ to denote the $\leq_i$-maximal elements of a given set V.

Corollary 5. *Let $V \subseteq \mathcal{V}$ be a set of interpretations. There is an ADF D with $prf_3(D) = V$ if and only if there is an adm-characterization of some $V' \subseteq \mathcal{V}$ with $V \subseteq V'$ and $\max_{\leq_i} V' = V$.*

Algorithm for Deciding Realizability

The main algorithm for deciding realizability, Algorithm 1, is a propagate-and-guess algorithm in the spirit of the DPLL algorithm [82] for deciding propositional satisfiability [132]. It is generic with respect to (1) the formalism $\mathcal{F}$ and (2) the semantics σ_3 for which a given set of interpretations should be realized. To this end, the propagation part of the algorithm is kept exchangeable and will vary depending on formalism and semantics. Roughly, in the propagation step the algorithm uses the desired set V of interpretations to derive certain necessary properties of the realizing knowledge base (line 2). This is the essential part of the algorithm: the derivation rules (*propagators*) used there are based on characterizations of realizability with respect to formalism and semantics. The concrete propagators will be explained in detail in the next two subsections. Once propagation of properties has reached a fixed point (line 7), i.e. not further necessary properties can be derived by the propagators, the algorithm checks whether the derived information is sufficient to construct a knowledge base. If so, the knowledge base can be constructed and returned (line 9). Otherwise (no more information can be obtained through propagation and there is not enough information to construct a knowledge base yet), the algorithm guesses another assignment for the characterization (line 11) and calls itself recursively.

The main data structure that Algorithm 1 operates on is a set of triples $(v, a, \mathbf{x})$ consisting of a two-valued interpretation $v \in \mathcal{V}_2$, an argument $a \in A$ and a truth value $\mathbf{x} \in \{\mathbf{t}, \mathbf{f}\}$. This data structure is intended to represent the σ-characterizations introduced in Definitions 67, 70, and 71 for $\sigma \in \{com, adm, mod\}$, respectively. There, a σ-characterization is a function $f : \mathcal{V}_2 \mapsto \mathcal{V}_2$ from two-valued interpretations to two-valued interpretations. However, as the algorithm builds the σ-characterization step by step and there might

Algorithm 1 *realize*$(\mathcal{F}, \sigma_3, V, \mathbb{F})$

Input:
- a formalism $\mathcal{F}$
- a semantics σ_3 for $\mathcal{F}$
- a set V of interpretations $v : A \mapsto \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$
- a relation $\mathbb{F} \subseteq \mathcal{V}_2 \times A \times \{\mathbf{t}, \mathbf{f}\}$, initially empty

Output: a $\mathsf{kb} \in \mathcal{F}$ with $\sigma_3(\mathsf{kb}) = V$ or “no” if none exists

repeat
  set $\mathbb{F}_\Delta := \bigcup_{p \in P_\sigma^{\mathcal{F}}} p(V, \mathbb{F}) \setminus \mathbb{F}$
  set $\mathbb{F} := \mathbb{F} \cup \mathbb{F}_\Delta$
  if $\exists v \in \mathcal{V}_2, \exists a \in A : \{(v, a, \mathbf{t}), (v, a, \mathbf{f})\} \subseteq \mathbb{F}$ **then**
    return “no”
  end if
until $\mathbb{F}_\Delta = \emptyset$
if $\forall v \in \mathcal{V}_2, \forall a \in A, \exists \mathbf{x} \in \{\mathbf{t}, \mathbf{f}\} : (v, a, \mathbf{x}) \in \mathbb{F}$ **then**
  return $kb_\sigma^{\mathcal{F}}(\mathbb{F})$
end if
choose $v \in \mathcal{V}_2, a \in A$ with $(v, a, \mathbf{t}) \notin \mathbb{F}$, $(v, a, \mathbf{f}) \notin \mathbb{F}$, and $\mathbf{x} \in \{\mathbf{t}, \mathbf{f}\}$
if *realize*$(\mathcal{F}, \sigma_3, V, \mathbb{F} \cup \{(v, a, \mathbf{x})\}) \neq$ “no” **then**
  return *realize*$(\mathcal{F}, \sigma_3, V, \mathbb{F} \cup \{(v, a, \mathbf{x})\})$
else
  return *realize*$(\mathcal{F}, \sigma_3, V, \mathbb{F} \cup \{(v, a, \neg\mathbf{x})\})$
end if

not even be a σ-characterization in the end (because V is not realizable), we use a set $\mathbb{F}$ of triples $(v, a, \mathbf{x})$ to be able to represent both partial and incoherent states of affairs. The σ-characterization candidate induced by $\mathbb{F}$ is

- *partial* if we have that for some v and a, neither $(v, a, \mathbf{t}) \in \mathbb{F}$ nor $(v, a, \mathbf{f}) \in \mathbb{F}$; and
- *incoherent* if for some v and a, both $(v, a, \mathbf{t}) \in \mathbb{F}$ and $(v, a, \mathbf{f}) \in \mathbb{F}$.

If $\mathbb{F}$ is neither partial nor incoherent, it gives rise to a unique σ-characterization that can be used to construct the knowledge base realizing the desired set of interpretations. The correspondence to the potential σ-characterization is as follows:

Definition 72. Given a relation $\mathbb{F} \subseteq \mathcal{V}_2 \times A \times \{\mathbf{t}, \mathbf{f}\}$ that is not incoherent, we define the corresponding (partial) function $f_{\mathbb{F}}$ such that, for each $v \in \mathcal{V}_2$ and $a \in A$,

$$f_{\mathbb{F}}(v)(a) = \mathbf{x} \iff (v, a, \mathbf{x}) \in \mathbb{F}.$$

We may sometimes, with some abuse of notation, refer to a relation $\mathbb{F}$ as (partial) σ-characterization, implicitly meaning the corresponding (partial) function $f_{\mathbb{F}}$.

In the case where the constructed relation $\mathbb{F}$ becomes functional at some point, i.e. $\mathbb{F}$ is neither partial nor incoherent, the algorithm returns a realizing knowledge base $kb_\sigma^{\mathcal{F}}(\mathbb{F})$. When it comes to ADFs, we can use the canonical construction given in Definition 68 to obtain a realizing function.

Definition 73. Given a semantics $\sigma \in \{com, adm, mod\}$ and a σ-characterization $\mathbb{F}$, we define the canonical realization

$$kb_\sigma^{\mathrm{ADF}}(\mathbb{F}) = D_{f_{\mathbb{F}}}.$$

For the remaining formalisms we will introduce the respective realizing function in later subsections.

In our presentation of the algorithm we focused on its main features, therefore the guessing step (line 11) is completely "blind", i.e. the new assignment is randomly chosen from all unassigned combinations of interpretations and arguments. It is, however, possible to use techniques known from constraint satisfaction problems, such as shaving [193] (removing guessing possibilities that directly lead to inconsistency) to improve performance.

Moreover note that Algorithm 1 can be extended to enumerate all possible realizations of a given interpretation-set. This is done by keeping all choice points in the guessing step and thus proceeding the search after returning a realizing knowledge base. This exhaustively explores the whole search space.

The algorithm is parametric in two dimensions, namely with respect to the formalism $\mathcal{F}$ and with respect to the semantics σ_3. These two aspects come into the algorithm via so-called *propagators*. A propagator is a formalism-specific or semantics-specific set of derivation rules. Given a set V of desired interpretations and a partial σ-characterization $\mathbb{F}$, a propagator p derives new tuples $(v, a, \mathbf{x})$ that must necessarily be part of any total σ-characterization $\mathbb{F}$ of V. In what follows, we present semantics propagators for admissible, complete and model (in (SET)AF terms stable) semantics, and then describe formalism propagators for BADFs, AFs, and SETAFs.

Semantics Propagators

The semantics propagators are directly derived from the properties of σ-characterizations presented in Definitions 67, 70, and 71 for complete, admissible, and model semantics, respectively. While the definitions provide exact conditions to check whether a given function is a σ-characterization, the propagators allow us to derive definite values of partial characterizations that are necessary to fulfill the conditions for being a σ-characterization.

Admissible Semantics. For admissible semantics, the condition for a function f to be an *adm*-characterization of a given set of interpretations V (cf. Definition 70) can be split into a condition for desired interpretations $v \in V$ and two conditions for undesired interpretations $v \notin V$. The set of propagators is given by $P_{adm}^{\mathrm{ADF}} = \{p_{adm}^{\in}, p_{adm}^{\notin}, p_{adm}^{\nleq}\}$, as defined in Figure 3.29. They add tuples to the relation $\mathbb{F}$ as follows, in order to ensure

$$p^{\in}_{adm}(V, \mathbb{F}) = \{(v_2, a, v(a)) \mid v \in V, v_2 \in [v]_2, v(a) \neq \mathbf{u}\}$$

$$p^{\notin}_{adm}(V, \mathbb{F}) = \{(v_2, a, \neg v(a)) \mid v \in \mathcal{V} \setminus V, v_2 \in [v]_2, v(a) \neq \mathbf{u}, \\ \forall b \in A \setminus v^{\mathbf{u}}, \forall v_2' \in [v]_2 : (a, v_2) \neq (b, v_2') \Rightarrow (v_2', b, v(b)) \in \mathbb{F}\}$$

$$p^{\notin}_{adm}(V, \mathbb{F}) = \{(v, a, \mathbf{t}), (v, a, \mathbf{f}) \mid v \in \mathcal{V}_2, a \in A, v_{\mathbf{u}} \notin V\}$$

Figure 3.29: Semantics propagators for the admissible semantics.

that $f_{\mathbb{F}}$ is an *adm*-characterization of V if V is realizable under adm_3 or to ensure that $\mathbb{F}$ is incoherent otherwise.

- Propagator $p^{\in}_{adm}$ derives new tuples by considering interpretations $v \in V$. Here, for all two-valued interpretations v_2 that extend v, the value $f_{\mathbb{F}}(v_2)$ has to be in accordance with v on v's Boolean part, that is, the algorithm adds $(v_2, a, v(a))$ to $\mathbb{F}$ whenever $v(a) \neq \mathbf{u}$.
- On the other hand, $p^{\notin}_{adm}$ derives new tuples for $v \notin V$ in order to ensure that there is a two-valued interpretation v_2 extending v where $f_{\mathbb{F}}(v_2)$ differs from v on a Boolean value of v. It adds $(v_2, a, \neg v(a))$ to $\mathbb{F}$ whenever $v(a) \neq \mathbf{u}$, all other $v_2' \in [v]_2$ have $f_{\mathbb{F}}(v_2')(a) = v(a)$, and all other arguments b with $v(b) \neq \mathbf{u}$ have $f_{\mathbb{F}}(v_2')(b) = v(b)$ for all $v_2' \in [v]_2$.

Note that, while $p^{\in}_{adm}$ immediately allows us to derive information about $\mathbb{F}$ for each desired interpretation $v \in V$, propagator $p^{\notin}_{adm}$ is much weaker in the sense that it only derives a triple of $\mathbb{F}$ if there is no other way to meet the conditions for an undesired interpretation $v \notin V$.

- Special treatment is required for the interpretation $v_{\mathbf{u}}$ that maps all arguments to $\mathbf{u}$, since it is admissible for every ADF. This is not captured by $p^{\in}_{adm}$ and $p^{\notin}_{adm}$ as these deal only with interpretations that have Boolean mappings. Thus, propagator $p^{\notin}_{adm}$ serves to check whether $v_{\mathbf{u}} \in V$. If this is not the case, the propagator immediately makes the relation $\mathbb{F}$ incoherent and the algorithm correctly answers "no".

Example 40. Consider the set of interpretations $V_3 = \{\mathbf{uuu}, \mathbf{fuu}, \mathbf{uuf}, \mathbf{ftf}\}$. We use Algorithm 1 to test realizability of V_3 under admissible semantics. To this end, we consider a run of *realize*(ADF, $adm_3, V_3, \emptyset$). In the first iteration, propagator $p^{\in}_{adm}$ ensures that $\mathbb{F}_{\Delta}$ in line 2 contains, among others, the tuples $(\mathbf{fff}, a, \mathbf{f})$, $(\mathbf{ftf}, a, \mathbf{f})$, $(\mathbf{ftf}, c, \mathbf{f})$, and $(\mathbf{fff}, c, \mathbf{f})$. Based on the latter three tuples and $\mathbf{fuf} \notin V_3$, propagator $p^{\notin}_{adm}$ derives $(\mathbf{fff}, a, \mathbf{t}) \in \mathbb{F}$ in the second iteration which together with $(\mathbf{fff}, a, \mathbf{f}) \in \mathbb{F}$ causes the algorithm to return "no" in line 5. Consequently, V_3 is not adm_3-realizable. $\Diamond$

$$
\begin{aligned}
p^{\in,\mathbf{tf}}_{com}(V,\mathbb{F}) = \{(v_2, a, v(a)) \;\mid\; & v \in V, v_2 \in [v]_2, v(a) \neq \mathbf{u}\} \\
p^{\in,\mathbf{u}}_{com}(V,\mathbb{F}) = \{(v_2, a, \neg\mathbf{x}) \;\mid\; & v \in V, v_2 \in [v]_2, v(a) = \mathbf{u}, \mathbf{x} \in \{\mathbf{t}, \mathbf{f}\}, \\
& \forall v_2' \in [v]_2 : v_2 \neq v_2' \Rightarrow (v_2', a, \mathbf{x}) \in \mathbb{F}\} \\
p^{\notin,\mathbf{tf}}_{com}(V,\mathbb{F}) = \{(v_2, a, \neg v(a)) \;\mid\; & v \in \mathcal{V} \setminus V, v_2 \in [v]_2, v(a) \neq \mathbf{u}, \\
& \forall b \in A \setminus v^{\mathbf{u}}, \forall v_2' \in [v]_2 : (a, v_2) \neq (b, v_2') \Rightarrow (v_2', b, v(b)) \in \mathbb{F}, \\
& \forall b \in v^{\mathbf{u}}, \exists v_2'', v_2''' \in [v]_2 : (v_2'', b, \mathbf{t}), (v_2''', b, \mathbf{f}) \in \mathbb{F}\} \\
p^{\notin,\mathbf{u}}_{com}(V,\mathbb{F}) = \{(v_2, a, \neg\mathbf{x}) \;\mid\; & v \in \mathcal{V} \setminus V, v_2 \in [v]_2, v(a) = \mathbf{u}, \\
& \forall b \in A \setminus v^{\mathbf{u}}, \forall v_2' \in [v]_2 : (v_2', b, v(b)) \in \mathbb{F}, \\
& \forall b \in v^{\mathbf{u}} \setminus \{a\} : \exists v_2'', v_2''' \in [v]_2 : (v_2'', b, \mathbf{t}), (v_2''', b, \mathbf{f}) \in \mathbb{F}, \\
& \qquad \forall v_2'''' \in [v]_2 \setminus \{v_2\} : (v_2'''', b, \mathbf{x}) \in \mathbb{F}\}
\end{aligned}
$$

Figure 3.30: Semantics propagators for the complete semantics.

Complete Semantics. For complete semantics the propagators are derived from the notion of a *com*-characterization (cf. Definition 67) and are given by the set $P^{\mathrm{ADF}}_{com} = \{p^{\in,\mathbf{tf}}_{com}, p^{\in,\mathbf{u}}_{com}, p^{\notin,\mathbf{tf}}_{com}, p^{\notin,\mathbf{u}}_{com}\}$, defined in Figure 3.30. Given a set of interpretations V the propagators add tuples to the relation $\mathbb{F}$ as follows.

- Propagator $p^{\in,\mathbf{tf}}_{com}$ is equivalent to $p^{\in}_{adm}$ and derives tuples based on interpretations $v \in V$ just like in the admissible case.
- Propagator $p^{\in,\mathbf{u}}_{com}$ is also based on interpretations $v \in V$ and deals with arguments $a \in A$ having $v(a) = \mathbf{u}$. For these arguments there have to be at least two interpretations $v_2, v_2' \in [v]_2$ having $f_{\mathbb{F}}(v_2)(a) = \mathbf{t}$ and $f_{\mathbb{F}}(v_2')(a) = \mathbf{f}$. Hence $p^{\in,\mathbf{u}}_{com}$ derives triple $(v_2, a, \neg\mathbf{x})$ if for all other $v_2' \in [v]_2$ we find a triple $(v_2', a, \mathbf{x}) \in \mathbb{F}$ with the same $\mathbf{x} \in \{\mathbf{t}, \mathbf{f}\}$.
- For interpretations $v \notin V$ it must hold that there is some $a \in A$ such that (i) $v(a) \neq \mathbf{u}$ and $f_{\mathbb{F}}(v_2)(a) \neq v(a)$ for some $v_2 \in [v]_2$ or (ii) $v(a) = \mathbf{u}$ but for all $v_2 \in [v]_2$, $f_{\mathbb{F}}(v_2)$ assigns the same Boolean truth value $\mathbf{x} \in \{\mathbf{t}, \mathbf{f}\}$ to a. Now if neither (i) nor (ii) can be fulfilled by any argument $b \in (A \setminus \{a\})$ due to the current contents of $\mathbb{F}$, propagators $p^{\notin,\mathbf{tf}}_{com}$ and $p^{\notin,\mathbf{u}}_{com}$ derive tuple $(v_2, a, \neg v(a))$ for $v(a) \neq \mathbf{u}$ if needed for a to fulfill (i) (i.e. for all $v_2' \in ([v]_2 \setminus \{v_2\})$ we have $f_{\mathbb{F}}(v_2')(a) = v(a)$) and $(v_2, a, \neg\mathbf{x})$ for $v(a) = \mathbf{u}$ if needed for a to fulfill (ii) (i.e. for all $v_2' \in ([v]_2 \setminus \{v_2\})$ we have $f_{\mathbb{F}}(v_2')(a) = \mathbf{x}$), respectively.

Example 41. Consider the set of interpretations $V_4 = \{\mathbf{uuu}, \mathbf{tuu}, \mathbf{ftu}, \mathbf{tff}\}$ over arguments $\{a, b, c\}$ (note that V_4 extends V_1 from Example 39 by the interpretation $\mathbf{tuu}$). In order to check whether V_4 is realizable under complete semantics we simulate a

Table 3.3: Simulation of the execution of *realize*(ADF, $com_3, V_4, \emptyset$) for $V_4 = \{\mathbf{uuu}, \mathbf{tuu}, \mathbf{ftu}, \mathbf{tff}\}$. Rows represent the levels of recursion. The columns $p^{\in,\mathbf{tf}}_{com}$, $p^{\in,\mathbf{u}}_{com}$, $p^{\notin,\mathbf{tf}}_{com}$, and $p^{\notin,\mathbf{u}}_{com}$ contain the tuples added by the respective propagators. Tuples in the column "guess" are guessed if no tuple can be derived by a propagator. Two-valued interpretations in the column "done" fulfill the conditions given by a *com*-characterization and need not to be considered anymore.

	guess	$p^{\in,\mathbf{tf}}_{com}$	$p^{\in,\mathbf{u}}_{com}$	$p^{\notin,\mathbf{tf}}_{com}$	$p^{\notin,\mathbf{u}}_{com}$	done
1		$(\mathbf{ttt}, a, \mathbf{t})$, $(\mathbf{ttf}, a, \mathbf{t})$, $(\mathbf{tft}, a, \mathbf{t})$, $(\mathbf{tff}, a, \mathbf{t})$, $(\mathbf{tff}, b, \mathbf{f})$, $(\mathbf{tff}, c, \mathbf{f})$, $(\mathbf{ftt}, a, \mathbf{f})$, $(\mathbf{ftt}, b, \mathbf{t})$, $(\mathbf{ftf}, a, \mathbf{f})$, $(\mathbf{ftf}, b, \mathbf{t})$				$\mathbf{tff}$
2				$(\mathbf{ftt}, c, \mathbf{f})$, $(\mathbf{ftf}, c, \mathbf{t})$		$\mathbf{ftt}$, $\mathbf{ftf}$, $\mathbf{utt}$, $\mathbf{utf}$, $\mathbf{ftu}$, $\mathbf{fut}$, $\mathbf{fuf}$, $\mathbf{uut}$, $\mathbf{uuf}$, $\mathbf{uuu}$
3	$(\mathbf{ttt}, b, \mathbf{f})$					$\mathbf{ttt}$, $\mathbf{ttu}$, $\mathbf{utu}$
4	$(\mathbf{tft}, b, \mathbf{f})$					$\mathbf{tut}$
5			$(\mathbf{ttf}, b, \mathbf{t})$		$(\mathbf{tft}, c, \mathbf{f})$	$\mathbf{tft}$, $\mathbf{tfu}$, $\mathbf{uft}$
6				$(\mathbf{ttf}, c, \mathbf{t})$		$\mathbf{ttf}$, $\mathbf{tuf}$, $\mathbf{tuu}$
7	$(\mathbf{ttt}, c, \mathbf{t})$					
8	$(\mathbf{fft}, a, \mathbf{f})$					
9	$(\mathbf{fff}, a, \mathbf{f})$					
10					$(\mathbf{fft}, b, \mathbf{t})$, $(\mathbf{fff}, b, \mathbf{t})$	$\mathbf{fft}$, $\mathbf{fff}$, $\mathbf{fuu}$, $\mathbf{ffu}$, $\mathbf{uff}$, $\mathbf{ufu}$
11	$(\mathbf{fft}, c, \mathbf{f})$					
12	$(\mathbf{fff}, c, \mathbf{f})$					

run of *realize*(ADF, $com_3, V_4, \emptyset$), which seeks for a relation $\mathbb{F}$ such that $f_{\mathbb{F}}$ is a *com*-characterization of V_4. Table 3.3 shows, for each level of the recursion, the tuples added to $\mathbb{F}$ by the respective propagators. If no tuple can be inferred, we make an arbitrary guess and denote it in the second column of the table. Finally a few words are in order concerning the last column of Table 3.3: it lists those interpretations which meet the conditions given by the definition of a *com*-characterization at the end of the respective level of the recursion. That is, an interpretation $v \in \mathcal{V}$ appears in the "done"-column, if the tuples added to $\mathbb{F}$ in the current level of computation made the two conditions of Definition 67 fulfilled for all arguments (if $v \in V_4$), or violated for some argument (if $v \notin V_4$). Practically, it means that for interpretations which appear in the "done"-column, no further tuples have to be derived by the propagators.

As an example, consider row 5, and recall that $\mathbf{tuu} \in V_4$. Further observe that at this point of computation, we already have $f_{\mathbb{F}}(\mathbf{ttt})(b) = f_{\mathbb{F}}(\mathbf{tft})(b) = f_{\mathbb{F}}(\mathbf{tff})(b) = \mathbf{f}$. Hence, $f_{\mathbb{F}}$ must

$$p^{\in}_{mod}(V,\mathbb{F}) = \{(v,a,v(a)) \mid v \in V, a \in A\}$$

$$p^{\notin}_{mod}(V,\mathbb{F}) = \{(v,a,\neg v(a)) \mid v \in \mathcal{V}_2 \setminus V, a \in A, \forall c \in A \setminus \{a\} : (v,c,v(c)) \in \mathbb{F}\}$$

$$p^{\nsubseteq}_{mod}(V,\mathbb{F}) = \{(v,a,\mathbf{t}),(v,a,\mathbf{f}) \mid v \in \mathcal{V}_2, a \in A, V \not\subseteq \mathcal{V}_2\}$$

Figure 3.31: Semantics propagators for the model semantics.

assign $\mathbf{t}$ to b for the only remaining element of $[\mathbf{tuu}]_2$, i.e. $p^{\in,\mathbf{u}}_{com}$ adds the tuple $(\mathbf{ttf},b,\mathbf{t})$ to $\mathbb{F}$. Moreover, considering $\mathbf{tfu} \notin V_4$ and observing that $f_{\mathbb{F}}(\mathbf{tft})(a) = f_{\mathbb{F}}(\mathbf{tff})(a) = \mathbf{t}$ and $f_{\mathbb{F}}(\mathbf{tft})(b) = f_{\mathbb{F}}(\mathbf{tff})(b) = \mathbf{f}$, we must have that $f_{\mathbb{F}}(\mathbf{tft})(c) = f_{\mathbb{F}}(\mathbf{tff})(c) = \mathbf{x}$ for some $\mathbf{x} \in \{\mathbf{t},\mathbf{f}\}$. Hence $p^{\notin;\mathbf{u}}_{com}$ derives, knowing that $f_{\mathbb{F}}(\mathbf{tff})(c) = \mathbf{f}$, the tuple $(\mathbf{tft},c,\mathbf{f})$.

In the end, for every $v \in \mathcal{V}_2$ and $a \in A$ exactly one element $(v,a,\mathbf{x})$ with $\mathbf{x} \in \{\mathbf{t},\mathbf{f}\}$ is contained in $\mathbb{F}$. We have arrived at the following *com*-characterization $f_{\mathbb{F}}$ of V_4:

v	**ttt**	**ttf**	**tft**	**tff**	**ftt**	**ftf**	**fft**	**fff**
$f_{\mathbb{F}}(v)$	**tft**	**ttt**	**tff**	**tff**	**ftf**	**ftt**	**ftf**	**ftf**

Having established the *com*-characterization, the algorithm returns a realizing ADF by using the realizing function $kb^{\mathrm{ADF}}_{\sigma}(\mathbb{F})$. We get the ADF $D_{f_{\mathbb{F}}}$ which is given by the following acceptance conditions:

$$\begin{aligned}
\varphi_a &= (a \wedge b \wedge c) \vee (a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge c) \vee (a \wedge \neg b \wedge \neg c) = \\
&= a \\
\varphi_b &= (a \wedge b \wedge \neg c) \vee (\neg a \wedge b \wedge c) \vee (\neg a \wedge b \wedge \neg c) \vee (\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge \neg b \wedge \neg c) = \\
&= (a \wedge b \wedge \neg c) \vee \neg a \\
\varphi_c &= (a \wedge b \wedge c) \vee (a \wedge b \wedge \neg c) \vee (\neg a \wedge b \wedge \neg c) = \\
&= (a \wedge b) \vee (\neg a \wedge b \wedge \neg c)
\end{aligned}$$

$\Diamond$

Model Semantics. Finally, for two-valued model semantics, propagator $p^{\in}_{mod}$ derives new tuples by looking at interpretations $v \in V$. For those, we must find $f(v) = v$ in each *mod*-characterization f by definition. Thus the propagator adds $(v,a,v(a))$ for each $a \in A$ to the partial characterization $\mathbb{F}$. Propagator $p^{\notin}_{mod}$ looks at interpretations $v \in \mathcal{V}_2 \setminus V$, for which it must hold that $f(v) \neq v$. Thus there must be an argument $a \in A$ with $v(a) \neq f(v)(a)$, which is exactly what this propagator derives whenever it is clear that there is only one argument candidate left. This, in turn, is the case whenever all $b \in A$ with the opposite truth value $\neg v(a)$ and all $c \in A$ with $c \neq a$ cannot coherently

Algorithm 2 *realizePrf*$(\mathcal{F}, V)$

Input:
- a formalism $\mathcal{F}$
- a set V of interpretations $v : A \mapsto \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$

Output: Return some $\mathsf{kb} \in \mathcal{F}$ with $prf_3(\mathsf{kb}) = V$ if one exists or "no" otherwise.

1: **if** $\max_{\leq_i} V \neq V$ **then**
2:    **return** "no"
3: **end if**
4: **set** $V^{<_i} := \{v \in \mathcal{V} \mid \exists v' \in V : v <_i v'\}$
5: **set** $X := \emptyset$
6: **repeat**
7:    **choose** $V' \subseteq V^{<_i}$ with $V' \notin X$
8:    **set** $X := X \cup \{V'\}$
9:    **set** $V^{adm} := V \cup V'$
10:    **if** *realize*$(\mathcal{F}, adm_3, V^{adm}, \emptyset) \neq$ "no" **then**
11:       **return** *realize*$(\mathcal{F}, adm_3, V^{adm}, \emptyset)$
12:    **end if**
13: **until** $\forall V' \subseteq V^{<_i} : V' \in X$
14: **return** "no"

become the necessary witness any more. The propagator $p^{\lightning}_{mod}$ checks whether $V \subseteq \mathcal{V}_2$, that is, the desired set of interpretations consists entirely of two-valued interpretations. In that case this propagator makes the relation $\mathbb{F}$ incoherent, following a similar strategy as $p^{\lightning}_{adm}$.

Preferred Semantics. Realizing a given set of interpretations V under preferred semantics requires special treatment. We do not have a σ-characterization function for $\sigma = prf$ at hand to directly check realizability of V. Therefore our strategy is to find some $V' \subseteq \{v \in \mathcal{V} \mid \exists v' \in V : v <_i v'\}$ such that $V \cup V'$ is realizable under admissible semantics (cf. Corollary 5). Algorithm 2 implements this idea by guessing such a V' (line 7) and then making use of Algorithm 1 to try to realize $V \cup V'$ under admissible semantics (line 11). If *realize* returns a knowledge base kb realizing $V \cup V'$ under adm_3 we can directly use kb as solution of *realizePrf* since it holds that $prf_3(\mathsf{kb}) = V$, given that V is incompatible (line 2).

Formalism Propagators

Bipolar ADFs, SETAFs and AFs are all subclasses of ADFs by restricting the acceptance conditions of arguments. In bipolar ADFs, every link from argument a to argument b must be supporting or attacking, as manifested in the acceptance condition of b; for SETAFs the acceptance conditions of the corresponding ADF can always be written as a conjunction of disjunctions of negated atoms (cf. Definition 33); finally ADFs obtained from translating AFs have conjunctions of negated atoms as acceptance conditions (cf. Definition 32).

As we have seen in in Definition 73, when constructing an ADF realizing a given set V of interpretations under a semantics σ, the function $kb_\sigma^{\mathrm{ADF}}(\mathbb{F})$ makes use of the σ-characterization given by $\mathbb{F}$ in the following way: v is a model of the acceptance condition φ_a if and only if we find $(v, a, \mathbf{t}) \in \mathbb{F}$ (cf. Definition 68).

Combining these observations, the restrictions imposed by the ADF subclasses on the acceptance conditions also carry over to the σ-characterizations. Therefore we define propagators that use structural knowledge on the form of acceptance conditions of the respective formalisms to reduce the search space or to induce incoherence of $\mathbb{F}$ whenever V is not realizable.

In the following we define a propagator $p^{\mathcal{F}}$ for every formalism $\mathcal{F} \in \{\mathrm{AF}, \mathrm{SETAF}, \mathrm{BADF}\}$. The set of propagators for formalism $\mathcal{F}$ and semantics $\sigma \in \{\mathit{adm}, \mathit{com}, \mathit{mod}\}$ is then given by $P_\sigma^{\mathcal{F}} = P_\sigma^{\mathrm{ADF}} \cup \{p^{\mathcal{F}}\}$.

Bipolar ADFs. For bipolar ADFs, we use the fact that each of their links must have at least one polarity, that is, must be supporting or attacking. Therefore, if a link is not supporting, it must be attacking, and vice versa.

The propagator for BADFs, given a partial σ-characterization $\mathbb{F}$ and a set of interpretations V that we want to realize, is defined as follows:

$$\begin{aligned} p^{\mathrm{BADF}}(V, \mathbb{F}) = &\{(v|_{\mathbf{t}}^{b}, a, \mathbf{f}) \mid (v, a, \mathbf{f}) \in \mathbb{F}, (w, a, \mathbf{t}) \in \mathbb{F}, w(b) = \mathbf{f}, (w|_{\mathbf{t}}^{b}, a, \mathbf{f}) \in \mathbb{F}\} \cup \\ &\{(v|_{\mathbf{t}}^{b}, a, \mathbf{t}) \mid (v, a, \mathbf{t}) \in \mathbb{F}, (w, a, \mathbf{f}) \in \mathbb{F}, w(b) = \mathbf{f}, (w|_{\mathbf{t}}^{b}, a, \mathbf{t}) \in \mathbb{F}\} \end{aligned}$$

The first line propagates information about attacking links: the tuples $(w, a, \mathbf{t}) \in \mathbb{F}$ and $(w|_{\mathbf{t}}^{b}, a, \mathbf{f}) \in \mathbb{F}$ serve as witnesses that b is not supporting for a, since w and $w|_{\mathbf{t}}^{b}$ differ only in their assignment of b (by definition). Hence, by bipolarity, the link (b, a) must be attacking, which is propagated by adding $(v|_{\mathbf{t}}^{b}, a, \mathbf{f})$ to $\mathbb{F}$ for every $(v, a, \mathbf{f}) \in \mathbb{F}$. The second line propagates the fact that a link (b, a) must be supporting, having observed, by $(w, a, \mathbf{f}) \in \mathbb{F}$, $w(b) = \mathbf{f}$, and $(w|_{\mathbf{t}}^{b}, a, \mathbf{t}) \in \mathbb{F}$, that the link cannot be attacking.

In order to construct the realizing BADF, given a relation $\mathbb{F}$ which is neither incoherent nor partial, we can again make use of the canonical realization $D_{f_{\mathbb{F}}}$, but additionally have to add the link polarities. We can extract the link polarities out of $\mathbb{F}$, denoting the set of all non-attacking links by $L_{\mathbb{F}}^{+}$ and the set of all non-supporting links by $L_{\mathbb{F}}^{-}$, as follows.

Definition 74. Given a semantics $\sigma \in \{\mathit{com}, \mathit{adm}, \mathit{mod}\}$ and a σ-characterization $\mathbb{F}$, we define the canonical realization

$$kb_\sigma^{\mathrm{BADF}}(\mathbb{F}) = (D_{f_{\mathbb{F}}}, L_{\mathbb{F}}^{+}, L_{\mathbb{F}}^{-})$$

with

$$L^+_{\mathbb{F}} = \{(b, a) \mid (v, a, \mathbf{f}) \in \mathbb{F}, v(b) = \mathbf{f}, (v|^b_{\mathbf{t}}, a, \mathbf{t}) \in \mathbb{F}\}$$
$$L^-_{\mathbb{F}} = \{(b, a) \mid (v, a, \mathbf{t}) \in \mathbb{F}, v(b) = \mathbf{f}, (v|^b_{\mathbf{t}}, a, \mathbf{f}) \in \mathbb{F}\}.$$

AFs. Towards the propagator for AFs, recall the ordering $\mathbf{f} <_t \mathbf{t}$ on the Boolean truth values, which, lifted to interpretations, gives rise to the complete lattice $(\mathcal{V}_2, \leq_t)$ with glb $\sqcap_t$ and lub $\sqcup_t$. The lattice has the least element $v_{\mathbf{f}} : A \mapsto \{\mathbf{f}\}$ and the greatest element $v_{\mathbf{t}} : A \mapsto \{\mathbf{t}\}$.

Acceptance conditions of AF-based ADFs have the form of conjunctions of negative literals (cf. Definition 32). In the complete lattice $(\mathcal{V}_2, \leq_t)$, the model sets of AF acceptance conditions correspond to the lattice-theoretic concept of an *ideal*, a non-empty subset of $\mathcal{V}_2$ that is downward-closed with respect to $\leq_t$ and upward-closed with respect to $\sqcup_t$. This is due to the following observations about AFs:

- If an argument a is acceptable under some interpretation $v \in \mathcal{V}_2$, then no argument $b \in v^{\mathbf{t}}$ is attacking a. Hence, for every $v' \in \mathcal{V}_2$ with $v' \leq_t v$, there is no argument $b \in v'^{\mathbf{t}}$ attacking a, therefore a is acceptable under v'.
- If an argument a is acceptable under interpretations $v_1, v_2 \in \mathcal{V}_2$, then neither an argument in $v_1^{\mathbf{t}}$ nor an argument in $v_2^{\mathbf{t}}$ is attacking a. The interpretation $v' = v_1 \sqcup_t v_2$ has exactly $v'^{\mathbf{t}} = v_1^{\mathbf{t}} \cup v_2^{\mathbf{t}}$, i.e. no attacker of a is $\mathbf{t}$ in v', hence a is also acceptable under v'.

The propagator directly implements these closure properties, given a partial σ-characterization $\mathbb{F}$ and a set of interpretations V:

$$p^{\mathrm{AF}}(V, \mathbb{F}) = \{(v_{\mathbf{f}}, a, \mathbf{t}) \mid a \in A\} \cup \tag{3.1}$$
$$\{(w, a, \mathbf{t}) \mid (v, a, \mathbf{t}) \in \mathbb{F}, w \in \mathcal{V}_2, w <_t v\} \cup \tag{3.2}$$
$$\{(w, a, \mathbf{f}) \mid (v, a, \mathbf{f}) \in \mathbb{F}, w \in \mathcal{V}_2, v <_t w\} \cup \tag{3.3}$$
$$\{(v_1 \sqcup_t v_2, a, \mathbf{t}) \mid (v_1, a, \mathbf{t}) \in \mathbb{F}, (v_2, a, \mathbf{t}) \in \mathbb{F}\} \tag{3.4}$$

Application of p^{AF} ensures that when a σ-characterization $\mathbb{F}$ that is neither incoherent nor partial is found in line 8 of Algorithm 1, the set of interpretations $\{v \mid (v, a, \mathbf{t}) \in \mathbb{F}\}$ for each argument $a \in A$ is an ideal wrt. $(\mathcal{V}_2, \leq_t)$. 3.2 and 3.3 ensure downward-closure wrt. $\leq_t$, 3.1 can be derived due to downward-closure and non-emptiness, and 3.4 takes care of upward-closure wrt. $sqcup_t$.

In particular, $\{v \mid (v, a, \mathbf{t}) \in \mathbb{F}\}$ for each argument $a \in A$ has a least upper bound v_a such that $v \leq_t v_a$ iff $(v, a, \mathbf{t}) \in \mathbb{F}$. For the canonical realization, v_a is crucial for the acceptance condition, i.e. the attacks in AF terms, of a.

Definition 75. Given a semantics $\sigma \in \{com, adm, mod\}$ and a σ-characterization $\mathbb{F}$ we define the canonical realization

$$kb_\sigma^{\text{AF}}(\mathbb{F}) = (A, \{(b,a) \mid a, b \in A, v_a(b) = \mathbf{f}\})$$

with v_a such that $v \leq_t v_a$ iff $(v, a, \mathbf{t}) \in \mathbb{F}$ for each $v \in \mathcal{V}_2$.

SETAFs. The propagator for SETAFs, p^{SETAF}, is a weaker version of that of AFs, since we cannot presume upward-closure with respect to $\sqcup_t$ of the model sets of the acceptance conditions of SETAF-based ADFs. In this case every acceptance condition is in conjunctive normal form containing only negative literals (cf. Definition 33). By a transformation preserving logical equivalence one can always obtain an acceptance condition in disjunctive normal form, again with only negative literals; in other words, a disjunction of AF acceptance formulas. Thus, the model set of a SETAF acceptance condition is not necessarily an ideal, but a union of ideals. That means that it is downward-closed with respect to $\leq_t$: if an argument a is acceptable under interpretation v, then there is no $B \subseteq v^{\mathbf{t}}$ such that B attacks a. This also holds for any $v' \leq_t v$, hence a is acceptable under v'.

The propagator implements this closure, coinciding with p^{AF} except for the last part concerned with upward-closure.

$$\begin{aligned} p^{\text{SETAF}}(V, \mathbb{F}) = &\{(v_{\mathbf{f}}, a, \mathbf{t}) \mid a \in A\} \cup \\ &\{(w, a, \mathbf{t}) \mid (v, a, \mathbf{t}) \in \mathbb{F}, w \in \mathcal{V}_2, w <_t v\} \cup \\ &\{(w, a, \mathbf{f}) \mid (v, a, \mathbf{f}) \in \mathbb{F}, w \in \mathcal{V}_2, v <_t w\} \end{aligned}$$

Application of p^{SETAF} ensures that when a σ-characterization $\mathbb{F}$ is found in line 8 of Algorithm 1, the set of interpretations $\{v \mid (v, a, \mathbf{t}) \in \mathbb{F}\}$ for each argument $a \in A$ is downward-closed with respect to $\leq_t$. That means that the maximal elements of this set can be used to construct an appropriate acceptance condition for a in DNF with all negative literals. Transformation to CNF then gives rise to the actual attacking sets.

Definition 76. Given a semantics $\sigma \in \{com, adm, mod\}$ and a σ-characterization $\mathbb{F}$ we define the canonical realization

$$kb_\sigma^{\text{SETAF}}(\mathbb{F}) = (A, \{(\{b_1, \dots, b_n\}, a) \mid a \in A, \{\neg b_1, \dots, \neg b_n\} \in \chi^{\mathbb{F}}\})$$

with $\chi^{\mathbb{F}} = cnf\Big(\bigvee_{v \in \max_{\leq_t}\{w \mid (w,a,\mathbf{t}) \in \mathbb{F}\}} \bigwedge_{v(b)=\mathbf{f}} \neg b\Big)$.

Correctness

In the following we argue that Algorithm 1 is correct, i.e. that, given a set of interpretations V, a formalism $\mathcal{F} \in \{\text{AF}, \text{SETAF}, \text{BADF}, \text{ADF}\}$ and a semantics $\sigma_3 \in \{com_3, adm_3, mod_3\}$, it returns a knowledge base $\mathsf{kb} \in \mathcal{F}$ realizing V in $\mathcal{F}$ under σ_3 if this is possible and "no" otherwise. We begin by showing that the algorithm always terminates and then argue that it is both sound and complete.

Termination. With each recursive call, the set $\mathbb{F}$ can never decrease in size, as the only changes to $\mathbb{F}$ are adding the results of propagation in line 3 and adding the guesses in line 11. Also within the until-loop, the set $\mathbb{F}$ can never decrease in size; furthermore there is only an overall finite number of tuples that can be added to $\mathbb{F}$. Thus at some point we must have $\mathbb{F}_\Delta = \emptyset$ and leave the until-loop. Since $\mathbb{F}$ always increases in size, at some point it must either become functional (and return a realizing knowledge base in line 9) or incoherent (and return "no" in line 5). Therefore the algorithm terminates.

Soundness. If the algorithm returns $kb_\sigma^{\mathcal{F}}(\mathbb{F})$ as a realizing knowledge base, then according to the condition in line 8 the relation $\mathbb{F}$ induces a total function $f_\mathbb{F} : \mathcal{V}_2 \mapsto \mathcal{V}_2$. In particular, because the until-loop must have been run through at least once, there was at least one propagation step (line 2). Since the propagators are defined such that they enforce everything that must hold in a σ-characterization, we conclude that the induced function $f_\mathbb{F}$ indeed is a σ-characterization for V.

Lemma 15. *Let $\sigma \in \{com, adm, mod\}$, $\mathcal{F} \in \{ADF, BADF, SETAF, AF\}$, $V \subseteq \mathcal{V}$, $\mathbb{F} \subseteq \mathcal{V}_2 \times A \times \{\mathbf{t}, \mathbf{f}\}$ such that $\mathbb{F}$ is neither incoherent nor partial, and $f_\mathbb{F}$ is a σ-characterization of V and $p^{\mathcal{F}}(V, \mathbb{F}) \setminus \mathbb{F} = \emptyset$. It holds that $\sigma_3(kb_\sigma^{\mathcal{F}}(\mathbb{F})) = V$.*

Proof. Let $\mathcal{F} =$ ADF. Since $\mathbb{F}$ is a σ-characterization and $kb_\sigma^{\mathrm{ADF}}(\mathbb{F}) = D_{f_\mathbb{F}}$ (cf. Definition 73) it follows from Propositions 26, 29, and 30 $\sigma_3(kb_\sigma^{\mathcal{F}}(\mathbb{F})) = \sigma_3(D_{f_\mathbb{F}}) = V$ for $\sigma_3 = com_3$, $\sigma_3 = adm_3$, and $\sigma_3 = mod_3$, respectively.

For the remaining formalisms $\mathcal{F} \in \{\mathrm{BADF}, \mathrm{SETAF}, \mathrm{AF}\}$ it remains to show that the respective functions $kb_\sigma^{\mathcal{F}}$ are well-defined and that $\sigma_3(kb_\sigma^{\mathcal{F}}(\mathbb{F})) = \sigma_3(D_{f_\mathbb{F}})$.

Let $\mathcal{F} =$ BADF. Since $kb_\sigma^{\mathrm{BADF}}(\mathbb{F})$ is just $D_{f_\mathbb{F}}$ together with additional information (cf. Definition 74), it is clear that $\sigma_3(kb_\sigma^{\mathcal{F}}(\mathbb{F})) = \sigma_3(D_{f_\mathbb{F}})$. To show that $kb_\sigma^{\mathcal{F}}(\mathbb{F})$ is well-defined we have to show that $D_{f_\mathbb{F}}$ is indeed bipolar and $L_\mathbb{F}^-$ and $L_\mathbb{F}^+$ are the non-supporting and non-attacking links of $D_{f_\mathbb{F}}$, respectively. Assume, towards a contradiction, that $D_{f_\mathbb{F}}$ is not bipolar. That means there are arguments $a, b \in A$ such that b is neither supporting a nor attacking a. Let $\varphi_a^{f_\mathbb{F}}$ be the acceptance condition of a in $D_{f_\mathbb{F}}$. We get that there is some $v \in \mathcal{V}$ such that $v(\varphi_a^{f_\mathbb{F}}) = \mathbf{t}$ and $v|_\mathbf{t}^b(\varphi_a^{f_\mathbb{F}}) = \mathbf{f}$ for b not supporting a and there is some $w \in \mathcal{V}$ such that $w(\varphi_a^{f_\mathbb{F}}) = \mathbf{f}$ and $w|_\mathbf{t}^b(\varphi_a^{f_\mathbb{F}}) = \mathbf{t}$ for b not attacking a. By definition of $D_{f_\mathbb{F}}$ this means that $(v, a, \mathbf{t}), (v|_\mathbf{t}^b, a, \mathbf{f}) \in \mathbb{F}$ and also $(w, a, \mathbf{f}), (w|_\mathbf{t}^b, a, \mathbf{t}) \in \mathbb{F}$. From the former we get that $(w|_\mathbf{t}^b, a, \mathbf{f}) \in p^{\mathrm{BADF}}(V, \mathbb{F})$, a contradiction to $p^{\mathrm{BADF}}(V, \mathbb{F}) \setminus \mathbb{F} = \emptyset$ and $\mathbb{F}$ not being incoherent. To show that $L_\mathbb{F}^-$ contains all attacking links of $D_{f_\mathbb{F}}$ assume (b, a) is attacking. This means that there is no $v \in \mathcal{V}_2$ such that $(v, a, \mathbf{t}) \in \mathbb{F}$, $v(b) = \mathbf{f}$, and $(v|_\mathbf{t}^b, a, \mathbf{t}) \in \mathbb{F}$, hence $(b, a) \notin L_\mathbb{F}^+$. Assuming (b, a) is supporting we similarly get that $(b, a) \notin L_\mathbb{F}^-$.

Let $\mathcal{F} =$ AF. To show that kb_σ^{AF} is well-defined we need to show that for each $a \in A$ there is a unique interpretation v_a with $v \leq_t v_a$ iff $(v, a, \mathbf{t}) \in \mathbb{F}$ for each $v \in \mathcal{V}_2$. In other words, the set of interpretations $\mathbb{F}_a^\mathbf{t} = \{v \mid (v, a, \mathbf{t}) \in \mathbb{F}\}$ is an ideal wrt. $(\mathcal{V}_2, \leq_t)$. First, $\mathbb{F}_a^\mathbf{t}$ is non-empty since, by $p^{\mathrm{AF}}(V, \mathbb{F}) \setminus \mathbb{F} = \emptyset$, we must have $(v_\mathbf{f}, a, \mathbf{t}) \in \mathbb{F}$. Now assume $\mathbb{F}_a^\mathbf{t}$ is not downward-closed wrt. $\leq_t$. That means there are interpretations $v, v' \in \mathcal{V}_2$ with $v' <_t v$ such that $(v, a, \mathbf{t}) \in \mathbb{F}$ and $(v, a, \mathbf{t}) \notin \mathbb{F}$. But this is a contradiction to $p^{\mathrm{AF}}(V, \mathbb{F}) \setminus \mathbb{F} = \emptyset$ since we get $(v, a, \mathbf{t}) \in p^{\mathrm{AF}}(V, \mathbb{F})$. Finally assume $\mathbb{F}_a^{\mathbf{t}}$ is not upward-closed wrt. $\sqcup_t$, i.e. there are $v_1, v_2 \in \mathcal{V}_2$ with $(v_1, a, \mathbf{t}), (v_2, a, \mathbf{t}) \in \mathbb{F}$ but $(v_1 \sqcup_t v_2, a, \mathbf{t}) \notin \mathbb{F}$. Again we arrive at a contradiction to $p^{\mathrm{AF}}(V, \mathbb{F}) \setminus \mathbb{F} = \emptyset$ since we get $(v_1 \sqcup_t v_2, a, \mathbf{t}) \in p^{\mathrm{AF}}(V, \mathbb{F})$. Hence $\mathbb{F}_a^{\mathbf{t}}$ is an ideal and therefore we have a unique v_a with $v \leq_t v_a$ iff $(v, a, \mathbf{t}) \in \mathbb{F}$ for each $v \in \mathcal{V}_2$. Considering $kb_\sigma^{\mathrm{AF}}(\mathbb{F})$ we observe that the ADF associated to it, $D_{kb_\sigma^{\mathrm{AF}}(\mathbb{F})}$ (cf. Definition 32), the acceptance condition of argument a is $\varphi_a = \bigwedge_{v_a(b)=\mathbf{f}} \neg b$. The satisfying interpretations of this formula are just these $v \in \mathcal{V}_2$ with $v \leq_t v_a$, coinciding with those of the acceptance condition of $D_{f_\mathbb{F}}$ (cf. Definition 68). Therefore, recalling from Proposition 3 that $\sigma_3(kb_\sigma^{\mathrm{AF}}(\mathbb{F})) = \sigma_3(D_{kb_\sigma^{\mathrm{AF}}(\mathbb{F})})$, we get $\sigma_3(kb_\sigma^{\mathrm{AF}}(\mathbb{F})) = \sigma_3(D_{f_\mathbb{F}})$.

Let $\mathcal{F} = \mathrm{SETAF}$. The canonical realization $kb_\sigma^{\mathrm{SETAF}}$ is a SETAF by definition. It remains to show that $\sigma_3(kb_\sigma^{\mathrm{SETAF}}(\mathbb{F})) = \sigma_3(D_{f_\mathbb{F}})$. To this end consider an arbitrary argument $a \in A$ and let $\mathbb{F}_a^{\mathbf{t}} = \{v \mid (v, a, \mathbf{t}) \in \mathbb{F}\}$. First observe that $\mathbb{F}_a^{\mathbf{t}}$ is downward-closed wrt. $\leq_t$: for any $(v, a, \mathbf{t}) \in \mathbb{F}$ we get $(w, a, \mathbf{t}) \in p^{\mathrm{SETAF}}$ for each $w <_t v$; therefore, by the assumption that $p^{\mathrm{SETAF}}(V, \mathbb{F}) \setminus \mathbb{F} = \emptyset$, also $(w, a, \mathbf{t}) \in \mathbb{F}$ for each $w <_t v$. Therefore we get, letting $\varphi_a^{f_\mathbb{F}}$ be a's acceptance condition in $D_{f_\mathbb{F}}$ and φ_a be a's acceptance condition in $D_{kb_\sigma^{\mathrm{SETAF}}(\mathbb{F})}$, that $\varphi_a^{f_\mathbb{F}} \equiv \bigvee_{v \in \max_{\leq_t} \mathbb{F}_a^{\mathbf{t}}} \bigwedge_{v(b)=\mathbf{f}} \neg b \equiv cnf\Big(\bigvee_{v \in \max_{\leq_t} \mathbb{F}_a^{\mathbf{t}}} \bigwedge_{v(b)=\mathbf{f}} \neg b\Big) = \varphi_a$. Recalling from Proposition 5 that $\sigma_3(kb_\sigma^{\mathrm{AF}}(\mathbb{F})) = \sigma_3(D_{kb_\sigma^{\mathrm{AF}}(\mathbb{F})})$ it follows that $\sigma_3(kb_\sigma^{\mathrm{AF}}) = \sigma_3(D_{f_\mathbb{F}})$. $\square$

Hence we can conclude by Lemma 15 that $\sigma_3(kb_\sigma^{\mathcal{F}}(\mathbb{F})) = V$.

Completeness. If the algorithm answers "no", then the execution reached line 5. Thus, for the constructed relation $\mathbb{F}$, there must have been an interpretation $v \in \mathcal{V}_2$ and an argument $a \in A$ such that $\{(v, a, \mathbf{t}), (v, a, \mathbf{f})\} \subseteq \mathbb{F}$, that is, $\mathbb{F}$ is incoherent. Note that the guessing step cannot directly create incoherence, since exactly one truth value is guessed for v and a. Therefore, since $\mathbb{F}$ is initially empty, the only way it could get incoherent is in the propagation step in line 2. However, the propagators are defined such that they infer only tuples that are necessary for the given $\mathbb{F}$. Consequently, the given interpretation-set V is such that either there is no realization within the ADF fragment corresponding to formalism $\mathcal{F}$ (that is, the formalism propagator $p^{\mathcal{F}}$ derived the incoherence) or there is no σ-characterization of V with respect to general ADFs (that is, a semantics propagator in P_σ^{ADF} derived the incoherence). In any case, V is not σ_3-realizable for $\mathcal{F}$.

3.6.2 Implementation

As Algorithm 1 is based on propagation, guessing, and checking it is perfectly suited for an implementation using answer set programming [159, 151]. By that we can make use of conflict learning strategies and heuristics of modern ASP solvers [127]. Thus, we developed ASP encodings in the `gringo` language [128] for our approach. Similar as the algorithm, our declarative encodings are modular, consisting of a main part responsible for constructing the σ-characterization candidate $\mathbb{F}$ and separate encodings for the individual propagators. If one wants, e.g., to compute an AF realization under complete semantics for a set V of interpretations, an input program encoding V is joined with the main encoding, the propagator encoding for complete semantics as well as the propagator encoding for AFs. Every answer set of such a program encodes a respective characterization function. Our ASP encoding for preferred semantics is based on the admissible encoding and guesses further interpretations following the essential idea of Algorithm 2.

For constructing a knowledge base under the desired semantics in the desired formalism, we also provide two further ASP encodings: one that transforms the output to an ADF in the syntax of the `DIAMOND` tool [116, 117] (following the construction in Definition 73), and one that transforms the output to an AF in `ASPARTIX` syntax [113, 122] (following the construction in Definition 75). Both argumentation tools are based on ASP themselves.

The encodings for all the semantics and formalisms we covered in this work are available as the software system `UNREAL`. The system as well as the single encodings can be downloaded from `http://www.dbai.tuwien.ac.at/research/project/adf/unreal/`.

3.6.3 Expressiveness Results

In this section we present some results that we have obtained using our implementation. We will first show that, for a fixed semantics, there is a strict subset relation for the signatures of the semantics in AFs and SETAFs, SETAFs and bipolar ADFs, and bipolar ADFs and ADFs, respectively. Then we will see that, for different semantics, all signatures are incomparable, also complementing the relations of signatures in ADFs from Theorem 25.

We start by considering the signatures of AFs, SETAFs, BADFs and ADFs for the unary vocabulary.

Example 42. Let, for semantics σ_3 and formalism $\mathcal{F}$ and a set of arguments A, $\Sigma^{\sigma_3}_{\mathcal{F}_{\{a\}}}$ be the signature of σ_3 in $\mathcal{F}$ restricted to interpretations over A. Now consider $A = \{a\}$. In this case, every SETAF $((\{a\}, \emptyset)$ and $(\{a\}, \{(\{a\}, a)\}))$ is also an AF, and every ADF is also bipolar since the link (a, a) is[13] either supporting ($\varphi_a = a$) or attacking ($\varphi_a = \neg a$).

[13]Disregarding tautological (sub)formulas, which can also make the link both supporting and attacking.

$$
\begin{aligned}
\Sigma^{adm_3}_{\mathrm{AF}_{\{a\}}} &= \Sigma^{adm_3}_{\mathrm{SETAF}_{\{a\}}} = \{\{\mathbf{u}\}, \{\mathbf{u}, \mathbf{t}\}\} \\
\Sigma^{com_3}_{\mathrm{AF}_{\{a\}}} &= \Sigma^{com_3}_{\mathrm{SETAF}_{\{a\}}} = \{\{\mathbf{u}\}, \{\mathbf{t}\}\} \\
\Sigma^{prf_3}_{\mathrm{AF}_{\{a\}}} &= \Sigma^{prf_3}_{\mathrm{SETAF}_{\{a\}}} = \{\{\mathbf{u}\}, \{\mathbf{t}\}\} \\
\Sigma^{mod_3}_{\mathrm{AF}_{\{a\}}} &= \Sigma^{mod_3}_{\mathrm{SETAF}_{\{a\}}} = \{\emptyset, \{\mathbf{t}\}\} \\
\Sigma^{adm_3}_{\mathrm{ADF}_{\{a\}}} &= \Sigma^{adm_3}_{\mathrm{BADF}_{\{a\}}} = \Sigma^{adm_3}_{\mathrm{AF}_{\{a\}}} \cup \{\{\mathbf{u}, \mathbf{f}\}, \{\mathbf{u}, \mathbf{t}, \mathbf{f}\}\} \\
\Sigma^{com_3}_{\mathrm{ADF}_{\{a\}}} &= \Sigma^{com_3}_{\mathrm{BADF}_{\{a\}}} = \Sigma^{com_3}_{\mathrm{AF}_{\{a\}}} \cup \{\{\mathbf{f}\}, \{\mathbf{u}, \mathbf{t}, \mathbf{f}\}\} \\
\Sigma^{prf_3}_{\mathrm{ADF}_{\{a\}}} &= \Sigma^{prf_3}_{\mathrm{BADF}_{\{a\}}} = \Sigma^{prf_3}_{\mathrm{AF}_{\{a\}}} \cup \{\{\mathbf{f}\}, \{\mathbf{t}, \mathbf{f}\}\} \\
\Sigma^{mod_3}_{\mathrm{ADF}_{\{a\}}} &= \Sigma^{mod_3}_{\mathrm{BADF}_{\{a\}}} = \Sigma^{mod_3}_{\mathrm{AF}_{\{a\}}} \cup \{\{\mathbf{f}\}, \{\mathbf{t}, \mathbf{f}\}\}
\end{aligned}
$$

The following result shows that the expressiveness of the formalisms under consideration is in line with the amount of restrictions they impose on acceptance conditions.

Theorem 27. *For any $\sigma_3 \in \{adm_3, com_3, prf_3, mod_3\}$:*

1. $\Sigma^{\sigma_3}_{AF} \subset \Sigma^{\sigma_3}_{SETAF}$;
2. $\Sigma^{\sigma_3}_{SETAF} \subset \Sigma^{\sigma_3}_{BADF}$;
3. $\Sigma^{\sigma_3}_{BADF} \subset \Sigma^{\sigma_3}_{ADF}$.

Proof. (1) $\Sigma^{\sigma_3}_{\mathrm{AF}} \subseteq \Sigma^{\sigma_3}_{\mathrm{SETAF}}$ is clear since every AF can be faithfully translated to a SETAF (by modeling individual attacks as attacks by singletons). For $\Sigma^{\sigma_3}_{\mathrm{SETAF}} \not\subseteq \Sigma^{\sigma_3}_{AF}$ the witnessing interpretation-sets over vocabulary $A = \{a, b, c\}$ are

- $\{\mathbf{uuu}, \mathbf{ttf}, \mathbf{tft}, \mathbf{ftt}\} \in \Sigma^{\sigma_3}_{\mathrm{SETAF}} \setminus \Sigma^{\sigma_3}_{\mathrm{AF}}$ for $\sigma_3 \in \{adm_3, com_3\}$, and
- $\{\mathbf{ttf}, \mathbf{tft}, \mathbf{ftt}\} \in \Sigma^{\tau_3}_{\mathrm{SETAF}} \setminus \Sigma^{\tau_3}_{\mathrm{AF}}$ for $\tau_3 \in \{prf_3, mod_3\}$.

By each pair of arguments of A being $\mathbf{t}$ in at least one model, a realizing AF cannot feature any attack, immediately giving rise to the model $\mathbf{ttt}$ under all semantics under consideration. The respective realizing SETAF, on the other hand, is given by the attack relation $X = \{(\{a, b\}, c), (\{a, c\}, b), (\{b, c\}, a)\}$.

(2) It is clear that $\Sigma^{\sigma_3}_{\mathrm{SETAF}} \subseteq \Sigma^{\sigma_3}_{\mathrm{BADF}}$ holds, since SETAFs are bipolar as all parents are always attacking (see also Proposition 5). For $\Sigma^{\sigma_3}_{\mathrm{BADF}} \not\subseteq \Sigma^{\sigma_3}_{\mathrm{SETAF}}$ the respective counterexamples can be read off the signatures restricted to the unary vocabulary from Example 42: for $\sigma_3 \in \{adm_3, com_3\}$ we find $\{\mathbf{u}, \mathbf{t}, \mathbf{f}\} \in \Sigma^{\sigma_3}_{\mathrm{BADF}} \setminus \Sigma^{\sigma_3}_{\mathrm{SETAF}}$ and for $\tau_3 \in \{prf_3, mod_3\}$ we find $\{\mathbf{t}, \mathbf{f}\} \in \Sigma^{\tau_3}_{\mathrm{BADF}} \setminus \Sigma^{\tau_3}_{\mathrm{SETAF}}$. The realizing bipolar ADF has acceptance condition $\varphi_a = a$.

(3) For $\sigma_3 = mod_3$ the result was shown by Strass [188, Theorem 14]; for the remaining semantics the interpretation-sets over vocabulary $A = \{a, b\}$ witnessing $\Sigma^{\sigma_3}_{\mathrm{ADF}} \not\subseteq \Sigma^{\sigma_3}_{\mathrm{BADF}}$ are

$$\{\mathbf{uu}, \mathbf{tu}, \mathbf{tt}, \mathbf{tf}, \mathbf{fu}\} \in \Sigma^{adm_3}_{\mathrm{ADF}} \setminus \Sigma^{adm_3}_{\mathrm{BADF}}$$
$$\{\mathbf{uu}, \mathbf{tu}, \mathbf{tt}, \mathbf{tf}, \mathbf{fu}\} \in \Sigma^{com_3}_{\mathrm{ADF}} \setminus \Sigma^{com_3}_{\mathrm{BADF}}$$
$$\{\mathbf{tt}, \mathbf{tf}, \mathbf{fu}\} \in \Sigma^{prf_3}_{\mathrm{ADF}} \setminus \Sigma^{prf_3}_{\mathrm{BADF}}$$

A witnessing ADF for all three semantics is given by $\varphi_a = a$ and $\varphi_b = (a \wedge \neg b) \vee (\neg a \wedge b)$ (note that both links (a, b) and (b, b) are neither attacking nor supporting). $\square$

Theorem 27 is concerned with the relative expressiveness of the formalisms under consideration, given a certain semantics. Considering different semantics we find that for all formalisms the signatures become incomparable:

Proposition 31. *It holds that $\Sigma^{\sigma_3}_{\mathcal{F}_1} \not\subseteq \Sigma^{\tau_3}_{\mathcal{F}_2}$ and $\Sigma^{\tau_3}_{\mathcal{F}_2} \not\subseteq \Sigma^{\sigma_3}_{\mathcal{F}_1}$ for all formalisms $\mathcal{F}_1, \mathcal{F}_2 \in \{AF, SETAF, BADF, ADF\}$ and all semantics $\sigma_3, \tau_3 \in \{adm_3, com_3, prf_3, mod_3\}$ with $\sigma_3 \neq \tau_3$.*

Proof. First, the result for $\sigma_3 = adm_3$ and $\tau_3 = com_3$ follows from $\{\mathbf{u}, \mathbf{t}\} \in \Sigma^{adm_3}_{\mathrm{AF}}$ (realized by the AF $(\{a\}, \emptyset)$), but $\{\mathbf{u}, \mathbf{t}\} \notin \Sigma^{com_3}_{\mathrm{ADF}}$ and the $\subset$-relations from Theorem 27. Similarly, for $\sigma_3 = com_3$ and $\tau_3 = adm_3$ we get the result by $\{\mathbf{t}\} \in \Sigma^{com_3}_{\mathrm{AF}}$, but $\{\mathbf{t}\} \notin \Sigma^{adm_3}_{\mathrm{ADF}}$.

Taking into account that the set of preferred interpretations (resp. two-valued models) is always incompatible for ADFs (and therefore also for the other formalisms) while the set of admissible (resp. complete) interpretations is never incompatible (given its size is at least 2), the result follows for $\sigma_3 \in \{adm_3, com_3\}$ and $\tau_3 \in \{prf_3, mod_3\}$ (e.g. $\{\mathbf{uu}, \mathbf{tf}, \mathbf{ft}\} \in \Sigma^{\sigma_3}_{\mathrm{AF}} \setminus \Sigma^{\tau_3}_{\mathrm{ADF}}$) as well as for $\sigma_3 \in \{prf_3, mod_3\}$ and $\tau_3 \in \{adm_3, com_3\}$ (e.g. $\{\mathbf{tf}, \mathbf{ft}\} \in \Sigma^{\sigma_3}_{\mathrm{AF}} \setminus \Sigma^{\tau_3}_{\mathrm{ADF}}$).

Finally, since a $\mathsf{kb} \in \mathcal{F}$ for any $\mathcal{F} \in \{\mathrm{AF}, \mathrm{SETAF}, \mathrm{BADF}, \mathrm{ADF}\}$ may not have any two-valued models and a preferred interpretation is not necessarily two-valued (e.g. $prf_3((\{a\}, \{(a, a)\})) = \{\mathbf{u}\}$), the result for prf_3 and mod_3 follows. $\square$

Disregarding the possibility of realizing the empty set of interpretations under the two-valued model semantics, we obtain the following relation for ADFs.

Proposition 32. $(\Sigma^{mod_3}_{ADF} \setminus \{\emptyset\}) \subseteq \Sigma^{prf_3}_{ADF}$.

Proof. Consider some $V \in \Sigma^{mod_3}_{\mathrm{ADF}}$ with $V \neq \emptyset$. Clearly $V \subseteq \mathcal{V}_2$ and by Proposition 30 there is a *mod*-characterization $f : \mathcal{V}_2 \mapsto \mathcal{V}_2$ of V, that is, $f(v) = v$ iff $v \in V$. Define $f' : \mathcal{V}_2 \mapsto \mathcal{V}_2$ such that $f'(v) = f(v) = v$ for all $v \in V$ and $f'(v)(a) = \neg v(a)$ for all $v \in \mathcal{V}_2 \setminus V$ and $a \in A$. It holds that f' is an *adm*-characterization of $V' = \{v \in \mathcal{V} \mid \forall v_2 \in [v]_2 : v_2 \in V\} \cup \{v_{\mathbf{u}}\}$: If, for some $v \in \mathcal{V}$, $\forall v_2 \in [v]_2 : v_2 \in V$ then also $\forall v_2 \in [v]_2 \forall a \in A : f'(v_2)(a) = v(a)$, i.e.

the condition from Definition 70 is fulfilled. On the other hand, if $\exists v_2 \in [v]_2 : v_2 \notin V$, then, for this $v_2 \in [v]_2$, it holds that $f'(v_2)(a) = \neg v_2(a)$ for all $a \in A$. Since $v \neq v_{\mathbf{u}}$, there is some $a \in A$ such that $f'(v_2)(a) \neq v(a)$, i.e. the condition is violated. Since $\max_{\leq_i} V' = V$ we get that the ADF D with acceptance formula $\varphi_a^{f'}$ for each $a \in A$ has $prf_3(D) = V$. Therefore $V \in \Sigma_{\mathrm{ADF}}^{prf_3}$. $\square$

In contrast, this relation does not hold for AFs. This follows directly from the result for the extension-based case in Theorem 12.5. For SETAFs and BADFs we have to leave the question open.

CHAPTER 4

Revision

In this chapter we study one of the central topics in the dynamics of argumentation, namely revision. That is concerned with changing a given knowledge base, in the form of an argumentation formalism, in the light of new information. We follow the prominent approach of AGM revision [4, 141] and aim for adherence of the postulates proposed there, reformulated for our setting. In the setting we study, the new information is provided in the same formalism as the original knowledge base, following other approaches to revision in fragments [84, 85]. In contrast to previous work on the revision of AFs by Coste-Marquis et al. [76, 77], we require the revision to result in a single knowledge base (that is, a single AF or a single ADF, respectively). For that, results on expressiveness of the respective formalisms presented in Chapter 3 will turn out to be essential.

The remainder of this chapter is organized as follows. We begin by introducing the main aspects of AGM revision in Section 4.1. Then, in Section 4.2, we study revision of AFs, beginning with a discussion of why this turns out to be more involved than revision of propositional formulas, and then presenting a representation result for revision operators under the general class of proper I-maximal semantics. Section 4.3 deals with the revision of ADFs and introduces a hybrid approach, combining admissible and preferred semantics. Then, in Section 4.4, we will study the complexity of Dalal's operator [80] for revision of AFs, showing completeness results up to Θ_3^{P}. Finally, we discuss some related issues in Section 4.5.

The results in Sections 4.2 and 4.4 are published in [89], of which an extended version is currently under review (a technical report [91] is available). Section 4.3 is based on [148].

4.1 Preliminaries

The most prominent approach to belief change was introduced by Alchourrón et al. [4] and is commonly known as AGM revision. It deals with revising knowledge bases in the light of new, possibly contradicting, information. The approach was reformulated for knowledge bases in the form of propositional formulas by Katsuno and Mendelzon [141]. Katsuno and Mendelzon define an equivalent version of the AGM *rationality postulates* [4, 125] for operators $\circ : \mathsf{P}_{\mathfrak{A}} \times \mathsf{P}_{\mathfrak{A}} \mapsto \mathsf{P}_{\mathfrak{A}}$ mapping pairs of formulas to revised formulas.[1] The application of $\psi \circ \mu$ to two formulas ψ and μ has the intended meaning that the original knowledge base, in the form of the propositional formula ψ, is revised by μ, the revising formula, returning the revised knowledge base again as a propositional formula. The postulates (KM1)–(KM6) specify properties which a revision operator should satisfy in order to be considered rational.

(KM1) $\psi \circ \mu \models \mu$.

(KM2) If $\psi \wedge \mu$ is satisfiable, then $\psi \circ \mu = \psi \wedge \mu$.

(KM3) If μ is satisfiable, then $\psi \circ \mu$ is also satisfiable.

(KM4) If $\psi_1 \equiv \psi_2$ and $\mu_1 \equiv \mu_2$, then $\psi_1 \circ \mu_1 \equiv \psi_2 \circ \mu_2$.

(KM5) $(\psi \circ \mu) \wedge \phi \models \psi \circ (\mu \wedge \phi)$.

(KM6) If $(\psi \circ \mu) \wedge \phi$ is satisfiable, then $\psi \circ (\mu \wedge \phi) \models (\psi \circ \mu) \wedge \phi$.

(KM1) says that the revising formula must be retained in the revised knowledge base. If original and revising formula are consistent, then, by (KM2), the result of the revision is obtained in the obvious way. (KM3) ensures that the result of the revision is satisfiable, unless the revising formula was inconsistent. (KM4) represents Dalal's *principle of irrelevance of syntax* [80]. Finally, (KM5) and (KM6) ensure that revision is performed with minimal change.

The main result of [141] is that there is a one-to-one correspondence between

- operators which are rational according to the AGM postulates, and
- operators obtained from mappings of formulas to certain rankings over interpretations.

Intuitively, a ranking over interpretations for a formula describes a preference relation among the interpretations. The corresponding operator shall then return a formula which has, as its own models, these models of the revision formula, which are most preferred according to the ranking. Given a ranking in the form of a preorder $\preceq$, the construction of the corresponding operator is then based on the following selection function, given an arbitrary set of interpretations $\mathbb{I}$:

$$\min(\mathbb{I}, \preceq) = \{I_1 \in \mathbb{I} \mid \nexists I_2 \in \mathbb{I} : I_2 \prec I_1\}.$$

[1] Recall that $\mathsf{P}_{\mathfrak{A}}$ denotes the collection of all propositional formulas over $\mathfrak{A}$.

The correspondence from rational operators is of course not to operators obtained from any kind of ranking over interpretations, but to what is called faithful rankings.

Definition 77. Given a propositional formula ψ, a *faithful ranking* for ψ is a total preorder $\preceq_\psi$ on $2^{\mathfrak{A}}$ such that, for any $I_1, I_2 \in 2^{\mathfrak{A}}$, it holds that

(*i*) if $I_1, I_2 \in \mathsf{Mod}(\psi)$ then $I_1 \approx_\psi I_2$, and

(*ii*) if $I_1 \in \mathsf{Mod}(\psi)$ and $I_2 \notin \mathsf{Mod}(\psi)$ then $I_1 \prec_\psi I_2$.

A *faithful assignment* maps every formula ψ to a faithful ranking $\preceq_\psi$ for ψ such that

(*iii*) $\preceq_{\psi_1} = \preceq_{\psi_2}$ for any formulas ψ_1, ψ_2 with $\psi_1 \equiv \psi_2$.

It turns out that faithful assignments not only give rise to operators which are rational according to postulates (KM1)–(KM6), but also exactly characterize such operators, i.e. each operator has a corresponding faithful assignment. This is captured in the following theorem, which can be regarded as the main insight of AGM revision of propositional knowledge bases.

Theorem 28 ([141]). *A revision operator* $\circ : \mathsf{P}_{\mathfrak{A}} \times \mathsf{P}_{\mathfrak{A}} \mapsto \mathsf{P}_{\mathfrak{A}}$ *satisfies postulates* (*KM*1)–(*KM*6) *if and only if there exists a faithful assignment mapping each formula* ψ *to a faithful ranking* $\preceq_\psi$ *for* ψ *such that*

$$\mathsf{Mod}(\psi \circ \phi) = \min(\mathsf{Mod}(\phi), \preceq_\psi)$$

holds for every formula ϕ.

An important aspect of Theorem 28 is the fact that, in propositional logic, any set of interpretations has a syntactic counterpart, i.e. is realizable in our terms. Therefore any desired outcome of the revision in terms of models, given by $\min(\mathsf{Mod}(\phi), \preceq_\psi)$, can indeed be achieved by $\psi \circ \phi$. In the upcoming sections, we will have to deal with limited expressiveness of the formalisms, which will also affect the kinds of representation results we will be able to obtain.

Abstracting away from propositional formulas to revision of knowledge bases in arbitrary formalisms, we will use the notion of an operator being induced by a ranking in the following way.

Definition 78. Let $\mathcal{F}$ be a formalism and σ a semantics of that formalism. Given a preorder $\preceq$ on the possible interpretations of $\mathcal{F}$ the *operator induced by* $\preceq$ is given by

$$\mathsf{kb} \circ \mathsf{kb}' = \rho_\sigma^{\mathcal{F}}(\min(\sigma(\mathsf{kb}'), \preceq)),$$

for arbitrary $\mathsf{kb}, \mathsf{kb}' \in \mathcal{F}$.

At this point recall the realizing function $\rho_\sigma^{\mathcal{F}}$ for a formalism $\mathcal{F}$ and a semantics σ of that formalism given in Definition 37, which returns, given some $\mathbb{I} \in \Sigma_{\mathcal{F}}^{\sigma}$, a realizing knowledge base $\mathsf{kb} \in \mathcal{F}$ with $\sigma(\mathsf{kb}) = \mathbb{I}$.

One of the most prominent concrete revision operators was introduced by Dalal [80]. Dalal's operator satisfies all rationality postulates and is therefore characterizable by faithful assignments. The ranking giving rise to the operator is based on the Hamming distance [134] between interpretations.

Definition 79. Given two-valued interpretations I_1 and I_2, their *Hamming distance* $\triangle_H$ is defined as

$$I_1 \triangle_H I_2 = |(I_1 \setminus I_2) \cup (I_2 \setminus I_1)|.$$

Dalal's operator then gives preference to interpretations with minimal distance to some model of the revising formula. The following is a generalization of the operator to arbitrary formalism under two-valued semantics.

Definition 80. Let $\mathcal{F}$ be a formalism and σ a semantics of $\mathcal{F}$ based on two-valued interpretations. Given an arbitrary knowledge base kb, the ranking $\preceq_{\mathsf{kb}}^{\sigma}$ is defined as

$$I_1 \preceq_{\mathsf{kb}}^{\sigma} I_2 \text{ if and only if } \min_{I \in \sigma(\mathsf{kb})} (I \triangle_H I_1) \leq \min_{I \in \sigma(\mathsf{kb})} (I \triangle_H I_2).$$

Dalal's operator $\circ_\sigma^D$ (induced by $\preceq_{\mathsf{kb}}^{D}$) returns $\mathsf{kb} \circ_\sigma^D \mathsf{kb}' = \rho_\sigma^{\mathcal{F}}(\min(\sigma(\mathsf{kb}'), \preceq_{\mathsf{kb}}^{\sigma}))$ for each $\mathsf{kb}' \in \mathcal{F}$.

4.2 Revision of AFs

In this section we consider revision of an AF by another AF. This approach is not only in line with other work on revision in fragments such as Horn theories [84] or logic programs [85], but follows the idea that knowledge (e.g. of agents) is exclusively represented within AFs.

The revision we deal with in this section is performed by operators of the type $*_\sigma : AF_{\mathfrak{A}} \times AF_{\mathfrak{A}} \mapsto AF_{\mathfrak{A}}$, where σ is a semantics. Such operators map an AF F (the original AF) and another AF G (the revising AF) to the revised AF $F *_\sigma G$. The intuitive idea is that the new information provided by G (given by the σ-extensions of G) is incorporated by bringing only minimal change to F. Therefore, the underlying concept of a model is now given by the argumentation semantics σ.

The rationality postulates (A1$_\sigma$)–(A6$_\sigma$) for revision of AFs, which can again be seen as requirements for an operator $*$ to be considered rational, are obtained from a reformulation of the postulates for knowledge base revision [141]. They are parametrized by the underlying semantics σ and formulated as follows.

(A1$_\sigma$) $\sigma(F * G) \subseteq \sigma(G)$.

(A2$_\sigma$) If $\sigma(F) \cap \sigma(G) \neq \emptyset$, then $\sigma(F * G) = \sigma(F) \cap \sigma(G)$.

(A3$_\sigma$) If $\sigma(G) \neq \emptyset$, then $\sigma(F * G) \neq \emptyset$.

(A4$_\sigma$) If $\sigma(F_1) = \sigma(F_2)$ and $\sigma(G) = \sigma(H)$, then $\sigma(F_1 * G) = \sigma(F_2 * H)$.

(A5$_\sigma$) $\sigma(F * G) \cap \sigma(H) \subseteq \sigma(F * \rho_\sigma^{\mathrm{AF}}(\sigma(G) \cap \sigma(H)))$.

(A6$_\sigma$) If $\sigma(F *_\sigma G) \cap \sigma(H) \neq \emptyset$, then $\sigma(F * \rho_\sigma^{\mathrm{AF}}(\sigma(G) \cap \sigma(H))) \subseteq \sigma(F * G) \cap \sigma(H)$.

For postulates (A5$_\sigma$) and (A6$_\sigma$) recall from Definition 38 that the realizing function $\rho_\sigma^{\mathrm{AF}}(\mathbb{S})$, given an extension-set $\mathbb{S}$, returns an AF having extensions $\mathbb{S}$ under σ if $\mathbb{S}$ is σ-realizable, and the empty AF otherwise.

The postulates make clear that the result of the revision depends solely on the semantic evaluation (i.e. the extensions) of the revising AF and not on its syntactic form, following the principle of irrelevance of syntax.

Our aim is to characterize rational operators by certain orderings of the possible extensions and get representation results similar to the case of propositional logic. We can overload the notions of faithful rankings and faithful assignments for the AF setting:

Definition 81. Given a semantics σ and an AF F, a *faithful ranking* for F is a total preorder $\preceq_F$ on $2^{\mathfrak{A}}$ such that, for any $E_1, E_2 \in 2^{\mathfrak{A}}$, it holds that

(*i*) if $E_1, E_2 \in \sigma(F)$ then $E_1 \approx_F E_2$, and

(*ii*) if $E_1 \in \sigma(F)$ and $E_2 \notin \sigma(F)$ then $E_1 \prec_F E_2$.

A *faithful assignment* maps every AF F to an faithful ranking $\preceq_F$ for F such that

(*iii*) $\preceq_{F_1} = \preceq_{F_2}$ for any AFs F_1, F_2 with $\sigma(F_1) = \sigma(F_2)$.

It is important to note that also for assigning faithful rankings to AFs, the syntax of AFs must not be taken into account.

We are striving for functions $*$ defined in terms of faithful rankings by

$$F * G = \rho_\sigma^{\mathrm{AF}}(\min(\sigma(G)), \preceq_F))$$

for AFs F and G, such than we can obtain a correspondence similar to Theorem 28 to functions satisfying the postulates. Recall at this point the realizing function $\rho_\sigma^{\mathrm{AF}}$ from Definition 38 which returns, given an extension-set $\mathbb{S}$, an AF realizing $\mathbb{S}$ under σ whenever $\mathbb{S} \in \Sigma_{\mathrm{AF}}^{\sigma}$. By our thorough investigations in Section 3.2 it is, however, clear that the definition of $*$ above is suggestive. This is because $\min(\sigma(F)), \preceq_F)$ is not necessarily realizable under σ and therefore $\rho_\sigma^{\mathrm{AF}}$ might not behave as expected.

It already becomes apparent that the restricted expressiveness of argumentation semantics compared to propositional logic, where every set of models has a corresponding formula, causes some problems in representing rational revision operators. We will now identify some specific problems and outline our respective solutions.

Non-existence of rational operators. Inspecting the rationality postulates one can see that (A2$_\sigma$) puts a severe requirement on the expressiveness of the semantics under which revision is to be done. In order for a revision operator $*$ to fulfill (A2$_\sigma$), the signature of the semantics σ has to be closed under intersection. That means that, for arbitrary AFs F and G, it must hold that $\sigma(F) \cap \sigma(G) \in \Sigma^{\sigma}_{\mathrm{AF}}$ in order to be able to achieve $\sigma(F * G) = \sigma(F) \cap \sigma(G)$.

We have already studied this property in Section 3.2. While the majority of semantics studied in this work enjoy a signature which is closed under intersection (cf. Theorem 6), we showed in Example 16 that it is not the case for complete semantics. We are able to state the following impossibility result.

Theorem 29. *There exists no operator* $*\colon AF_{\mathfrak{A}} \times AF_{\mathfrak{A}} \mapsto AF_{\mathfrak{A}}$ *that satisfies* ($A2_{com}$).

Proof. We have seen in Example 16 that there are AFs F and G such that $com(F) \cap com(G) \neq \emptyset$ and $com(F) \cap com(G) \notin \Sigma^{com}_{\mathrm{AF}}$. Now for $*$ to satisfy (A2$_{com}$), it must hold that $F * G \in AF_{\mathfrak{A}}$ with $com(F * G) = com(F) \cap com(G)$, which is not possible. Hence the result follows. $\square$

We will discuss this form of impossibility and show it in a more general manner in Section 4.5.

Limited expressiveness. If the signature of the semantics under which we want to revise is closed under intersection, we know that it is, in principal, possible to obtain rational operators. The definition of the operator $*$ as $F * G = \rho^{\mathrm{AF}}_{\sigma}(\min(\sigma(G)), \preceq_F))$, however, already suggests that, without further restriction of the possible orderings $\preceq_F$, it might be necessary to realize any subset of $\sigma(G)$. The following example elaborates on this issue.

Example 43. Consider revision by operator $*_{nai}$ of the AF F depicted on the left of Figure 4.1 under the naive semantics. Assume the ranking $\preceq_F$ for F given in Figure 4.2, where sets of arguments in the same level are ranked equally, i.e., for instance, $\{a, b, c\} \approx_F \{a', b, c\} \approx_F \{a, b', c\} \approx_F \{a, b, c'\}$, and an arrow denotes a strict relation, e.g. $\{a', b', c'\} \prec_F \{a', b', c\}$. Note that $\preceq_F$ is a faithful ranking for F, since $nai(F) = \{\{a, b, c\}, \{a', b, c\}, \{a, b', c\}, \{a, b, c'\}\}$ are equally ranked and all other sets of arguments are ranked strictly lower. Let the AF G depicted in the center of Figure 4.1 be the revising AF and observe that $nai(G) = \{\{a', b', c'\}, \{a', b', c\}, \{a', b, c'\}, \{a, b', c'\}\}$. We get $F *_{nai} G = \rho^{\mathrm{AF}}_{nai}(\min(nai(G), \preceq_F)) = \rho^{\mathrm{AF}}_{nai}(\{\{a', b', c\}, \{a', b, c'\}, \{a, b', c'\}\})$. The problem that we are facing is that the set $\{\{a', b', c\}, \{a', b, c'\}, \{a, b', c'\}\}$ is not realizable under

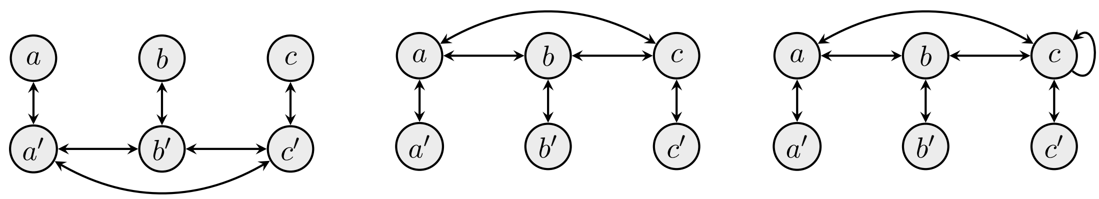


Figure 4.1: AFs F, G, and H, from left to right, having $nai(F) = \{\{a,b,c\}, \{a',b,c\}, \{a,b',c\}, \{a,b,c'\}\}$, $nai(G) = \{\{a',b',c'\}, \{a',b',c\}, \{a',b,c'\}, \{a,b',c'\}\}$, and $nai(H) = \{\{a,b',c'\}, \{a',b,c'\}, \{a',b',c'\}\}$.

$$\{a,b,c\}, \{a',b,c\}, \{a,b',c\}, \{a,b,c'\}$$

$$\uparrow$$

$$\{a,b',c'\}, \{a',b,c'\}, \{a',b',c\}$$

$$\uparrow$$

$$\{a',b',c'\}$$

$$\uparrow$$

rest

Figure 4.2: Faithful ranking for AF F in Figure 4.1, used in Example 43.

nai, as discussed in Example 15. We get $\rho^{\mathrm{AF}}_{nai}(\{\{a',b',c\}, \{a',b,c'\}, \{a,b',c'\}\}) = (\emptyset, \emptyset)$ and hence $nai(F *_{nai} G) = \{\emptyset\}$, violating the postulate (A1$_{nai}$).

Of course, the definition of ρ^{AF}_{nai} to deliver the empty AF for non-realizable extension-sets is a somehow arbitrary choice. One could also try to define the realizing function in way that the AF realizes a set of extensions which is similar to the desired set. So let $\rho'^{\mathrm{AF}}_{nai} : 2^{2^{\mathfrak{A}}} \mapsto AF_{\mathfrak{A}}$ be an alternative realizing function which behaves as ρ^{AF}_{nai} if the given set is realizable and, given some $\mathbb{S} \notin \Sigma^{nai}_{\mathrm{AF}}$, it is defined such that $nai(\rho'^{\mathrm{AF}}_{nai}(\mathbb{S})) = \mathbb{S}'$, with $\mathbb{S}'$ being a $\subseteq$-maximal set $\mathbb{S}' \subset \mathbb{S}$. As $\mathbb{S}'$ is not unique, we assume $\rho'^{\mathrm{AF}}_{nai}$ to make an arbitrary choice among the candidates.

Now consider again the revision of F by G, this time with the alternative realizing function $\rho'^{\mathrm{AF}}_{nai}$. We may get $F *_{nai} G = \rho'^{\mathrm{AF}}_{nai}(\{\{a',b',c\}, \{a',b,c'\}, \{a,b',c'\}\})$ with

$$nai(F *_{nai} G) = \{\{a',b,c'\}, \{a',b',c\}\}.$$ [2]

Let H be the AF depicted on the right of Figure 4.1 and observe that $nai(H) =$

[2]Other possible choices given the definition of $\rho'^{\mathrm{AF}}_{nai}$ are $nai(F *_{nai} G) = \{\{a,b',c'\}, \{a',b',c\}\}$ and $nai(F *_{nai} G) = \{\{a',b',c\}, \{a',b,c'\}\}$. In any case, we end up violating (A6$_{nai}$) for specific AFs H.

$\{\{a, b', c'\}, \{a', b, c'\}, \{a', b', c'\}\}$. We obtain the following:

$$nai(F *_{nai} G) \cap nai(H) = \{\{a', b, c'\}\}$$
$$nai(F *_{nai} \rho'^{\text{AF}}_{nai}(nai(G) \cap nai(H))) = \{\{a, b', c'\}, \{a', b, c'\}\}$$

This means that (A6$_{nai}$) is violated.

Similarly, we run into problems when we try to define the realizing function such that it realizes some superset of the desired non-realizable set. Intuitively, in both cases it leads to more change than prescribed by the ranking and is therefore conflicting with (A5$_\sigma$) and (A6$_\sigma$), the postulates ensuring minimal change. $\Diamond$

In order to obtain a representation result between rationality postulates and rankings, one has to put further restrictions on the rankings. For revision within the Horn fragment [84], where a similar problem arises, Delgrande and Peppas introduce the notion of *compliant* rankings which restrict faithful rankings appropriately. We will introduce a refined version of compliance for revision of ADFs in Section 4.3.

In this section we focus on a particular class of semantics, which we call proper I-maximal. The name is inspired by the notion of an I-maximal semantics [11], which is a semantics always yielding an incomparable set of extensions.

Definition 82. A semantics σ is called *proper I-maximal* if for each $\mathbb{S} \in \Sigma^\sigma_{\text{AF}}$ it holds that

1. $\mathbb{S}$ is incomparable,
2. $\mathbb{S}' \in \Sigma^\sigma_{\text{AF}}$ for any $\mathbb{S}' \subseteq \mathbb{S}$ with $\mathbb{S}' \neq \emptyset$, and
3. for any incomparable $S_1, S_2 \in 2^{\mathfrak{A}}$ it holds that $\{S_1, S_2\} \in \Sigma^\sigma_{\text{AF}}$.

In words, an I-maximal [11] semantics σ (i.e. a semantics where members of $\Sigma^\sigma_{\text{AF}}$ are always incomparable) is proper if, on the one hand, it holds that for any AF F we can realize under σ any non-empty subset of $\sigma(F)$, and, on the other hand, any pair of incomparable sets of arguments, is realizable under σ. While this might seem like a rather strong requirement in the first place, it turns out that most of the major semantics are indeed proper I-maximal. This is shown by the next result, which follows from the characterizations of signatures in Section 3.2.

Proposition 33. *Preferred, stable, semi-stable and stage semantics are proper I-maximal.*

Proof. Let $\sigma \in \{prf, stb, sem, stg\}$. We can establish the fact that properties (1) to (3) from Definition 82 hold for each $\mathbb{S} \in \Sigma^\sigma_{\text{AF}}$ by results from Section 3.2: (1) follows from Theorem 1, (2) was shown in Theorem 5, and (3) is by Proposition 11. $\square$

On the ranking side we can now define a less demanding version of faithful assignments, which is adjusted to the nature of (proper) I-maximal semantics.

Definition 83. A preorder $\preceq$ on $2^{\mathfrak{A}}$ is *I-total* if $E_1 \preceq E_2$ or $E_2 \preceq E_1$ for any pair E_1, E_2 of incomparable sets of arguments.

Given a semantics σ and an AF F, an *I-faithful ranking* for F is an I-total preorder $\preceq_F$ on $2^{\mathfrak{A}}$ such that, for any $E_1, E_2 \in 2^{\mathfrak{A}}$, it holds that

(*i*) if $E_1, E_2 \in \sigma(F)$ then $E_1 \approx_F E_2$, and

(*ii*) if $E_1 \in \sigma(F)$ and $E_2 \notin \sigma(F)$ then $E_1 \prec_F E_2$.

An *I-faithful assignment* maps every AF F to an I-faithful ranking $\preceq_F$ for F such that

(*iii*) $\preceq_{F_1} = \preceq_{F_2}$ for any ADFs F_1, F_2 with $\sigma_3(F_1) = \sigma_3(F_2)$.

I-faithful assignments differ from faithful assignments in that they require the rankings to be only I-total, thus allowing, but not requiring, them to be partial with respect to $\subseteq$-comparable pairs of extensions. Our use of I-faithful assignments is motivated by how proper I-maximal semantics work. Given an operator $*$ for revision under semantics σ and $F \in AF_{\mathfrak{A}}$, the natural way to rank two extensions E_1 and E_2 is by inspection of $F * \rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\})$: if $E_1 \in \sigma(F * \rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\}))$, then E_1 is considered “at least as plausible” as E_2 and it should hold that $E_1 \preceq_F E_2$. However, by proper I-maximality of σ, $\rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\})$ is only guaranteed to exist if E_1 and E_2 are incomparable. Thus, if E_1 and E_2 are $\subseteq$-comparable, $*$ might not have any means to decide between E_1 and E_2, hence it is natural to allow them to be incomparable with respect to $\preceq_F$.

Rational operators induced by pseudo-preorders. The next example shows that in the presence of limited expressiveness we can get rational operators from rankings which are usually considered unintuitive.

Example 44. Let σ be a proper I-maximal semantics and suppose that for an AF F we have a ranking $\preceq_F$ on $2^{\mathfrak{A}}$ which behaves as in Figure 4.3 for the extensions $\{a\}, \{b, c\}, \{a, c\}$ and $\{b\}$, and as a faithful ranking otherwise. Again, an arrow means that the relation is strict: for example, $\{a\} \preceq_F \{b, c\}$ and $\{b, c\} \not\preceq_F \{a\}$, abbreviated as $\{a\} \prec_F \{b, c\}$. The relation $\preceq_F$, then, contains a non-transitive cycle and is therefore not a preorder. However, an inspection of the ranking reveals that for any non-empty σ-realizable set $\mathbb{S}$, $\min(\mathbb{S}, \preceq_F)$ is still well-defined and non-empty: recall that we are assuming σ to be proper I-maximal. Therefore elements of $\mathbb{S}$ are pairwise incomparable. Hence there is no $\mathbb{S} \subseteq \Sigma_{\mathrm{AF}}^\sigma$ such that $\mathbb{S} \supseteq \{\{a\}, \{a, c\}\}$ or $\mathbb{S} \supseteq \{\{b\}, \{b, c\}\}$. On the other hand, for instance, if $\mathbb{S} = \{\{a\}, \{b, c\}\}$, then $\min(\mathbb{S}, \preceq_F) = \{\{a\}\}$. Thus we can define an operator $*_\sigma$ in the familiar way, by taking $F *_\sigma G = \rho_\sigma^{\mathrm{AF}}(\min(\sigma(G), \preceq_F))$, and it is then straightforward to verify that this operator $*_\sigma$ is well-defined and satisfies postulates $(\mathrm{A1}_\sigma)$–$(\mathrm{A6}_\sigma)$. $\Diamond$

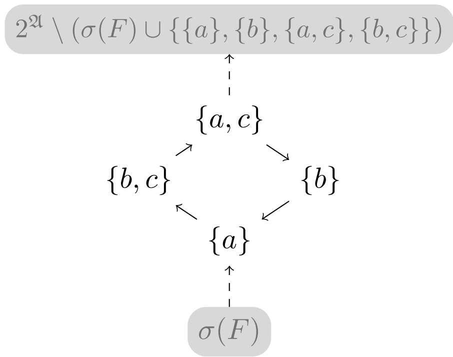


Figure 4.3: Cycles in rankings on extensions.

However, we want to avoid non-transitive cycles: since a natural reading of the rankings on $2^{\mathfrak{A}}$ is that they are plausibility relations, one would expect them to be transitive. The correspondence between postulates and preorders could still be established if there was another ranking $\preceq'_F$, which is indeed a preorder, that induces the operator discussed in Example 44. The following example shows that this is also not the case.

Example 45. Assume there is a ranking $\preceq'_F$ which is transitive and yields the same revision operator as the pseudo-preorder in Figure 4.3. To do so, it has to satisfy $\min(\{\{a\},\{b,c\}\},\preceq'_F) = \{\{a\}\}$, because we know that $\sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{\{a\},\{b,c\}\})) = \{\{a\}\}$. Thus it holds that $\{a\} \prec'_F \{b,c\}$. Similarly, we get that $\{b,c\} \prec'_F \{a,c\} \prec'_F \{b\} \prec'_F \{a\}$, and the cycle is reiterated. $\diamondsuit$

As it is undesirable to have revision operators that characterize non-transitive rankings, we prevent this situation by making use of the additional postulate (Acyc_σ). This postulate is adapted from [84] and is motivated by the fact that, without it, postulates ($\mathrm{A1}_\sigma$)–($\mathrm{A6}_\sigma$) can characterize revision operators generated with unsuitable rankings (see Examples 44 and 45).[3]

(Acyc_σ) If for $0 \leq i < n$, $\sigma(F * G_{i+1}) \cap \sigma(G_i) \neq \emptyset$ and $\sigma(F * G_0) \cap \sigma(G_n) \neq \emptyset$ then $\sigma(F * G_n) \cap \sigma(G_0) \neq \emptyset$.

Intuitively, (Acyc_σ) prevents cycles as the one in Figure 4.3. The following example goes into more detail.

[3] Note that the (Acyc)-postulate is redundant for revision in formalisms with full expressiveness such as propositional logic (see Proposition 3 of [84]).

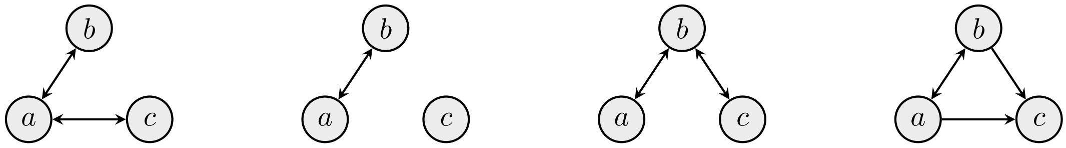


Figure 4.4: AFs $G_0, \ldots, G_3$, from left to right, having $\sigma(G_0) = \{\{a\}, \{b, c\}\}$, $\sigma(G_1) = \{\{b, c\}, \{a, c\}\}$, $\sigma(G_2) = \{\{a, c\}, \{b\}\}$, and $\sigma(G_3) = \{\{a\}, \{b\}\}$ for $\sigma \in \{prf, stb, sem, stg\}$.

Example 46. Let σ be a proper I-maximal semantics and consider the AFs $G_0, \ldots, G_3$ with

$$\begin{aligned}
\sigma(G_0) &= \{\{a\}, \{b, c\}\},\\
\sigma(G_1) &= \{\{b, c\}, \{a, c\}\},\\
\sigma(G_2) &= \{\{a, c\}, \{b\}\}, \text{ and}\\
\sigma(G_3) &= \{\{a\}, \{b\}\}.
\end{aligned}$$

For $\sigma \in \{prf, stb, sem, stg\}$ such AFs are depicted in Figure 4.4. Moreover let F be an AF with $\sigma(F) \cap \{\{a\}, \{b\}, \{a, c\}, \{b, c\}\} = \emptyset$. Assume a rational operator $*$ such that the following holds:

$$\begin{aligned}
\sigma(F * G_1) \cap \sigma(G_0) &= \{\{a\}\},\\
\sigma(F * G_2) \cap \sigma(G_1) &= \{\{b, c\}\},\\
\sigma(F * G_3) \cap \sigma(G_2) &= \{\{a, c\}\}, \text{ and}\\
\sigma(F * G_0) \cap \sigma(G_3) &= \{\{b\}\}.
\end{aligned}$$

First note that this means that a corresponding ranking must have $\{a\} \preceq_F \{b, c\} \preceq_F \{a, c\} \preceq_F \{b\} \preceq_F \{a\}$. Applying (Acyc$_\sigma$) it follows that

$$\sigma(F * G_3) \cap \sigma(G_0) \neq \emptyset.$$

That means, by $*$ satisfying (A1$_\sigma$), that $\{a\} \in \sigma(F * G_3)$. Hence, in the corresponding ranking we also have $\{a\} \preceq_F \{b\}$, i.e. $\{a\} \approx_F \{b\}$. The same can be derived for the other pairs of extensions and we arrive at $\{a\} \approx_F \{b, c\} \approx_F \{a, c\} \approx_F \{b\}$. That way (Acyc$_\sigma$) prevents $*$ from giving rise to a non-transitive cycle in the corresponding ranking. $\Diamond$

Before coming to the main representation result of this section, we state a preliminary lemma, which we will use later. It says that I-total preorders always yield most preferred elements among the extensions under a proper I-maximal semantics.

Lemma 16. *Let σ be a proper I-maximal semantics and $\preceq$ an I-total preorder on $2^{\mathfrak{A}}$. For each $\mathbb{S} \in \Sigma^\sigma_{AF}$ it holds that* $\min(\mathbb{S}, \preceq) \neq \emptyset$.

Proof. First note that the extension-set $\mathbb{S}$ is finite. Moreover, since σ is proper I-maximal, S_i and S_j are incomparable for each $1 \leq i, j \leq n$, hence $S_i \preceq S_j$ or $S_j \preceq S_i$ (or both). Hence it is well-known that $\mathbb{S}$ must have at least one minimal element wrt. $\preceq$. $\square$

With these preliminaries, we can now state our main representation results. The first direction shows that each I-faithful assignment gives rise to a rational revision operator.

Theorem 30. *Let σ be a proper I-maximal semantics. If there exists an I-faithful assignment mapping any $F \in AF_{\mathfrak{A}}$ to an I-faithful ranking $\preceq_F$ for F, then the revision operator $*_\sigma \colon AF_{\mathfrak{A}} \times AF_{\mathfrak{A}} \mapsto AF_{\mathfrak{A}}$ defined as*

$$F *_\sigma G = \rho_\sigma^{AF}(\min(\sigma(G), \preceq_F))$$

satisfies postulates ($A1_\sigma$)–($A6_\sigma$) and ($Acyc_\sigma$).

Proof. Let $F \in AF_{\mathfrak{A}}$ be an AF and $\preceq_F$ be the I-faithful ranking for F obtained from the I-faithful assignment. Since σ is proper I-maximal, any non-empty subset of $\sigma(G)$ (and, in particular, $\min(\sigma(G), \preceq_F)$, which is non-empty by Lemma 16) is realizable under σ. Thus, by definition of $\rho_\sigma^{\mathrm{AF}}$ (cf. Definition 38), it holds that $\sigma(\rho_\sigma^{\mathrm{AF}}(\min(\sigma(G), \preceq_F))) = \min(\sigma(G), \preceq_F)$. Hence, for any AF G, it holds that $\sigma(F *_\sigma G) = \min(\sigma(G), \preceq_F)$, which we use without further comment in the remainder of the proof.

It is now straightforward that (A1_σ) is satisfied.

As $\preceq_F$ is I-faithful, elements of $\sigma(F)$ are the minimal elements of $\preceq_F$, hence, if $\sigma(F) \cap \sigma(G) \neq \emptyset$, then $\min(\sigma(F), \preceq_F) = \sigma(F) \cap \sigma(G)$, i.e. (A$2_\sigma$) is satisfied.

Postulate (A3_σ) follows from Lemma 16.

As $\preceq_F$ has been obtained from a faithful assignment, it holds, due to condition (*iii*) of Definition 83, that for each AF F_2 with $\sigma(F) = \sigma(F_2)$, also $\preceq_F = \preceq_{F_2}$. Therefore if $\sigma(G_1) = \sigma(G_2)$ for arbitrary AFs $G_1, G_2 \in AF_{\mathfrak{A}}$, then $\min(\sigma(G_1), \preceq_F) = \min(\sigma(G_2), \preceq_{F_2})$. Hence postulate (A$4_\sigma$) is satisfied.

Postulates (A5_σ) and (A6_σ) are trivially satisfied if $\sigma(F *_\sigma G) \cap \sigma(H) = \emptyset$. Assume $\sigma(F *_\sigma G) \cap \sigma(H) \neq \emptyset$. That means, by $\sigma(F *_\sigma G) \subseteq \sigma(G)$ (cf. (A1_σ)), that also $\sigma(G) \cap \sigma(H) \neq \emptyset$. Hence $\sigma(\rho_\sigma^{\mathrm{AF}}(\sigma(G) \cap \sigma(H))) = \sigma(G) \cap \sigma(H)$. Now assume further, towards a contradiction, that there is some $E \in \min(\sigma(G), \preceq_F) \cap \sigma(H)$ with $E \notin \min(\sigma(\rho_\sigma^{\mathrm{AF}}(\sigma(G) \cap \sigma(H))), \preceq_F) = \min(\sigma(G) \cap \sigma(H), \preceq_F)$. Since $E \in \sigma(G) \cap \sigma(H)$ there must then be some $E' \in \sigma(G) \cap \sigma(H)$ with $E' \prec_F E$, a contradiction to $E \in \min(\sigma(G), \preceq_F)$. Therefore $\sigma(F *_\sigma G) \cap \sigma(H) \subseteq \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\sigma(G) \cap \sigma(H)))$, i.e. (A$5_\sigma$) is satisfied. To show that $\sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\sigma(G) \cap \sigma(H))) \subseteq \sigma(F *_\sigma G) \cap \sigma(H)$ also holds, assume $E \in \min(\sigma(G) \cap \sigma(H), \preceq_F)$ and $E \notin \min(\sigma(G), \preceq_F) \cap \sigma(H)$. Since $E \in \sigma(H)$, it follows that $E \notin \min(\sigma(G), \preceq_F)$. Let $E' \in \min(\sigma(G), \preceq_F) \cap \sigma(H)$ (assumed to be non-empty). Then $E' \in \sigma(G) \cap \sigma(H)$ holds. Moreover, since σ is an I-maximal semantics, E and E' are incomparable. Now as $E \in \min(\sigma(G) \cap \sigma(H), \preceq_F)$ and the fact that $\preceq_F$ is I-total, it must hold that $E \preceq_F E'$. Hence from $E' \in \min(\sigma(G), \preceq_F)$ it follows that $E \in \min(\sigma(G), \preceq_F)$, a contradiction. Hence also (A6_σ) is satisfied.

It remains to be shown that (Acyc$_\sigma$) also holds. Let $G_0, G_1, \ldots, G_n$ be a sequence of AFs such that for all $0 \leq i < n$, it holds that $\sigma(F *_\sigma G_{i+1}) \cap \sigma(G_i) \neq \emptyset$ and $\sigma(F *_\sigma G_0) \cap \sigma(G_n) \neq$

$\emptyset$. From $\sigma(F *_\sigma G_1) \cap \sigma(G_0) \neq \emptyset$ we derive that $\min(\sigma(G_1), \preceq_F) \cap \sigma(G_0) \neq \emptyset$. Hence there is an extension $E'_0 \in \sigma(G_0)$ such that $E'_0 \preceq_F E_1$ for all $E_1 \in \sigma(G_1)$. Likewise we get, for any $0 \leq i < n$, from $\sigma(F *_\sigma G_{i+1}) \cap \sigma(G_i) \neq \emptyset$ that there is an extension $E'_i \in \sigma(G_i)$ such that $E'_i \preceq_F E_{i+1}$ for all $E_{i+1} \in \sigma(G_{i+1})$. In particular, there is an extension $E'_{n-1} \in \sigma(G_{n-1})$ such that $E'_{n-1} \preceq_F E_n$ for all $E_n \in \sigma(G_n)$. From transitivity of $\preceq_F$ we get $E'_0 \preceq_F E_n$ for all $E_n \in \sigma(G_n)$. Finally, from $\sigma(F *_\sigma G_0) \cap \sigma(G_n) \neq \emptyset$ it follows that there is some $E'_n \in \sigma(G_n)$ with $E'_n \in \sigma(G_0)$ and $E'_n \preceq_F E_0$ for all $E_0 \in \sigma(G_0)$ (in particular for E'_0). Now from $E'_n \preceq_F E'_0 \preceq_F E_n$ (for all $E_n \in \sigma(G_n)$) it follows that $E'_n \in \min(\sigma(G_n), \preceq_F)$. Hence $\sigma(F *_\sigma G_n) \cap \sigma(G_0) \neq \emptyset$. □

The other direction shows that also every rational revision operator has a corresponding I-faithful assignment.

Theorem 31. *Let σ be a proper I-maximal semantics. If $*_\sigma \colon AF_{\mathfrak{A}} \times AF_{\mathfrak{A}} \mapsto AF_{\mathfrak{A}}$ is an operator satisfying postulates (A1$_\sigma$)–(A6$_\sigma$) and (Acyc$_\sigma$), then there exists an I-faithful assignment mapping every $F \in AF_{\mathfrak{A}}$ to an I-faithful ranking $\preceq_F$ for F such that* $\min(\sigma(G), \preceq_F) = \sigma(F *_\sigma G)$*, for any $G \in AF_{\mathfrak{A}}$.*

Proof. Assume there is an operator $*_\sigma \colon AF_{\mathfrak{A}} \times AF_{\mathfrak{A}} \mapsto AF_{\mathfrak{A}}$ satisfying postulates (A1$_\sigma$)–(A6$_\sigma$) and (Acyc$_\sigma$), and take an arbitrary $F \in AF_{\mathfrak{A}}$. We construct $\preceq_F$ in two steps and then show that $\preceq_F$ is indeed an I-faithful ranking for F such that $\min(\sigma(G), \preceq_F) = \sigma(F *_\sigma G)$. We will, as part of the proof, give some intermediate lemmas in the interest of readability.

In the first step we define the relation $\preceq'_F$ on $2^{\mathfrak{A}}$ such that $E \preceq'_F E$ for any $E \in 2^{\mathfrak{A}}$ and for any two incomparable $E, E' \in 2^{\mathfrak{A}}$,

$$E \preceq'_F E' \text{ if and only if } E \in \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E, E'\})).$$

Note that $\preceq'_F$ is obviously reflexive. In the next step we build the transitive closure of $\preceq'_F$ to obtain the relation $\preceq_F$. In other words, we define

$$E \preceq_F E' \text{ if and only if there exist } E_1, \dots, E_n \in 2^{\mathfrak{A}} \text{ such that: } E_1 = E, E_n = E' \text{ and } E_1 \preceq'_F \cdots \preceq'_F E_n.$$

In the remainder of the proof we show that $\preceq_F$ is an I-faithful ranking for F such that $\min(\sigma(G), \preceq_F) = \sigma(F * G)$. First, notice that if $E_1 \preceq'_F E_2$ then $E_1 \preceq_F E_2$. Hence $\preceq_F$ is reflexive and, by construction, it is transitive, which makes it a preorder on $2^{\mathfrak{A}}$. Additionally, for any two incomparable sets of arguments $E_1, E_2 \in 2^{\mathfrak{A}}$, proper I-maximality of σ guarantees that $\{E_1, E_2\} \in \Sigma^\sigma_{AF}$ and therefore $\sigma(\rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\})) = \{E_1, E_2\}$. By (A1$_\sigma$) and (A3$_\sigma$), $\sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\}))$ is then a non-empty subset of $\{E_1, E_2\}$, thus $E_1 \preceq'_F E_2$ or $E_2 \preceq'_F E_1$. It follows that also $E_1 \preceq_F E_2$ or $E_2 \preceq_F E_1$, hence $\preceq_F$ is I-total.

Next we argue that $\preceq_F$ is an I-faithful ranking for F. Due to the fact that $\{E_1, E_2\}$ is realizable whenever E_1 and E_2 are incomparable by proper I-maximality of σ, we will make use of $\{E_1, E_2\} = \sigma(\rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\}))$ without further comment in the remainder of the proof.

Lemma 17. *If $E_1, E_2 \in \sigma(F)$, then $E_1 \approx_F E_2$.*

Proof. From (A2$_\sigma$) and proper I-maximality of σ, we get $\sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\})) = \sigma(F) \cap \{E_1, E_2\} = \{E_1, E_2\}$. Thus $E_1 \preceq'_F E_2$ and $E_2 \preceq'_F E_1$, which implies $E_1 \preceq_F E_2$ and $E_2 \preceq_F E_1$, i.e. $E_1 \approx_F E_2$. □

Lemma 17 shows that $\preceq_F$ satisfies property (*i*) of I-faithful rankings. For property (*ii*) we make use of the following lemmas. It is in this context that (Acyc$_\sigma$) proves crucial.

Lemma 18. *If $E_1, \dots, E_n \in 2^{\mathfrak{A}}$ are pairwise distinct extensions with $E_1 \preceq'_F E_2 \preceq'_F \cdots \preceq'_F E_n \preceq'_F E_1$, then $E_1 \preceq'_F E_n$.*

Proof. If $n = 2$ the conclusion follows immediately. In the following we assume that $n > 2$. From the hypothesis we have, by the definition of $\preceq'_F$, that

$$\begin{aligned} E_i &\in \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_i, E_{i+1}\})), \text{ for } 1 \leq i < n, \text{ and} \\ E_n &\in \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_n, E_1\})). \end{aligned}$$

This means that

$$\begin{aligned} E_1 &\in \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\})) \cap \{E_n, E_1\}, \\ E_i &\in \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_i, E_{i+1}\})) \cap \{E_{i-1}, E_i\}, \text{ for } 1 < i < n, \text{ and,} \\ E_n &\in \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_n, E_1\})) \cap \{E_{n-1}, E_n\}. \end{aligned}$$

Since $*_\sigma$ satisfies (Acyc$_\sigma$), it follows that

$$\sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_n, E_1\})) \cap \{E_1, E_2\} \neq \emptyset.$$

From $*_\sigma$ satisfying (A5$_\sigma$) and (A6$_\sigma$) it follows that

$$\sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_n, E_1\})) \cap \{E_1, E_2\} = \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_n, E_1\} \cap \{E_1, E_2\})).$$

Since $\{E_n, E_1\} \cap \{E_1, E_2\} = \{E_1\}$ we get by (A4$_\sigma$) that

$$\sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_n, E_1\} \cap \{E_1, E_2\})) = \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_1\})).$$

Finally, using (A1$_\sigma$) and (A3$_\sigma$) we conclude that $\sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_1\})) = \{E_1\}$, and thus

$$E_1 \in \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_n, E_1\})),$$

which implies, by definition of $\preceq'_F$, that $E_1 \preceq'_F E_n$. □

Lemma 19. *For any $E, E' \in 2^{\mathfrak{A}}$, it holds that if $E \prec'_F E'$ then $E \prec_F E'$.*

Proof. Assume $E \prec'_F E'$. From the definition of $\preceq_F$ it is clear that $E \preceq_F E'$. It remains to be shown that $E' \not\preceq_F E$. Suppose, towards a contradiction, that $E' \preceq_F E$. Then there exist $E_1, \dots, E_n \in 2^{\mathfrak{A}}$ such that $E_1 = E'$, $E_n = E$ and $E_1 \preceq'_F \cdots \preceq'_F E_n$. Since we also have $E \preceq'_F E'$ (i.e. $E_n \preceq'_F E_1$) by assumption, we can apply Lemma 18 to get $E_1 \preceq'_F E_n$, that is $E' \preceq'_F E$, a contradiction to $E \prec_F E'$. □

Lemma 20. *Given $E_1, E_2 \in 2^{\mathfrak{A}}$, if $E_1 \in \sigma(F)$, $E_2 \notin \sigma(F)$, and E_1 and E_2 are incomparable, then $E_1 \prec_F E_2$.*

Proof. By proper I-maximality of σ and (A2$_\sigma$) we get

$$\sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\})) = \sigma(F) \cap \{E_1, E_2\} = \{E_1\}.$$

This implies, by definition of $\preceq'_F$, that $E_1 \prec'_F E_2$. By Lemma 19 we get $E_1 \prec_F E_2$. □

Lemma 20 gives us property (*ii*) of I-faithful rankings. For property (*iii*) of faithful assignments assume an AF $F' \in AF_{\mathfrak{A}}$ with $\sigma(F) = \sigma(F')$. (A4$_\sigma$) ensures that $\preceq'_F = \preceq'_{F'}$ and therefore it also holds that $\preceq_F = \preceq_{F'}$.

It remains to show that the σ-extensions of $F *_\sigma G$, for any $G \in AF_{\mathfrak{A}}$, are the minimal elements of $\sigma(G)$ with respect to $\preceq_F$.

Lemma 21. *For any $E_1, E_2 \in 2^{\mathfrak{A}}$ and any $G \in AF_{\mathfrak{A}}$, it holds that if $E_1 \in \sigma(G)$, $E_2 \in \sigma(F *_\sigma G)$ and $E_1 \preceq'_F E_2$, then $E_1 \in \sigma(F *_\sigma G)$.*

Proof. First observe that, by $E_2 \in \sigma(F *_\sigma G)$ and (A1$_\sigma$), also $E_2 \in G$. Hence, by proper I-maximality of σ, E_1 and E_2 are incomparable, hence $\sigma(\rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\})) = \{E_1, E_2\}$, an observation we will use throughout the proof.

From the assumption that $E_2 \in \sigma(F *_\sigma G)$, we have $\sigma(F *_\sigma G) \cap \{E_1, E_2\} \neq \emptyset$. By (A5$_\sigma$) and (A6$_\sigma$) we get

$$\sigma(F *_\sigma G) \cap \{E_1, E_2\} = \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\sigma(G) \cap \{E_1, E_2\})).$$

Now recall that $\{E_1, E_2\} \subseteq \sigma(G)$. Thus $\sigma(G) \cap \{E_1, E_2\} = \{E_1, E_2\}$. From this and (A4$_\sigma$) it follows that

$$\sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\sigma(G) \cap \{E_1, E_2\})) = \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\})).$$

Putting these equalities together with the fact that $E_1 \in \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\}))$ (since $E_1 \preceq'_F E_2$ by assumption), we get that $E_1 \in \sigma(F *_\sigma G)$. □

Lemma 22. *For any $G \in AF_{\mathfrak{A}}$, it holds that* $\min(\sigma(G), \preceq'_F) = \sigma(F *_\sigma G)$.

Proof. $\subseteq$: Let $E_1 \in \min(\sigma(G), \preceq'_F)$. This implies that $\sigma(G) \neq \emptyset$ and, by (A3$_\sigma$), $\sigma(F *_\sigma G) \neq \emptyset$. Thus we can take an arbitrary $E_2 \in \sigma(F *_\sigma G)$ for which, by (A1$_\sigma$), also $E_2 \in \sigma(G)$. Now by $E_1, E_2 \in \sigma(G)$ we follow by proper I-maximality of σ that E_1 and E_2 are incomparable. As $E_1 \in \min(\sigma(G), \preceq'_F)$ and $\preceq'_F$ was already shown to be I-total, it follows that $E_1 \preceq'_F E_2$. Thus, by Lemma 21, we get $E_1 \in \sigma(F *_\sigma G)$.

$\supseteq$: Let $E_1 \in \sigma(F *_\sigma G)$ and assume $E_1 \notin \min(\sigma(G), \preceq'_F)$, i.e. there is some $E_2 \in \sigma(G)$ such that $E_2 \prec'_F E_1$. From proper I-maximality of σ it follows that E_1 and E_2 are incomparable, hence $\sigma(\rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\})) = \{E_1, E_2\}$. We have that $\sigma(F *_\sigma G) \cap \{E_1, E_2\} \neq \emptyset$ and thus, by (A5$_\sigma$) and (A6$_\sigma$):

$$\sigma(F *_\sigma G) \cap \{E_1, E_2\} = \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\sigma(G) \cap \{E_1, E_2\})).$$

Recalling that $\sigma(G) \cap \{E_1, E_2\} = \{E_1, E_2\}$ it follows from (A4$_\sigma$) that:

$$\sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\sigma(G) \cap \{E_1, E_2\})) = \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\})).$$

Putting these equations together with the fact that $E_1 \in \sigma(F *_\sigma G) \cap \{E_1, E_2\}$, we get that $E_1 \in \sigma(F *_\sigma \rho_\sigma^{\mathrm{AF}}(\{E_1, E_2\}))$ and thus $E_1 \preceq'_F E_2$, which is a contradiction to $E_2 \prec'_F E_1$. □

Lemma 23. *For any $G \in AF_{\mathfrak{A}}$,* $\min(\sigma(G), \preceq_F) = \min(\sigma(G), \preceq'_F)$.

Proof. $\subseteq$: Let $E_1 \in \min(\sigma(G), \preceq_F)$ and assume, towards a contradiction, that $E_1 \notin \min(\sigma(G), \preceq'_F)$, i.e. there exists some $E_2 \in \sigma(G)$ with $E_2 \prec'_F E_1$. By Lemma 19, this implies that $E_2 \prec_F E_1$, a contradiction to $E_1 \in \min(\sigma(G), \preceq_F)$.

$\supseteq$: Let $E_1 \in \min(\sigma(G), \preceq'_F)$ and take any $E_2 \in \sigma(G)$. If $E_2 = E_1$, it follows that $E_1 \preceq'_F E_2$. If $E_2 \neq E_1$, then by proper I-maximality of σ, E_1 and E_2 are incomparable and thus $E_1 \preceq'_F E_2$ or $E_2 \preceq'_F E_1$. We cannot have that $E_2 \prec'_F E_1$, since this would contradict the hypothesis that $E_1 \in \min(\sigma(G), \preceq'_F)$, therefore $E_1 \preceq'_F E_2$ must hold. In both cases it follows that $E_1 \preceq_F E_2$, hence $E_1 \in \min(\sigma(G), \preceq_F)$. □

Lemmas 22 and 23 imply that for any $G \in AF_{\mathfrak{A}}$, it holds that $\sigma(F *_\sigma G) = \min(\sigma(F), \preceq_F)$. This concludes the proof of the theorem. □

We have established, for revision under proper I-maximal semantics, a correspondence between operators fulfilling postulates (A1$_\sigma$)–(A6$_\sigma$) and (Acyc$_\sigma$) on the one hand side, and operators obtained from I-faithful assignments on the other. Regarding concrete operators, notice that every I-faithful assignment is also a faithful assignment. Hence, any faithful assignment for AFs can be used, via Theorem 30, to represent a rational revision operator $*_\sigma : AF_{\mathfrak{A}} \times AF_{\mathfrak{A}} \to AF_{\mathfrak{A}}$. Thus, any model-based revision operator from the standard literature on belief change, in particular Dalal's operator [80], can be used as a revision operator of AFs by AFs. The following example illustrates the revision of an AF by Dalal's operator.

Figure 4.5: AF F (left) being revised by the AF G (right) in Example 47.

Example 47. Consider the AFs F and G in Figure 4.5 an let $\sigma \in \{stb, prf, sem, stg\}$. Observe that $\sigma(F) = \{\{a,c\},\{b,c\}\}$ and therefore the preorder $\preceq_F^\sigma$ obtained from Hamming distance is as follows:

$$\{a,c\} \approx_F^\sigma \{b,c\} \prec_F^\sigma \{a\} \approx_F^\sigma \{b\} \approx_F^\sigma \{c\} \approx_F^\sigma \{a,b,c\} \prec_F^\sigma \{a,b\} \approx_F^\sigma \emptyset$$

Note that $\preceq_F^\sigma$ is a faithful ranking for F. Revising F by G using Dalal's operator then gives us $F *_\sigma^D G = \rho_\sigma^{\mathrm{AF}}(\min(\sigma(G), \preceq_F^\sigma))$. Observing that $\sigma(G) = \{\{a,b\},\{c\}\}$ it we get $F *_\sigma^D G = \rho_\sigma^{\mathrm{AF}}(\{\{c\}\})$. $\Diamond$

4.3 Revision of ADFs

In this section we apply the AGM approach to the revision of ADFs by studying operators $* : ADF_{\mathfrak{A}} \times ADF_{\mathfrak{A}} \mapsto ADF_{\mathfrak{A}}$. As usual for ADFs, we will use three-valued semantics σ_3 for the evaluation of ADFs (cf. Definition 31). The rationality postulates (A1$_{\sigma_3}$)–(A6$_{\sigma_3}$) for revision under σ_3 carry over from Section 4.2, but using ADF realizing functions when needed.[4]

(A1$_{\sigma_3}$) $\sigma_3(F * G) \subseteq \sigma_3(G)$.

(A2$_{\sigma_3}$) If $\sigma_3(F) \cap \sigma_3(G) \neq \emptyset$, then $\sigma_3(F * G) = \sigma_3(F) \cap \sigma_3(G)$.

(A3$_{\sigma_3}$) If $\sigma_3(G) \neq \emptyset$, then $\sigma_3(F * G) \neq \emptyset$.

(A4$_{\sigma_3}$) If $\sigma_3(F_1) = \sigma_3(F_2)$ and $\sigma_3(G) = \sigma_3(H)$, then $\sigma_3(F_1 * G) = \sigma_3(F_2 * H)$.

(A5$_{\sigma_3}$) $\sigma_3(F * G) \cap \sigma_3(H) \subseteq \sigma_3(F * \rho_{\sigma_3}^{\mathrm{ADF}}(\sigma_3(G) \cap \sigma_3(H)))$.

(A6$_{\sigma_3}$) If $\sigma_3(F*G) \cap \sigma_3(H) \neq \emptyset$, then $\sigma_3(F * \rho_{\sigma_3}^{\mathrm{ADF}}(\sigma_3(G) \cap \sigma_3(H))) \subseteq \sigma_3(F*G) \cap \sigma_3(H)$.

In order for the operator to satisfy (A2$_{\sigma_3}$), i.e. to return the consensus of original and revising ADF whenever this is not empty, the signature of σ_3 in ADFs must be closed under intersection. This already rules out rational operators under the complete semantics (cf. Table 3.2).

[4]Note that until now we usually denoted ADFs by the letter D. In this section, however, we will use letters F, G, and H to denote ADFs as the alphabetic successors of D are already used differently.

Theorem 32. *There exists no operator* $* : ADF_{\mathfrak{A}} \times ADF_{\mathfrak{A}} \mapsto ADF_{\mathfrak{A}}$ *that satisfies* $(A2_{com_3})$.

We will focus on preferred and admissible semantics. In Section 4.3.1 we will obtain a representation result for preferred semantics by adjusting the conditions on rankings to the expressiveness of the semantics and again employing a variant of the (Acyc)-postulate. Moreover, we will define a three-valued version of Dalal's operator. Admissible semantics, on the other hand, yield only a single operator satisfying the postulates, as we will see in Section 4.3.2. Since, as we will argue, both approaches have some weaknesses, we propose a hybrid approach which bases rankings on preferred interpretations but allows admissible interpretations of the revising ADF to be the result of the revision in Section 4.3.3.

4.3.1 Revision under Preferred Semantics

In this subsection we will focus on the preferred semantics. To fulfill the postulates, a revision operator will have to result in an ADF having certain preferred interpretations. However, as can be already seen by Theorem 24, preferred semantics underlies certain limits in terms of expressiveness. That is, certain desired outcomes may not be realizable. It will not be necessary to know the exact characterization of $\Sigma_{\mathrm{ADF}}^{prf_3}$, but we will make frequent use of the following sufficient conditions for containment in the signature, i.e. realizability.

Proposition 34. *A set of interpretations* $V \subseteq \mathcal{V}$ *is realizable under* prf_3 *if one of the following holds:*

1. $V \subseteq prf_3(F)$ *and* $V \neq \emptyset$ *for some* $F \in ADF_{\mathfrak{A}}$*;*
2. $V = \{v_1, v_2\}$ *and* v_1 *and* v_2 *are incompatible; or*
3. $V = \{v\}$.

Proof. (1) was shown in Proposition 27 and (2) and (3) are immediate by the characterization of $\Sigma_{\mathrm{ADF}}^{prf_3}$ given in Theorem 24. □

As usual we aim for representing operators satisfying the postulates by means of rankings on the universe of interpretations. Now observe that pairs of interpretations under preferred semantics are always incompatible. Also we have no means to find out, given two compatible interpretations, which of these interpretations the operator gives precedence to. Hence we will make use of the following rankings, which are customized to the expressiveness of preferred semantics.

Definition 84. A preorder $\preceq$ on $\mathcal{V}$ is *i-max-total* if $v_1 \preceq v_2$ or $v_2 \preceq v_1$ for any $v_1, v_2 \in \mathcal{V}$ with $v_1 \not\leq_i v_2$ and $v_2 \not\leq_i v_1$.

Given a semantics σ_3 and an ADF F, an *i-max-faithful ranking* for F is an i-max-total preorder $\preceq_F$ on $\mathcal{V}$ such that, for any $v_1, v_2 \in \mathcal{V}$ with $v_1 \not\leq_i v_2$ and $v_2 \not\leq_i v_1$, it holds that

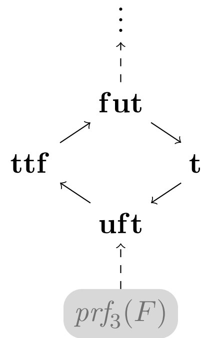


Figure 4.6: Cycles in rankings on interpretation.

(*i*) if $v_1, v_2 \in \sigma_3(F)$ then $v_1 \approx_F v_2$, and

(*ii*) if $v_1 \in \sigma_3(F)$ and $v_2 \notin \sigma_3(F)$ then $v_1 \prec_F v_2$.

An *i-max-faithful assignment* maps every ADF F to an i-max-faithful ranking $\preceq_F$ for F such that

(*iii*) $\preceq_{F_1} = \preceq_{F_2}$ for any ADFs F_1, F_2 with $\sigma_3(F_1) = \sigma_3(F_2)$.

In words, an i-max-faithful ranking has to behave like a faithful ranking, but only has to put incompatible interpretations into relation. It can leave compatible interpretations unrelated, as there is no need for an operator to decide between those interpretations.

The following example shows that the standard set of postulates is again not enough to get a correspondence to preorders on interpretations.

Example 48. Let $A = \{a, b, c\}$ and consider an arbitrary ADF F and the binary relation $\preceq$ having $prf_3(F)$ as least elements, containing the cycle $\mathbf{uft} \prec \mathbf{ttf} \prec \mathbf{fut} \prec \mathbf{tuf} \prec \mathbf{uft}$ and being an arbitrary linear order otherwise (cf. Figure 4.6). Note that $\preceq$ is not transitive and therefore only a pseudo-preorder. However, the revision operator $*$ induced by $\preceq$ can be shown to satisfy all postulates (A1$_{prf_3}$)–(A6$_{prf_3}$).

Moreover, every binary relation $\preceq'$ inducing the same operator $*$ must contain this cycle. Consider the pair of interpretations $\mathbf{uft}$ and $\mathbf{ttf}$. They are incompatible, hence $\{\mathbf{uft}, \mathbf{ttf}\}$ is realizable under prf_3 (cf. Proposition 34). To behave like the operator before, the revision of F by $\rho^{\mathrm{ADF}}_{prf_3}(\{\mathbf{uft}, \mathbf{ttf}\})$ must have $\mathbf{uft}$ as single preferred interpretation, hence $\mathbf{uft} \prec' \mathbf{ttf}$. This holds for every neighboring pair of the cycle, hence $\preceq'$ must contain the same non-transitive cycle. $\Diamond$

Therefore we will again use the postulate (Acyc$_{\sigma_3}$) to get a correspondence to i-max-faithful ranking.

(Acyc$_{\sigma_3}$) If for $1 \leq i < n$, $\sigma_3(F * G_{i+1}) \cap \sigma_3(G_i) \neq \emptyset$ and $\sigma_3(F * G_1) \cap \sigma_3(G_n) \neq \emptyset$ then $\sigma_3(F * G_n) \cap \sigma_3(G_1) \neq \emptyset$.

Similar as in the AF setting, we get that the rankings we are working with always have minimal elements.

Lemma 24. *Let $\preceq$ be an i-max-total preorder on $\mathcal{V}$. For each $V \in \Sigma_{ADF}^{prf}$ it holds that* $\min(V, \preceq) \neq \emptyset$.

We are now ready to give the first direction of the representation result, showing that every i-max-faithful assignment gives rise to an operator fulfilling the standard postulates and (Acyc$_{prf_3}$).

Theorem 33. *If there exists an i-max-faithful assignment mapping any ADF F to an i-max-faithful ranking $\preceq_F$ for F, then the revision operator $* : ADF_{\mathfrak{A}} \times ADF_{\mathfrak{A}} \mapsto ADF_{\mathfrak{A}}$ defined as*

$$F * G = \rho_{prf_3}^{ADF}(\min(prf_3(G), \preceq_F))$$

satisfies postulates ($A1_{prf_3}$)–($A6_{prf_3}$) and ($Acyc_{prf_3}$).

Proof. Let F and G be ADFs and $\preceq_F$ be the i-max-faithful ranking for F obtained from the i-max-faithful assignment. We show that $*$ satisfies (A1$_{prf_3}$)–(A6$_{prf_3}$) and (Acyc$_{prf_3}$).

By the definition of $\rho_{prf_3}^{\mathrm{ADF}}$, $\min(prf_3(G), \preceq_F)$ being non-empty by Lemma 24, and the fact that any non-empty $V \subseteq prf_3(G)$ is realizable under prf_3 (cf. Proposition 34.1) it holds that $prf_3(\rho_{prf_3}^{\mathrm{ADF}}(\min(prf_3(G), \preceq_F))) = \min(prf_3(G), \preceq_F)$, i.e. $prf_3(F * G) = \min(prf_3(G), \preceq_F)$. This equality not only shows that $*$ satisfies (A1$_{prf_3}$), but will also be useful throughout the proof.

For (A2$_{prf_3}$), assume $prf_3(F) \cap prf_3(G) \neq \emptyset$. Since $\preceq_F$ is i-max-faithful we get that $\min(prf_3(G), \preceq_F) = prf_3(F) \cap prf_3(G)$ and hence $prf_3(F * G) = prf_3(F) \cap prf_3(G)$.

Lemma 24 implies (A3$_{prf_3}$).

For (A4$_{prf_3}$) let H and F_2 be further ADFs and assume that $prf_3(F) = prf_3(F_2)$ and $prf_3(G) = prf_3(H)$. Since $\preceq_F$ and $\preceq_{F_2}$ are obtained from an i-max-faithful assignment it follows, by property (*iii*) of Definition 84, that $\preceq_F = \preceq_{F_2}$. Therefore also $\min(prf_3(G), \preceq_F) = \min(prf_3(H), \preceq_{F_2})$, i.e. $*$ satisfies (A4$_{prf_3}$).

For (A5$_{prf_3}$) and (A6$_{prf_3}$) we consider the non-trivial case where $prf_3(F * G) \cap prf_3(H) \neq \emptyset$. Recalling that $prf_3(G) \cap prf_3(H)$ is realizable under prf_3 for arbitrary ADFs G and H (cf. Proposition 28), we have to show that $\min(prf_3(G), \preceq_F) \cap prf_3(H) = \min(prf_3(G) \cap prf_3(H), \preceq_F)$. To the contrary assume there is some $v \in \min(prf_3(G), \preceq_F) \cap prf_3(H)$ such that $v \notin \min(prf_3(G) \cap prf_3(H), \preceq_F)$. As then $v \in prf_3(G)$ and $v \in prf_3(H)$ there must be some $v' \in prf_3(G) \cap prf_3(H)$ with $v' \prec_F v$, contradicting $v \in \min(prf_3(G), \preceq_F)$. On the other hand assume, again to the contrary, that there is some $v \in \min(prf_3(G) \cap prf_3(H), \preceq_F)$ such that $v \notin \min(prf_3(G), \preceq_F) \cap prf_3(H)$. From $v \in prf_3(H)$ we get $v \notin$

$\min(prf_3(G), \preceq_F)$. As by assumption $prf_3(F*G) \cap prf_3(H) \neq \emptyset$, let $v' \in \min(prf_3(G), \preceq_F)$ and $v' \in prf_3(H)$. Then also $v' \in prf_3(G) \cap prf_3(H)$. Since $v, v' \in prf_3(H)$, v and v' are incompatible, $\preceq_F$ is i-max-total and $v \in \min(prf_3(G) \cap prf_3(H), \preceq_F)$ by assumption, we get $v \preceq_F v'$. Thus $v \in \min(prf_3(G), \preceq_F)$ because $v' \in \min(prf_3(G), \preceq_F)$, a contradiction.

For (Acyc_{prf_3}) consider a sequence of ADFs $G_0, \ldots, G_n$ such that $prf_3(F * G_{i+1}) \cap prf_3(G_i) \neq \emptyset$ for $0 \leq i < n$ and $prf_3(F * G_0) \cap prf_3(G_n) \neq \emptyset$. Let $0 \leq i < n$. By definition of $*$ we have $prf_3(\rho^{\mathrm{ADF}}_{prf_3}(\min(prf_3(G_{i+1}), \preceq_F))) \cap prf_3(G_i) \neq \emptyset$. Then, as any subset of $prf_3(G_{i+1})$ is again realizable (cf. Proposition 34), $\min(prf_3(G_{i+1}), \preceq_F) \cap prf_3(G_i) \neq \emptyset$ follows. Hence there is some $v'_i \in prf_3(G_i)$ such that $v'_i \preceq_F v_{i+1}$ for all $v_{i+1} \in prf_3(G_{i+1})$. From transitivity of $\preceq_F$ we infer that there is a $v'_1 \in prf_3(G_1)$ such that $v'_1 \preceq_F v_n$ for all $v_n \in prf_3(G_n)$. From $prf_3(F * G_1) \cap prf_3(G_n) \neq \emptyset$ it follows that there is some $v''_1 \in \min(G_1, \preceq_F)$ (hence also $v''_1 \in prf_3(G_1)$ and $v''_1 \preceq_F v'_1$) with $v''_1 \in prf_3(G_n)$. We have $v''_1 \preceq_F v'_1 \preceq_F v_n$ (for each $v_n \in prf_3(G_n)$), hence $v''_1 \in \min(prf_3(G_n), \preceq_F)$. This together with $v''_1 \in prf_3(G_0)$ means that $prf_3(F * G_n) \cap prf_3(G_1) \neq \emptyset$, which was to show. $\square$

The second direction shows that the existence of an i-max-faithful assignment is also a necessary condition to get an operator satisfying the postulates.

Theorem 34. *Let $* : ADF_{\mathfrak{A}} \times ADF_{\mathfrak{A}} \mapsto ADF_{\mathfrak{A}}$ be a revision operator satisfying postulates ($A1_{prf_3}$)–($A6_{prf_3}$) and ($Acyc_{prf_3}$). Then there is an i-max-faithful assignment mapping each ADF F to an i-max-faithful ranking $\preceq_F$ for F such that $prf_3(F * G) = \min(prf_3(G), \preceq_F)$ for every ADF G.*

Proof. Assume an arbitrary ADF F. We define $\preceq_F$ and show that $\preceq_F$ as well as the corresponding assignment is i-max-faithful. Moreover, we will show that $prf_3(F * G) = \min(prf_3(G), \preceq_F)$ holds.

First let $\preceq'_F$ be the relation on $\mathcal{V}$ such that for each $v \in \mathcal{V}$, $v \approx'_F v$, and for any incompatible interpretations $v_1, v_2 \in \mathcal{V}$,

$$v_1 \preceq'_F v_2 \text{ if and only if } v_1 \in prf_3(F * \rho^{\mathrm{ADF}}_{prf_3}(\{v_1, v_2\})).$$

The relation $\preceq_F$ is defined as the transitive closure of $\preceq'_F$:

$$v \preceq_F v' \text{ if and only if } \exists w_1, \ldots, w_n : v \preceq'_F w_1 \preceq'_F \cdots \preceq'_F w_n \preceq'_F v'.$$

First, $\preceq_F$ is clearly reflexive and transitive, making it a preorder on $\mathcal{V}$. Moreover, for incompatible interpretations $v_1, v_2 \in \mathcal{V}$ we know from Proposition 34 that $\{v_1, v_2\}$ is realizable under prf_3, hence $prf_3(\rho^{\mathrm{ADF}}_{prf_3}(\{v_1, v_2\})) = \{v_1, v_2\}$. By ($\mathrm{A1}_{prf_3}$) and ($\mathrm{A3}_{prf_3}$) we therefore get that either $v_1 \preceq'_F v_2$ or $v_2 \preceq'_F v_1$, and, consequently, also $v_1 \preceq_F v_2$ or $v_2 \preceq_F v_1$, hence $\preceq_F$ is i-max-total.

We proceed by showing that $\preceq_F$ is i-max-faithful, i.e. we show properties (*i*) to (*iii*) from Definition 84. To show (*i*), let $v_1, v_2 \in prf_3(F)$ and note that $prf_3(\rho^{\mathrm{ADF}}_{prf_3}(\{v_1, v_2\})) = \{v_1, v_2\}$. Hence, by ($\mathrm{A2}_{prf_3}$), we get $prf_3(F * \rho^{\mathrm{ADF}}_{prf_3}(\{v_1, v_2\})) = \{v_1, v_2\}$. Therefore, by definition of $\preceq'_F$, $v_1 \preceq'_F v_2$ and $v_2 \preceq'_F v_1$, i.e. $v_1 \approx'_F v_2$. Hence also $v_1 \approx_F v_2$. For (ii), we begin with the intermediate statement

$$\text{for } v_1 \dots v_n \in \mathcal{V} : v_1 \preceq'_F \cdots \preceq'_F v_n \preceq'_F v_1 \text{ implies } v_1 \preceq'_F v_n \tag{4.1}$$

For $n \leq 2$ the statement is immediate. Assume $n > 2$. By definition of $\preceq'_F$ we first get that v_i and v_{i+1} for $1 \leq i < n$ as well as v_n and v_1 are incompatible, hence $\mathit{prf}_3(\rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_i, v_{i+1}\})) = \{v_i, v_{i+1}\}$ and $\mathit{prf}_3(\rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_n, v_1\}) = \{v_n, v_1\}$ by Proposition 34. Moreover, we get $v_i \in \mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_i, v_{i+1}\}))$ for $1 \leq i < n$ and $v_n \in \mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_n, v_1\}))$. It follows that $v_1 \in \mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_1, v_2\})) \cap \{v_n, v_1\}$, $v_i \in \mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_i, v_{i+1}\})) \cap \{v_{i-1}, v_i\}$ for $1 < i < n$, and $v_n \in \mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_n, v_1\})) \cap \{v_{n-1}, v_n\}$. Considering (Acyc$_{\mathit{prf}_3}$) we get $\mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_n, v_1\})) \cap \{v_1, v_2\} \neq \emptyset$, meaning further by (A5$_{\mathit{prf}_3}$) and (A6$_{\mathit{prf}_3}$) that $\mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_n, v_1\})) \cap \{v_1, v_2\} = \mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_n, v_1\} \cap \{v_1, v_2\})) = \mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_1\}))$. By $\mathit{prf}_3(\rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_1\})) = \{v_1\}$ (cf. Proposition 34), (A1$_{\mathit{prf}_3}$) and (A3$_{\mathit{prf}_3}$), it follows that $v_1 \in \mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_n, v_1\}))$, meaning that $v_1 \preceq'_F v_n$, concluding the proof for (4.1). We proceed by showing the statement

$$\text{for } v_1, v_2 \in \mathcal{V} : v_1 \prec'_F v_2 \text{ implies } v_1 \prec_F v_2 \tag{4.2}$$

$v_1 \preceq_F v_2$ is clear by definition. Assume, towards a contradiction, that $v_2 \preceq_F v_1$. Then $\exists w_1, \dots, w_n$ such that $v_1 \preceq'_F w_1 \preceq'_F \cdots \preceq'_F w_n \preceq'_F v_2$. As by assumption $v_1 \preceq'_F v_2$ it follows by (4.1) that $v_2 \preceq'_F v_1$, a contradiction to $v_1 \prec'_F v_2$, showing (4.2).

Now let v_1 and v_2 be incompatible interpretations such that $v_1 \in \mathit{prf}_3(F)$ and $v_2 \notin \mathit{prf}_3(F)$. By (A2$_{\mathit{prf}_3}$) we get $\mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_1, v_2\})) = \mathit{prf}_3(F) \cap \{v_1, v_2\} = \{v_1\}$, implying $v_1 \preceq'_F v_2$. Therefore, by (4.2), also $v_1 \preceq_F v_2$, showing (ii). Finally, consider another ADF F' with $\mathit{prf}_3(F) = \mathit{prf}_3(F')$. By (A4$_{\mathit{prf}_3}$) is holds that $\preceq'_F = \preceq'_{F'}$ and therefore also $\preceq_F = \preceq_{F'}$. Consequently $\preceq_F$ as well as the assignment it is obtained from is i-max-faithful.

Before showing that $*$ is indeed simulated by $\preceq_F$, we prove

$$\begin{gathered}\text{for } v_1, v_2 \in \mathcal{V} \text{ such that } v_1 \preceq'_F v_2 \text{ and } G \in \mathit{ADF}_{\mathfrak{A}} : \\ v_1 \in \mathit{prf}_3(G) \text{ and } v_2 \in \mathit{prf}_3(F * G) \text{ implies } v_1 \in \mathit{prf}_3(F * G)\end{gathered} \tag{4.3}$$

Let $G \in \mathit{ADF}_{\mathfrak{A}}$ such that $v_1 \in \mathit{prf}_3(G)$ and $v_2 \in \mathit{prf}_3(F * G)$. First note that, by $*$ fulfilling (A1$_{\mathit{prf}_3}$), also $v_2 \in \mathit{prf}_3(G)$, meaning that v_1 and v_2 are incompatible and therefore $\mathit{prf}_3(\rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_1, v_2\})) = \{v_1, v_2\}$. From (A5$_{\mathit{prf}_3}$) and (A6$_{\mathit{prf}_3}$) we then get that $\mathit{prf}_3(F * G) \cap \{v_1, v_2\} = \mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\mathit{prf}_3(G) \cap \{v_1, v_2\})) = \mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_1, v_2\}))$. By the assumption that $v_1 \preceq'_F v_2$ it holds that $v_1 \in \mathit{prf}_3(F * \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_1, v_2\}))$, hence (4.3) follows.

The last intermediate step is to show that

$$\text{for } G \in \mathit{ADF}_{\mathfrak{A}} : \min(\mathit{prf}_3(G), \preceq_F) = \min(\mathit{prf}_3(G), \preceq'_F) \tag{4.4}$$

Consider some $G \in ADF_{\mathfrak{A}}$. ($\subseteq$) Let $v_1 \in \min(prf_3(G), \preceq_F)$ and suppose there exists an $v_2 \in prf_3(G)$ with $v_2 \prec'_F v_1$. This means, by (4.2), that also $v_2 \preceq_F v_1$, a contradiction. Hence $v_2 \not\prec'_F v_1$ for all $v_2 \in prf_3(G)$, i.e. $v_1 \in \min(prf_3(G), \preceq'_F)$. ($\supseteq$) Let $v_1 \in \min(prf_3(G), \preceq'_F)$ and $v_2 \in prf_3(G)$. We show that $v_1 \preceq'_F v_2$, since then $v_1 \preceq_F v_2$ and, consequently, $v_1 \in \min(prf_3(G), \preceq_F)$ follows by definition of $\preceq_F$. If $v_1 = v_2$ we have $v_1 \preceq'_F v_2$ by definition of $\preceq'_F$. If $v_1 \neq v_2$ observe that, by $v_1, v_2 \in prf_3(G)$, v_1 and v_2 are incompatible, hence at least one of $v_1 \preceq'_F v_2$ and $v_2 \preceq'_F v_1$ must hold. By $v_1 \in \min(prf_3(G), \preceq'_F)$ it cannot hold that $v_2 \prec'_F v_1$, hence $v_1 \preceq'_F v_2$.

We are now ready to show that, for any ADF G, $prf_3(F * G) = \min(prf_3(G), \preceq_F)$. Considering (4.4) we just have to show that

$$\text{for } G \in ADF_{\mathfrak{A}} : prf_3(F * G) = \min(prf_3(G), \preceq'_F) \tag{4.5}$$

($\subseteq$) Let $v \in prf_3(F * G)$ and keep in mind that, by ($\mathrm{A1}_{prf_3}$), also $v \in prf_3(G)$. We show for each $w \in prf_3(G)$ that $v \preceq'_F w$. Consider an arbitrary $w \in prf_3(G)$. Note that by $v, w \in prf_3(G)$ we have that $prf_3(\rho^{\mathrm{ADF}}_{prf_3}(\{v, w\})) = \{v, w\}$. From ($\mathrm{A5}_{prf_3}$) and ($\mathrm{A6}_{prf_3}$) we get $prf_3(F * G) \cap \{v, w\} = prf_3(F * \rho^{\mathrm{ADF}}_{prf_3}(prf_3(G) \cap \{v, w\})) = prf_3(F * \rho^{\mathrm{ADF}}_{prf_3}(\{v, w\}))$. As by assumption $v \in prf_3(F * G)$ we get $v \preceq'_F w$ by definition of $\preceq'_F$. ($\supseteq$) Towards a contradiction, assume some $v \in \min(prf_3(G), \preceq'_F)$ such that $v \notin prf_3(F * G)$ (again note that also $v \in prf_3(G)$ by ($\mathrm{A1}_{prf_3}$)). By ($\mathrm{A3}_{prf_3}$) and the fact that $prf_3(G) \neq \emptyset$ there is some $w \in prf_3(F * G)$. From (4.3) we infer that $v \not\preceq'_F w$. But by assumption also $w \not\preceq'_F v$. Since v and w must be incompatible by $v, w \in prf_3(G)$, this means $prf_3(F * \rho^{\mathrm{ADF}}_{prf_3}(\{v, w\})) \cap \{v, w\} = \emptyset$ and by $*$ fulfilling ($\mathrm{A1}_{prf_3}$) even $prf_3(F * \rho^{\mathrm{ADF}}_{prf_3}(\{v, w\})) = \emptyset$, a contradiction to $*$ satisfying ($\mathrm{A3}_{prf_3}$). □

With Theorems 33 and 34 we have obtained a one-to-one correspondence between revision operators induced by i-max-faithful rankings and revision operators satisfying postulates ($\mathrm{A1}_{prf_3}$)–($\mathrm{A6}_{prf_3}$) and (Acyc_{prf_3}).

To exemplify the obtained result, we introduce a three-valued version of Dalal's operator. We first have to define a suitable distance measure for three-valued interpretations, a measure also used, for instance, by Arieli [8].

Definition 85. The symmetric distance function $\triangle$ between truth values is defined as follows:

$$\begin{aligned} \mathbf{t} \triangle \mathbf{f} = \mathbf{f} \triangle \mathbf{t} &= 1, \\ \mathbf{t} \triangle \mathbf{u} = \mathbf{f} \triangle \mathbf{u} = \mathbf{u} \triangle \mathbf{t} = \mathbf{u} \triangle \mathbf{f} &= \frac{1}{2}, \\ \mathbf{t} \triangle \mathbf{t} = \mathbf{f} \triangle \mathbf{f} = \mathbf{u} \triangle \mathbf{u} &= 0. \end{aligned}$$

For interpretations $v_1, v_2 \in \mathcal{V}$, their distance function $\triangle : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{N}$ is then defined as

$$v_1 \triangle v_2 = \sum_{a \in A} v_1(a) \triangle v_2(a).$$

Based on this distance measure we can define the ranking giving rise to the three-valued version of Dalal's revision operator.

Definition 86. Given an ADF F and semantics σ_3, the ranking $\preceq_F^{\sigma_3}$ based on three-valued distance is defined as

$$v_1 \preceq_F^{\sigma_3} v_2 \text{ if and only if } \min_{v \in \sigma_3(F)} (v \triangle v_1) \leq \min_{v \in \sigma_3(F)} (v \triangle v_2)$$

for each $v_1, v_2 \in \mathcal{V}$.

The operator $*_{\sigma_3}^D$ induced by $\preceq_F^{\sigma_3}$ returns $F *_{\sigma_3}^D G = \rho_{prf_3}^{\mathrm{ADF}}(\min(\sigma_3(G), \preceq_F^{\sigma_3}))$ for each $G \in ADF_{\mathfrak{A}}$.

It is easy to see that $\preceq_F^{\sigma_3}$ is i-max-faithful, as the minimal distance to $\sigma_3(F)$ is 0 for interpretations $v \in \sigma_3(F)$ and greater than 0 for all interpretations $v \notin \sigma_3(F)$. Hence, by Theorem 33, the ranking for preferred semantics $*_{prf_3}^D$ satisfies all postulates ($\mathrm{A1}_{prf_3}$)–($\mathrm{A6}_{prf_3}$) and (Acyc_{prf_3}).

We show the behaviour of this operator in the following example.

Example 49. Consider the ADF $F = \{\langle a, a\rangle, \langle b, a\rangle, \langle c, \neg a \wedge b\rangle\}$, and observe that $prf_3(F) = \{\mathbf{ttf}, \mathbf{fff}\}$. First note that the minimal elements of $\preceq_F^{prf_3}$ coincide with $prf_3(F)$, i.e.

$$\mathbf{ttf} \approx_F^{prf_3} \mathbf{fff} \prec_F^{prf_3} \textit{others}.$$

Now consider the revision by the ADF G having $prf_3(G) = \{\mathbf{tft}, \mathbf{ttu}, \mathbf{ffu}\}$ (e.g. the ADF of the form $G = \{\langle a, a \wedge (b \vee c)\rangle, \langle b, a \wedge b\rangle, \langle c, (a \wedge (\neg b \vee \neg c)) \vee (\neg b \wedge \neg c)\rangle\}$) and observe that

$$\mathbf{ttu} \approx_F^{prf_3} \mathbf{ffu} \prec_F^{prf_3} \mathbf{tft}$$

This is because $\mathbf{ttu}$ and $\mathbf{ffu}$ have minimal distance to $prf_3(F)$ of $\frac{1}{2}$, while $\mathbf{tft}$ has minimal distance of 2. Therefore we get $F *_{prf_3}^D G = \rho_{prf_3}^{\mathrm{ADF}}(\{\mathbf{ttu}, \mathbf{ffu}\})$.

On the other hand consider the ADF $G' = \{\langle a, \top\rangle, \langle b, \neg a\rangle, \langle c, \neg b\rangle\}$, having $prf_3(G') = \{\mathbf{tft}\}$. The revision of F by G' obviously results in an ADF also having $\mathbf{tft}$ – minimal distance 2 to $prf_3(F)$ – as only preferred interpretation. Inspecting the set of admissible interpretations of G', which can be seen as reasonable (but not maximal) positions in the revising ADF, $adm_3(G') = \{\mathbf{tft}, \mathbf{tfu}, \mathbf{tuu}, \mathbf{uuu}\}$, we observe that it contains elements which are closer to $prf_3(F)$ than $\mathbf{tft}$. In particular, the interpretation $\mathbf{tuu}$ has distance 1 to $prf_3(F)$ and is even admissible in F. $\Diamond$

The latter part of the previous example already suggests that revision under preferred semantics can lead to results which can be considered counterintuitive. Admissible interpretations which are not maximal are not taken into account by the operator. After showing in the next section that revising under admissible interpretations is also not satisfactory, we will present an approach combining these two variants in Section 4.3.3.

4.3.2 Revision under Admissible Semantics

Example 49 suggests to take the admissible interpretations into account when revising with respect to the preferred interpretations. A quite radical step would be to just revise with respect to admissible interpretations instead. But by the fact that $adm_3(F_1) \cap adm_3(F_2) \neq \emptyset$ for all ADFs $F_1, F_2 \in ADF_{\mathfrak{A}}$ we get only one operator satisfying postulate (A2$_{adm_3}$). The following result then immediately follows.

Theorem 35. *An operator $* : ADF_{\mathfrak{A}} \times ADF_{\mathfrak{A}} \mapsto ADF_{\mathfrak{A}}$ fulfills $(A1_{adm_3})$–$(A6_{adm_3})$ if and only if $*$ is defined as*

$$F * G = \rho^{ADF}_{adm_3}(adm_3(F) \cap adm_3(G))$$

for any ADFs F and G.

It is important to note that admissible semantics is closed under intersection (cf. Proposition 28), therefore $\rho^{\mathrm{ADF}}_{adm_3}(adm_3(F) \cap adm_3(G))$ always realizes $adm_3(F) \cap adm_3(G)$. We illustrate this operator in the following example.

Example 50. Again consider the ADFs F and G' from Example 49 and note that $adm_3(F) = \{\mathbf{ttf}, \mathbf{fff}, \mathbf{ttu}, \mathbf{tuf}, \mathbf{ffu}, \mathbf{tuu}, \mathbf{fuu}, \mathbf{uuu}\}$ and $adm_3(G') = \{\mathbf{tft}, \mathbf{tfu}, \mathbf{tuu}, \mathbf{uuu}\}$. Moreover, let $*_{adm_3}$ be the operator from Theorem 35. As expected, we get $F *_{adm_3} G' = \rho^{\mathrm{ADF}}_{adm_3}(\{\mathbf{tuu}, \mathbf{uuu}\})$, i.e. the resulting ADF has **tuu** as single preferred interpretation, which was seen as one of the more desired scenarios in Example 49.

But now consider the ADF G'' having $adm_3(G'') = \{\mathbf{utf}, \mathbf{uuu}\}$ (for instance $G'' = \{\langle a, \neg a\rangle, \langle b, \neg b \vee \neg c\rangle, \langle c, \neg b \wedge \neg c\rangle\}$) and observe that $F *_{adm_3} G'' = \rho^{\mathrm{ADF}}_{adm_3}(\{\mathbf{uuu}\})$. From the perspective of the preferred interpretations of F (being $\{\mathbf{ttf}, \mathbf{fff}\}$) this might not be desired, as **utf** is admissible in G'' and has a distance of only $\frac{1}{2}$ to $prf_3(F)$, while the result of the revision has distance $\frac{3}{2}$. ♢

4.3.3 Hybrid Approach

Due to the problems illustrated in Examples 49 and 50 we are interested in taking both admissible and preferred semantics into account when revising ADFs. In this section we do so by studying revision operators $\star : ADF_{\mathfrak{A}} \times ADF_{\mathfrak{A}} \mapsto ADF_{\mathfrak{A}}$ that select out of the admissible interpretations of the revising ADF (in a sense accepting all reasonable positions as valid outcomes of the revision), but base the amount of change on the preferred interpretations of the original ADF. To this end we reformulate the postulates to this setting:

(H1) $prf_3(F \star G) \subseteq adm_3(G)$.

(H2) If $prf_3(F) \cap adm_3(G) \neq \emptyset$, then $prf_3(F \star G) = prf_3(F) \cap adm_3(G)$.

(H3) If $adm_3(G) \neq \emptyset$, then $prf_3(F \star G) \neq \emptyset$.

(H4) If $\mathit{prf}_3(F_1) = \mathit{prf}_3(F_2)$ and $\mathit{adm}_3(G) = \mathit{adm}_3(H)$, then $\mathit{prf}_3(F_1 \star G) = \mathit{prf}_3(F_2 \star H)$.

(H5) $\mathit{prf}_3(F \star G) \cap \mathit{adm}_3(H) \subseteq \mathit{prf}_3(F \star \rho^{\text{ADF}}_{\mathit{adm}_3}(\mathit{adm}_3(G) \cap \mathit{adm}_3(H)))$.

(H6) If $\mathit{prf}_3(F \star G) \cap \mathit{adm}_3(H) \neq \emptyset$, then $\mathit{prf}_3(F \star \rho^{\text{ADF}}_{\mathit{adm}_3}(\mathit{adm}_3(G) \cap \mathit{adm}_3(H))) \subseteq \mathit{prf}_3(F \star G) \cap \mathit{adm}_3(H)$.

(HAcyc) If for $1 \leq i < n$, $\mathit{prf}_3(F \star G_{i+1}) \cap \mathit{adm}_3(G_i) \neq \emptyset$ and $\mathit{prf}_3(F \star G_1) \cap \mathit{adm}_3(G_n) \neq \emptyset$ then $\mathit{prf}_3(F \star G_n) \cap \mathit{adm}_3(G_1) \neq \emptyset$.

As admissible semantics may give pairwise compatible interpretations, we will not restrict ourselves to i-max-faithful rankings for the representation result. However, we face another challenge, as illustrated in the following example.

Example 51. Consider the preorder $\preceq$ given by $\mathbf{ff} \prec \mathit{others} \prec \mathbf{tu} \approx \mathbf{ut} \prec \mathbf{tt} \prec \mathbf{uu}$ and the ADFs

$$F = \{\langle a, \bot\rangle, \langle b, \bot\rangle\},$$
$$G = \{\langle a, \top\rangle, \langle b, \top\rangle\}, \text{ and}$$
$$H = \{\langle a, \neg a \vee b\rangle, \langle b, a \vee \neg b\rangle\}.$$

We have $\mathit{prf}_3(F) = \{\mathbf{ff}\}$, $\mathit{adm}_3(G) = \{\mathbf{uu}, \mathbf{ut}, \mathbf{tu}, \mathbf{tt}\}$, and $\mathit{adm}_3(H) = \{\mathbf{uu}, \mathbf{tt}\}$. It can be seen that $\preceq$ is a faithful ranking for F. However, the revision operator $\star$ induced by $\preceq$ gives us $F \star G = \rho^{\text{ADF}}_{\mathit{prf}_3}(\min(\mathit{adm}_3(G), \preceq)) = \rho^{\text{ADF}}_{\mathit{prf}_3}(\{\mathbf{ut}, \mathbf{tu}\})$ and we further get

- $\mathit{prf}_3(F \star G) \cap \mathit{adm}_3(H) = \{\mathbf{uu}\}$, but
- $\mathit{prf}_3(F \star \rho^{\text{ADF}}_{\mathit{adm}_3}(\mathit{adm}_3(G) \cap \mathit{adm}_3(H))) = \{\mathbf{tt}\}$.

Therefore $\star$ violates (H5) and (H6). The problem arises because of the fact that $\mathbf{ut}$ and $\mathbf{tu}$ are compatible. That is, the set of interpretations $\{\mathbf{ut}, \mathbf{tu}\}$ is not incompatible and hence cannot be realized under preferred semantics (cf. Theorem 24). Therefore, $\mathit{prf}_3(\rho^{\text{ADF}}_{\mathit{prf}_3}(\{\mathbf{ut}, \mathbf{tu}\})) = \{\mathbf{uu}\}$. Again, the choice of realizing the set $\{v_{\mathbf{u}}\}$ instead of an unrealizable interpretation-set is an arbitrary one. However, similar as in Example 43, also alternative definitions of $\rho^{\text{ADF}}_{\mathit{prf}_3}$ lead to the same problem. $\Diamond$

To overcome this issue we introduce the concept of compliance, a notion similarly used in work on revision of logic programs [85], Horn theories [84], and AFs by propositional formulas [89]. The following definition of compliance differs from the previous versions in that it is parametrized by two semantics.

Definition 87. Given two semantics σ and τ, a preorder $\preceq$ is σ-τ-compliant if, for every ADF $G \in \mathit{ADF}_{\mathfrak{A}}$, it holds that $\min(\tau(G), \preceq) \in \Sigma^{\sigma}_{\mathit{ADF}}$.

The intuitive idea of a σ-τ-compliant ranking $\preceq$ is that using $\preceq$ to select the most plausible results of the evaluation of a knowledge base under τ gives a result that is realizable under σ.

We will make use of the following properties of the adm-closure (cf. Definition 66) in the upcoming results. Note that, in the remainder of this section, we will often denote $cl(\{v_1, v_2\})$ for arbitrary pairs of interpretations $v_1, v_2 \in \mathcal{V}$ by $cl(v_1, v_2)$.

Lemma 25. *For each $V, V_1, V_2 \subseteq \mathcal{V}$ and $v, v' \in \mathcal{V}$ it holds:*

1. $cl(V) = cl(cl(V))$ *(*idempotence*)*
2. $V_1 \subseteq V_2 \Rightarrow cl(V_1) \subseteq cl(V_2)$ *(*monotonicity*)*
3. $\forall v'' \in cl(v, v') : cl(v, v'') \subseteq cl(v, v')$*.*

Proof. Note that $V \subseteq cl(V)$ for any $V \subseteq \mathcal{V}$ is clear by definition.

(1) $cl(V) \subseteq cl(cl(V))$ follows from the initial observation. To show that also $cl(V) \supseteq cl(cl(V))$, assume there is some $v \in cl(cl(V))$ with $v \notin cl(V)$. The latter means that there is some $a \in (v^{\mathbf{t}} \cup v^{\mathbf{f}})$ and some $v_2 \in [v]_2$ such that there is no $v' \in V$ with $v' \leq_i v_2 \wedge v'(a) = v(a)$. Now for these particular a and v_2 it holds, by $v \in cl(cl(V))$, that there is some $w \in cl(V)$ such that $w \leq_i v_2 \wedge w(a) = v(a)$. In order for $w \in cl(V)$ it must hold that there is some $w' \in V$ with $w' \leq_i v_2$ and $w'(a) = w(a)$. We have $w'(a) = w(a) = v(a)$, a contradiction to the fact that there is no $v' \in V$ with $v' \leq_i v_2 \wedge v'(a) = v(a)$.

(2) Let $v \in cl(V_1)$ and consider arbitrary $a \in v^{\mathbf{t}} \cup v^{\mathbf{f}}$ and $v_2 \in [v]_2$. It must hold that there is some $v' \in V_1$ such that $v' \leq_i v_2$ and $v(a) = v'(a)$. As $V_1 \subseteq V_2$ by assumption, also $v' \in V_2$, hence $v \in cl(V_2)$.

(3) Consider some $v'' \in cl(v, v')$, i.e. for each $a \in v''^{\mathbf{t}} \cup v''^{\mathbf{f}}$ and each $v_2 \in [v'']_2$ it holds that $v \leq_i v_2 \wedge v(a) = v''(a)$ or $v' \leq_i v_2 \wedge v'(a) = v''(a)$. Assume there is some $w \in cl(v, v'')$ and $w \notin cl(v, v')$. The latter means that there is some $a \in w^{\mathbf{t}} \cup w^{\mathbf{f}}$ and some $w_2 \in [w]_2$ such that neither $v \leq_i w_2 \wedge v(a) = w(a)$ nor $v' \leq_i w_2 \wedge v'(a) = w(a)$ holds. Hence, by $w \in cl(v, v'')$, we get for this particular a and w_2 that $v'' \leq_i w_2$ and $v''(a) = w(a)$. From $a \in w^{\mathbf{t}} \cup w^{\mathbf{f}}$ and $v''(a) = w(a)$ it follows that $a \in v''^{\mathbf{t}} \cup v''^{\mathbf{f}}$ and from $v'' \leq_i w_2$ we get $w_2 \in [v'']_2$. Therefore, from $v'' \in cl(v, v')$ and $\neg(v' \leq_i w_2 \wedge v'(a) = w(a))$, we get $v \leq_i w_2$ and $v(a) = v''(a)$ and, consequently, $v(a) = w(a)$, a contradiction. $\square$

We now show the representation result for our hybrid operators which work on the admissible interpretations of the revising ADF but basing the distance measure on the preferred interpretations of the original ADF. The first direction follows similar to Theorem 33 with the help of prf_3-adm_3-compliance.

Theorem 36. *If there exists a faithful assignment mapping each ADF F to a a prf_3-adm_3-compliant, faithful ranking $\preceq_F$ for F, then the revision operator $\star : ADF_{\mathfrak{A}} \times ADF_{\mathfrak{A}} \mapsto ADF_{\mathfrak{A}}$ defined as*

$$F \star G = \rho^{ADF}_{prf_3}(\min(adm_3(G), \preceq_F))$$

satisfies postulates $(H1)$–$(H6)$ *and* $(HAcyc)$.

Proof. Let F be an ADF and $\preceq_F$ be the i-max-faithful ranking for F that is prf_3-adm_3-compliant. First note that, since $\preceq_F$ is prf_3-adm_3-compliant, it holds, for any ADF G, that $\min(\mathit{adm}_3(G), \preceq_F) \in \Sigma^{\mathit{prf}_3}_{\mathrm{ADF}}$. Hence $\mathit{prf}_3(\rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\min(\mathit{adm}_3(G), \preceq_F))) = \min(\mathit{adm}_3(G), \preceq_F)$ and, consequently, $\mathit{prf}_3(F \star G) = \min(\mathit{adm}_3(G), \preceq_F)$ for every ADF G.

(H1) follows immediately. (H2) holds since $\preceq_F$ is faithful for F. By finiteness of $\mathit{adm}_3(G)$ and transitivity of $\preceq_F$, $\min(\mathit{adm}_3(G), \preceq_F) \neq \emptyset$ holds, i.e. (H3) is satisfied. If $\mathit{prf}_3(F) = \mathit{prf}_3(F_2)$ and $\mathit{adm}_3(G) = \mathit{adm}_3(H)$, for further ADFs F_2, G, and H, then $\preceq_F = \preceq_{F_2}$ as they are obtained from a faithful assignment, and, consequently, $\min(\mathit{adm}(G), \preceq_F) = \min(\mathit{adm}(H), \preceq_{F_2})$, showing the (H4) holds. (H5) and (H6) can be shown analogously to Theorem 33, keeping in mind that also $\mathit{adm}_3(G) \cap \mathit{adm}_3(H)$ is realizable under adm_3 for arbitrary ADFs G and H. Finally, also (HAcyc) can be shown just as in Theorem 33. □

The other direction of the proof differs from the previous ones, as we have to construct a total preorder, but testing the operator on pairs of interpretations does not always give insight about the desired ranking. More formally, given two interpretations v_1 and v_2, it is not guaranteed that $\{v_1, v_2\} \in \Sigma^{\mathit{adm}_3}_{\mathrm{ADF}}$. Therefore we build the adm-closure to construct an ADF containing v_1 and v_2, i.e. $\rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(\{v_1, v_2\}))$. However, for a rational operator and an ADF F, we can get now that $\{v_1, v_2\} \cap \mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(\{v_1, v_2\})) = \emptyset$. Hence we have no indication whether to prefer v_1 or v_2. In the course of the proof we will show how we overcome this issue.

Theorem 37. *Let* $\star : \mathit{ADF}_{\mathfrak{A}} \times \mathit{ADF}_{\mathfrak{A}} \mapsto \mathit{ADF}_{\mathfrak{A}}$ *be a revision operator satisfying postulates* $(H1)$–$(H6)$ *and* $(HAcyc)$. *Then there is a faithful assignment mapping every ADF F to a faithful ranking $\preceq_F$ for F that is prf_3-adm_3-compliant and such that* $\mathit{prf}_3(F \star G) = \min(\mathit{adm}_3(G), \preceq)$ *for every ADF G.*

Proof. Let $\star : \mathit{ADF}_{\mathfrak{A}} \times \mathit{ADF}_{\mathfrak{A}} \mapsto \mathit{ADF}_{\mathfrak{A}}$ be an operator satisfying (H1)–(H6) and (HAcyc) and F be an arbitrary ADF. We will gradually define the ranking $\preceq$ and show that it is faithful for F and prf_3-adm_3-compliant and it indeed simulates $\star$ by $\mathit{prf}_3(F \star G) = \min(\mathit{adm}_3(G), \preceq)$.

Throughout out the proof, we will make use of the following observation: given an arbitrary set of interpretations $V \subseteq \mathcal{V}$, $\mathit{cl}(V) = \mathit{cl}(\mathit{cl}(V))$ (cf. Lemma 25.1), hence $\mathit{cl}(V) \in \Sigma^{\mathit{adm}_3}_{\mathrm{ADF}}$ and, consequently, $\mathit{adm}_3(\rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(V))) = \mathit{cl}(V)$.

First, we define $\preceq'$ as $v \preceq' v$ for each $v \in \mathcal{V}$ and

$$v_1 \preceq' v_2 \text{ if and only if } v_1 \in \mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(v_1, v_2)))$$

for each $v_1, v_2 \in \mathcal{V}$ with $v_1 \neq v_2$. Note that $\preceq'$ is reflexive, but neither transitive nor total. The latter is because there might be interpretations $v_1, v_2 \in \mathcal{V}$ for which

$prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2))) \cap \{v_1, v_2\} = \emptyset$ due to $cl(v_1, v_2) \supset \{v_1, v_2\}$. After showing three properties of $\preceq'$ we will extend it first to the transitive ranking $\preceq^t$ and then to the desired ranking $\preceq$.

Lemma 26. *For any $v_1, v_2 \in \mathcal{V}$ such that $v_1 \preceq' v_2$ and any $G \in ADF_{\mathfrak{A}}$, it holds that if $v_1 \in adm_3(G)$ and $v_2 \in prf_3(F \star G)$ then $v_1 \in prf_3(F \star G)$.*

Proof. Let $G \in ADF_{\mathfrak{A}}$, $v_1 \in adm_3(G)$, $v_2 \in prf_3(F \star G)$ with $v_1 \preceq' v_2$. First, we get $v_2 \in adm_3(G)$ from (H1). Moreover, from (H5) and (H6) we get $prf_3(F \star G) \cap cl(v_1, v_2) = prf_3(F \star \rho^{\text{ADF}}_{adm_3}(adm_3(G) \cap cl(v_1, v_2))$. As both $v_1, v_2 \in adm_3(G)$ and $cl(adm_3(G)) = adm_3(G)$ we get that $cl(v_1, v_2) \subseteq adm_3(G)$ from Lemma 25.2, hence $prf_3(F \star G) \cap cl(v_1, v_2) = prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2))$ by (H4). Now as $v_1 \preceq' v_2$ by assumption it must hold that $v_1 \in prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2)))$, hence $v_1 \in prf_3(F \star G)$. □

Lemma 27. *For each $G \in ADF_{\mathfrak{A}}$ it holds that* $\min(adm_3(G), \preceq') = prf_3(F \star G)$.

Proof. $\subseteq$: Towards a contradiction, assume there is some $v_1 \in \min(adm_3(G), \preceq')$ such that $v_1 \notin prf_3(F \star G)$. From (H3) we know $prf_3(F \star G) \neq \emptyset$, so assume an arbitrary $v_2 \in prf_3(F \star G)$. From Lemma 26 it follows that $v_1 \not\preceq' v_2$ and, consequently, from $v_1 \in \min(adm_3(G), \preceq')$ also $v_2 \not\preceq' v_1$. By the definition of $\preceq'$ this means that $v_1, v_2 \notin prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2)))$ and, considering (H3), there must then be some $v_3 \in prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2)))$. From (H1) it follows that $v_3 \in adm_3(\rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2)))$, i.e. $v_3 \in cl(v_1, v_2)$. Then from (H5) and (H6) we get

$$prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2))) \cap cl(v_1, v_3) = prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2) \cap cl(v_1, v_3))).$$

From Lemma 25.3 it follows that $cl(v_1, v_3) \subseteq cl(v_1, v_2)$, hence

$$prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2))) \cap cl(v_1, v_3) = prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_3)))$$

by (H4). Recalling that $v_1 \notin prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2)))$ and $v_3 \in prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2)))$ it follows that $v_1 \notin prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_3)))$ and $v_3 \in prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_3)))$, hence $v_3 \prec' v_1$. Finally, note that, by $v_1, v_2 \in adm_3(G)$ and $cl(adm_3(G)) = adm_3(G)$, $cl(v_1, v_2) \subseteq adm_3(G)$, hence $v_3 \in adm_3(G)$, a contradiction to $v_1 \in \min(adm_3(G), \preceq')$.

$\supseteq$: Let $v_1 \in prf_3(F \star G)$ and consider an arbitrary $v_2 \in adm_3(G)$. Observing $v_1 \in adm_3(G)$ by (H1) we get $prf_3(F \star G) \cap cl(v_1, v_2) = prf_3(F \star \rho^{\text{ADF}}_{adm_3}(adm_3(G) \cap cl(v_1, v_2)))$ by (H5) and (H6). Moreover, $cl(v_1, v_2) \subseteq adm_3(G)$ by Lemma 25.2, hence $prf_3(F \star G) \cap cl(v_1, v_2) = prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2)))$ by (H4) and, consequently, $v_1 \in prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_1, v_2)))$, meaning $v_1 \preceq' v_2$. Therefore, recalling that v_2 was chosen arbitrarily, $v_1 \in \min(adm_3(G), \preceq')$. □

We show the following similarly as (4.1).

Lemma 28. *Given interpretations $v_1, \ldots, v_n \in \mathcal{V}$, it holds that if $v_1 \preceq' \cdots \preceq' v_n \preceq' v_1$ then $v_1 \preceq' v_n$.*

Proof. For $n \leq 2$ the statement is immediate. Assume $n > 2$. By definition of $\preceq'$ we first get that $v_i \in \mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_i, v_{i+1}\}))$ for $1 \leq i < n-1$ and $v_n \in \mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{prf}_3}(\{v_n, v_1\}))$. It follows that

$$\begin{aligned} v_1 &\in \mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(v_1, v_2))) \cap \mathit{cl}(v_n, v_1), \\ v_i &\in \mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(v_i, v_{i+1}))) \cap \mathit{cl}(v_{i-1}, v_i) \text{ for } 1 < i < n, \text{ and} \\ v_n &\in \mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(v_n, v_1))) \cap \mathit{cl}(v_{n-1}, v_n). \end{aligned}$$

Considering (HAcyc) we get $\mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}}(\mathit{cl}(v_n, v_1))) \cap \mathit{cl}(v_1, v_2) \neq \emptyset$, meaning further by (H5) and (H6) that

$$\mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(v_n, v_1))) \cap \mathit{cl}(v_1, v_2) = \mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(v_n, v_1) \cap \mathit{cl}(v_1, v_2))).$$

Moreover, from $\mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(v_1, v_2))) \cap \mathit{cl}(v_n, v_1) \neq \emptyset$ we derive by (H5) and (H6) that

$$\mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(v_1, v_2))) \cap \mathit{cl}(v_n, v_1) = \mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(v_1, v_2) \cap \mathit{cl}(v_n, v_1))).$$

Hence, from $v_1 \in \mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(v_1, v_2))) \cap \mathit{cl}(v_n, v_1)$ it follows that $v_1 \in \mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(v_1, v_2) \cap \mathit{cl}(v_n, v_1)))$ and, consequently, $v_1 \in \mathit{prf}_3(F \star \rho^{\mathrm{ADF}}_{\mathit{adm}_3}(\mathit{cl}(v_n, v_1)))$. Therefore $v_1 \preceq' v_n$. $\square$

Now we define $\preceq^t$ to be the transitive closure of $\preceq'$. As a consequence of Lemma 28 we infer, as in Lemma 19, that

$$\text{for } v_1, v_2 \in \mathcal{V} : v_1 \prec' v_2 \text{ implies } v_1 \prec^t v_2 \tag{4.6}$$

We define, for any interpretation-set V, $\max(V, \preceq^t)$ as the set $\{v_1 \in V \mid \nexists v_2 \in V : v_1 \prec^t v_2\}$. We get, by Lemma 28 and the fact that $\mathcal{V}$ is finite, that

$$\text{for } V \subseteq \mathcal{V} : V \neq \emptyset \text{ implies } \max(V, \preceq^t) \neq \emptyset \tag{4.7}$$

We are now ready to define $\preceq$. To this end consider the sequence of sets of interpretations $V_0, V_1, \ldots$ defined as

$$\begin{aligned} V_0 &= \max(\mathcal{V}, \preceq^t), \\ V_1 &= \max\left(\mathcal{V} \setminus V_0, \preceq^t\right), \\ V_i &= \max(\mathcal{V} \setminus \bigcup_{0 \leq j < i} V_j, \preceq^t) \text{ for } i > 1. \end{aligned}$$

Since $\mathcal{V}$ is finite we conclude from (4.7) that the sequence will reach the empty set of interpretations at some point and each of the following elements will also be empty.

The sequence $V_1, \dots, V_m$ of non-empty sets of interpretation then forms a partition of $\mathcal{V}$. Based on this we define $\preceq$ as

$$v_1 \preceq v_2 \text{ if and only if } \exists V_i, V_j \text{ s.t. } v_1 \in V_i, v_2 \in V_j, i \geq j$$

for each $v_1, v_2 \in \mathcal{V}$. As each interpretation is contained in exactly one set of the sequence, it is easy to see that $\preceq$ is total, reflexive, and transitive. It remains to show that its minimal elements coincide with $\preceq'$.

Lemma 29. *For each $G \in ADF_{\mathfrak{A}}$ it holds that* $\min(adm_3(G), \preceq) = \min(adm_3(G), \preceq')$.

Proof. Let V_k be the last set in the sequence $V_0, \dots, V_m$ such that $V_k \cap adm_3(G) \neq \emptyset$. By definition of $\preceq$, $\min(adm_3(G), \preceq) = V_k \cap adm_3(G)$. Hence we have to show that $V_k \cap adm_3(G) = \min(adm_3(G), \preceq')$.

$\subseteq$: Assume there is some $v \in V_k \cap adm_3(G)$ such that $v \notin \min(adm_3(G), \preceq')$. From the latter it follows that $\exists v_0 \in adm_3(G) : v_0 \prec' v$. From (4.6) we get $v_0 \prec^t v$, hence $v_0 \notin \max(V_k, \preceq^t)$. As V_k is the last set with $V_k \cap adm_3(G) \neq \emptyset$ it must hold that $v_0 \in V_j$ with $j < k$, i.e. $v_0 \in \max(\mathcal{V} \setminus \bigcup_{0 \leq i < j} V_i, \preceq^t)$. Therefore, recalling $v_0 \prec^t v$, $v \notin \mathcal{V} \setminus \bigcup_{0 \leq i < j} V_i$, contradicting $v \in V_k$ and $j < k$.

$\supseteq$: Assume there is some $v_0 \in \min(adm_3(G), \preceq')$ such that $v_0 \notin V_k \cap adm_3(G)$. That means $v_0 \in adm_3(G)$ and $v_0 \notin V_k$ and further that $v_0 \in V_j$ for some $j < k$. Now let $v_1 \in V_k \cap adm_3(G)$. As $j < k$ it holds that $v_1 \in \mathcal{V} \setminus \bigcup_{0 \leq i < j} V_i$. Since v_0 is maximal wrt. $\preceq^t$ in this set, $v_0 \not\prec^t v_1$ and further, by the contrapositive of (4.6), $v_0 \not\prec' v_1$. It holds that $v_0 \in prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_0, v_1)))$ and therefore $v_0 \preceq' v_1$ though. We show this by assuming, towards a contradiction, that $v_0 \notin prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_0, v_1)))$. Hence $v_0 \not\preceq' v_1$. As $v_0 \in \min(adm_3(G), \preceq')$ by assumption and $v_1 \in adm_3(G)$, then also $v_1 \not\preceq' v_0$. By (H3) there has to be some $v_2 \in prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_0, v_1)))$. As also $v_2 \in cl(v_0, v_2)$ we get by (H5) and (H6) that $v_2 \in prf_3(F \star \rho^{\text{ADF}}_{adm_3}(cl(v_0, v_1) \cap cl(v_0, v_2)))$. From Lemma 25.2 we infer that $cl(v_0, v_2) \subseteq cl(v_0, v_1)$, hence $v_2 \in prf_3(F \star \rho^{\text{ADF}}_{adm_3}(v_0, v_2))$ by (H4), meaning that $v_2 \preceq' v_0$. Moreover, $v_0 \notin prf_3(F \star \rho^{\text{ADF}}_{adm_3}(v_0, v_2))$, hence even $v_2 \prec' v_0$. As $v_2 \in adm_3(G)$ from $v_0, v_1 \in adm_3(G)$ and $cl(v_0, v_1) \subseteq cl(adm_3(G)) = adm_3(G)$, we get a contradiction to $v_0 \in \min(adm_3(G), \preceq')$. Hence $v_0 \preceq'_F v_1$. Now consider an arbitrary $v_3 \in \mathcal{V} \setminus \bigcup_{0 \leq i < j} V_i$ such that $v_1 \preceq^t v_3$. From $v_0 \preceq' v_1 \preceq^t v_3$ we get $v_0 \preceq^t v_3$. But since $v_0 \in \max(\mathcal{V} \setminus \bigcup_{0 \leq i < j} V_i, \preceq^t)$ it must also hold that $v_3 \preceq^t v_0$, meaning, together with $v_0 \preceq' v_1$, that $v_3 \preceq^t v_1$. As v_3 was chosen arbitrarily we have that $v_1 \in \max(\mathcal{V} \setminus \bigcup_{0 \leq i < j} V_i, \preceq^t)$, i.e. $v_1 \in V_j$, a contradiction to $v_1 \in V_k$ and $j < k$. $\square$

The fact that $\preceq$ indeed simulates $\star$ is now obtained from Lemmas 27 and 29: we get that $prf_3(F \star G) = \min(adm_3(G), \preceq)$ for each ADF G. This also makes $\preceq$ prf_3-adm_3-compliant. To show that $\preceq$ is faithful for F observe that, by (H2), it holds that $prf_3(F \star \rho^{\text{ADF}}_{adm_3}(\mathcal{V})) = prf_3(F)$ (note that for the set of all interpretations, $cl(\mathcal{V}) = \mathcal{V}$), hence $prf_3(F) = \min(\mathcal{V}, \preceq)$, meaning that (i) $v_1 \approx v_2$ for $v_1, v_2 \in prf_3(F)$ and (ii) $v_1 \prec v_2$ for $v_1 \in prf_3(F)$ and $v_2 \notin prf_3(F)$. Finally, for ADFs F_1 and F_2 with $prf(F_1) = prf(F_2)$

$$\begin{array}{c}\mathbf{tt},\mathbf{ff}\\ \uparrow\\ \mathbf{tu},\mathbf{ut},\mathbf{uf},\mathbf{fu}\\ \uparrow\\ \mathbf{tf},\mathbf{ft},\mathbf{uu}\end{array}$$

Figure 4.7: Faithful ranking $\preceq_F^{prf_3}$ for AF F used in Example 52.

we get, by definition of $\preceq'$, $\preceq^t$, $\preceq$ and (H4), that $\preceq_{F_1}=\preceq_{F_2}$, hence the assignment is faithful. This concludes the proof. $\square$

We have established a correspondence between operators induced by *prf*$_3$-*adm*$_3$-compliant rankings and operators satisfying the postulates (H1)–(H6) and (HAcyc). Of course, the condition of *prf*$_3$-*adm*$_3$-compliance depends on the concrete capabilities in terms of realizability of σ_3 and τ_3. Fortunately, we can capture *prf*$_3$-*adm*$_3$-compliance with conditions on the ranking.

Proposition 35. *A preorder $\preceq$ is prf$_3$-adm$_3$-compliant if and only if*

$$\forall v_1, v_2 \in \mathcal{V} : v_1 \neq v_2 \wedge v_1, v_2 \textit{ compatible} \wedge v_1 \approx v_2 \textit{ implies } \exists v_3 \in cl(v_1, v_2) : v_3 \prec v_1, v_2.$$

Proof. For the if-direction assume that for all $v_1, v_2 \in \mathcal{V}$ the implication holds. Consider an arbitrary ADF G and assume that $\min(adm_3(G), \preceq) \notin \Sigma_{\text{ADF}}^{prf_3}$. Hence $\min(adm_3(G), \preceq)$ is not incompatible, i.e. there exist $v_1, v_2 \in \min(adm_3(G), \preceq)$ ($v_1 \neq v_2$) which are compatible. Moreover, since both are minimal wrt. $\preceq$, it holds that $v_1 \approx v_2$. Hence we get that $\exists v_3 \in cl(v_1, v_2) : v_3 \prec v_1, v_2$. By monotonicity of cl (cf. Lemma 25.2) it follows that also $v_3 \in adm_3(G)$, a contradiction to $v_1 \in \min(adm_3(G), \preceq)$.

For the only-if-direction assume that $\preceq$ is *prf*$_3$-*adm*$_3$-compliant. Let $v_1, v_2 \in \mathcal{V}$ such that $v_1 \neq v_2$, v_1 and v_2 are compatible and $v_1 \approx v_2$. Further consider the ADF G such that $adm(G) = cl(\{v_1, v_2\})$ (note that such an ADF exists since cl is idempotent by Lemma 25.1). Since $\min(adm(G), \preceq) \in \Sigma_{\text{ADF}}^{prf_3}$ but for any $V \subseteq \{v_1, v_2\}$, $V \notin \Sigma_{\text{ADF}}^{prf_3}$ (recall that v_1 and v_2 are compatible), there must be some $v_3 \in adm(G) = cl(\{v_1, v_2\})$ with $v_3 \prec v_1, v_2$, which was to show. $\square$

With the insights from Theorems 36 and 37 we obtain concrete operators from faithful and *prf*$_3$-*adm*$_3$-compliant rankings. For instance, a valid operator is induced from the ranking $\preceq_F$ where *prf*$_3(F)$ are the minimal elements and all other interpretations form a $\prec_F$-chain. The three-valued version of Dalal's operator (cf. Definition 86) is not directly applicable here, as $\preceq_F^{prf_3}$ does not yield a *prf*$_3$-*adm*$_3$-compliant ranking for every ADF, as we will see in the following example.

Example 52. Consider the ADF $F = \{\langle a, a \wedge b\rangle, \langle b, a \wedge b\rangle\}$ and observe that $prf_3(F) = \{\mathbf{tt}, \mathbf{ff}\}$. It yields the ranking $\preceq_F^{prf_3}$ depicted in Figure 4.7. Now consider the compatible interpretations $\mathbf{tu}$ and $\mathbf{uf}$ and observe that all $v \in cl(\{\mathbf{tu}, \mathbf{uf}\}) = \{\mathbf{uu}, \mathbf{tu}, \mathbf{uf}, \mathbf{tf}\}$ have $v \not\prec_F^{prf_3} \mathbf{tu}$ and $v \not\prec_F^{prf_3} \mathbf{uf}$. Therefore, according to Proposition 35, $\preceq_F^{prf_3}$ is not prf_3-adm_3-compliant. In practice, this means that $F *_{prf_3}^{D} G$, where $adm_3(G) = \{\mathbf{uu}, \mathbf{tu}, \mathbf{uf}, \mathbf{tf}\}$ (e.g. $G = \{\langle a, \top\rangle, \langle b, \bot\rangle\}$), would yield $\rho_{prf_3}^{\mathrm{ADF}}(\{\mathbf{tu}, \mathbf{uf}\})$; but as $\{\mathbf{tu}, \mathbf{uf}\}$ is not realizable under prf_3 we do net get the preferred interpretations prescribed by the postulates. $\Diamond$

A refinement of the distance measure in order to result in prf_3-adm_3-compliant rankings is subject to future work.

4.4 Complexity

In this section we study the complexity of Dalal's operator for revision of AFs. We will consider the following decision problem for semantics $\sigma \in \{stb, prf, sem, stg\}$:

GIVEN:	the original AF F, the revising AF G, and a set of arguments E,
DECIDE:	whether E is a σ-extension of the revision of F by G, i.e. $E \in \sigma(F *_{\sigma}^{D} G)$?

The problem we are interested in is closely related to model checking in propositional logic revision, i.e. given a set of atoms E and formulas ϕ and ψ, deciding whether $E \in \mathsf{Mod}(\phi \circ^{D} \psi)$. The complexity of that problem was studied by Liberatore and Schaerf [144] and shown to be Θ_2^{P}-complete. We will study the considered problem for the proper I-maximal semantics stable, preferred, semi-stable and stage. We will show that the complexity for revision under stable semantics has the same complexity as in the case of propositional logic, while for revision under preferred (resp. semi-stable) semantics it lies one level up in the polynomial hierarchy, i.e. it is Θ_3^{P}-complete. For stage semantics we will show membership in this class.

We begin with the complexity of Dalal's operator for revision by AFs under stable semantics. We will make use of the following construction, which is adapted from reductions used in complexity proofs by Dimopoulos and Torres [92] and Dunne and Bench-Capon [97]. Moreover, for a set of arguments $X = \{x_1, \dots, x_n\}$ we denote by $\overline{X}$ the set of arguments $\{\overline{x_1}, \dots, \overline{x_n}\}$.

Definition 88. Given a propositional formula $\varphi(X) = \bigwedge_{c \in C} c$ in CNF, we define $F_\varphi = (A_\varphi, R_\varphi)$ as:

$$\begin{aligned}
A_\varphi =& X \cup \overline{X} \cup C \cup \{\varphi, \overline{\varphi}\},\\
R_\varphi =& \{(x, \overline{x}), (\overline{x}, x) \mid x \in X\} \cup \{(c, c') \mid c, c' \in C, c \neq c'\} \cup\\
& \{(x, c) \mid x \text{ occurs in } c\} \cup \{(\overline{x}, c) \mid \neg x \text{ occurs in } c\} \cup\\
& \{(c, \varphi) \mid c \in C\} \cup \{(\varphi, \overline{\varphi})\}.
\end{aligned}$$

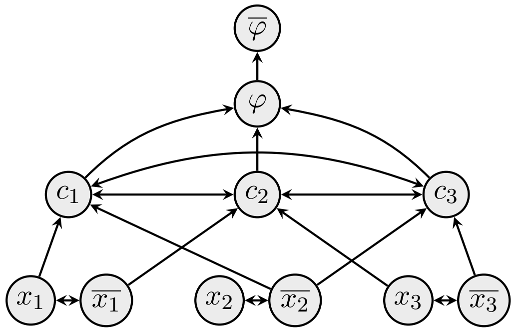


Figure 4.8: AF F_φ for $\varphi(X) = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_3)$.

Figure 4.8 depicts F_φ for an exemplary CNF formula $\varphi(X)$.

Lemma 30. *Given a propositional formula $\varphi(X) = \bigwedge_{c \in C} c$ in CNF, it holds that:*

1. *φ is satisfiable if and only if there exists an $E \in stb(F_\varphi)$ such that $\overline{\varphi} \notin E$;*
2. *for each $E, E' \in stb(F_\varphi)$ such that $\overline{\varphi} \notin E$ and $\overline{\varphi} \in E'$, it holds that $|E| + 1 = |E'|$;*
3. *for each $E \in stb(F_\varphi)$ such that $\overline{\varphi} \notin E$ and each $E' \in stb(F_\varphi - (C \cup \{\overline{\varphi}\}))$ it holds that $|E| = |E'|$.*

Proof. We begin with the observation that every stable extension of F_φ as well as every stable extension of $F_\varphi - (C \cup \{\overline{\varphi}\})$ contains $S \cup (\overline{X \setminus S})$ for some $S \subseteq X$, since each argument $x \in X$ is in symmetric conflict with $\overline{x}$ and neither of them receives any further attacks.

1. $(\Rightarrow)$: Assume φ is satisfiable, then there is some $S \subseteq X$ such that for each $c \in C$, it holds that $S \models c$. Hence, by construction of F_φ, $S \cup (\overline{X \setminus S})$ attacks all $c \in C$. Thus $S \cup (\overline{X \setminus S}) \cup \{\varphi\} \in stb(F_\varphi)$. $(\Leftarrow)$: Let $E \in stb(F_\varphi)$ with $\overline{\varphi} \notin E$. Moreover let $S \subseteq X$ for which $S \cup (\overline{X \setminus S}) \subseteq E$ (recall from before that such an S must exist). Since φ is the only attacker of $\overline{\varphi}$ it follows that $\varphi \in E$ and further $c \notin E$ for all $c \in C$. Therefore $S \cup (\overline{X \setminus S})$ must attack each $c \in C$, meaning by construction of F_φ that $S \models c$ for each $c \in C$, hence $S \models \varphi$; that is, φ is satisfiable.
2. From the $(\Leftarrow)$-direction of (1) we get that each $E \in stb(F_\varphi)$ with $\overline{\varphi} \notin E$ has $|E| = |X| + 1$. For an arbitrary $E' \in stb(F_\varphi)$ with $\overline{\varphi} \in E'$ it must hold that $\varphi \notin E'$, hence for at least one $c \in C$ we must have $c \in E'$. Since, as we know, $S \cup (\overline{X \setminus S}) \subseteq E'$ for some $S \subseteq X$, and by C forming a clique, $c \in E$ for at most one $c \in C$, it follows that $|E'| = |X| + 2$, that is $|E| + 1 = |E'|$.
3. Each $E' \in stb(F_\varphi - (C \cup \{\overline{\varphi}\}))$ is of the form $E' = S \cup (\overline{X \setminus S}) \cup \{\varphi\}$ for some $S \subseteq X$. Therefore $|E'| = |X| + 1$. Hence, from the observation in (2), the result follows.

□

Given these observations we can show the exact complexity of Dalal's operator for revision under stable semantics.

Theorem 38. *Given AFs $F, G \in AF_{\mathfrak{A}}$ and $E \subseteq \mathfrak{A}$, deciding whether $E \in stb(F *^D_{stb} G)$ is Θ_2^{P}-complete.*

Proof. For membership in Θ_2^{P} we sketch an algorithm that decides $E \in stb(F *^D_{stb} G)$ in polynomial time with $\mathcal{O}(\log m)$ calls to an NP oracle, where $m = |A_F| + |A_G|$. First we check whether $E \in stb(G)$ (in P); if no, then we return with a negative answer. Then the minimal distance $z = \min\{\triangle_H(S,T) \mid S \in stb(F), T \in stb(G)\}$ is determined. It holds that $z \leq m$, since $S \subseteq A_F$ (resp. $T \subseteq A_G$) for each $S \in stb(F)$ (resp. $T \in stb(G)$). Now z can be computed by binary search among $\{0, \dots, m\}$ with $\mathcal{O}(\log m)$ calls to the following NP procedure: guess $S \subseteq A_F$, $T \subseteq A_G$ and check, in polynomial time, whether $S \in stb(F)$, $T \in stb(G)$ and $\triangle_H(S,T) < z$. Having obtained z, we finally call another NP oracle to check whether there is an $S \in stb(F)$ such that $\triangle_H(S,E) = z$; if such an S does exist, $E \in stb(F *^D_{stb} G)$, otherwise $E \notin stb(F *^D_{stb} G)$.

To show Θ_2^{P} hardness we give a polynomial-time reduction from the following Θ_2^{P}-complete problem (recall that a 1-existential QBF being false is equivalent to a propositional formula being unsatisfiable):

Given:	propositional formulas $\varphi_1(X_1), \dots, \varphi_m(X_m)$ such that φ_i unsatisfiable implies φ_{i+1} unsatisfiable, for $1 \leq i < m$,
Decide:	whether $k = \max\{1 \leq i \leq m \mid \varphi_i \text{ is satisfiable}\}$ is odd.

Without loss of generality we can assume that:

(i) $X_i \cap X_j = \emptyset$ for all $1 \leq i, j \leq m$, $i \neq j$;

(ii) a fixed $n = |X_i| = |X_j|$ for all $1 \leq i, j \leq m$;

(iii) each φ_i is in CNF with C_i denoting the set of clauses of φ_i; and

(iv) m is odd.

Now, given an instance of this problem, define $F = \bigcup_{1 \leq i \leq m} F_{\varphi_i} \cup F_i$ where F_{φ_i} is given by Definition 88 and:

$$
\begin{aligned}
F_i &= (\{\overline{\varphi}_i, \overline{\varphi}_{i+1}\} \cup C_i, \{(\overline{\varphi}_{i+1}, \overline{\varphi}_i)\} \cup \{(\overline{\varphi}_{i+1}, c) \mid c \in C_i\}) && \text{for } 1 \leq i < m \\
F_m &= (\{\overline{\varphi}_m, x, x'\} \cup C_m, \{(x,x'), (x',x), (x', \overline{\varphi}_m)\} \cup \{(x', c) \mid c \in C_m\})\,.
\end{aligned}
$$

Intuitively, F contains the frameworks F_{φ_i} constructed according to Definition 88 together with "connecting frameworks" F_i which make $\overline{\varphi}_{i+1}$ attack $\overline{\varphi}_i$ and all "clause-arguments"

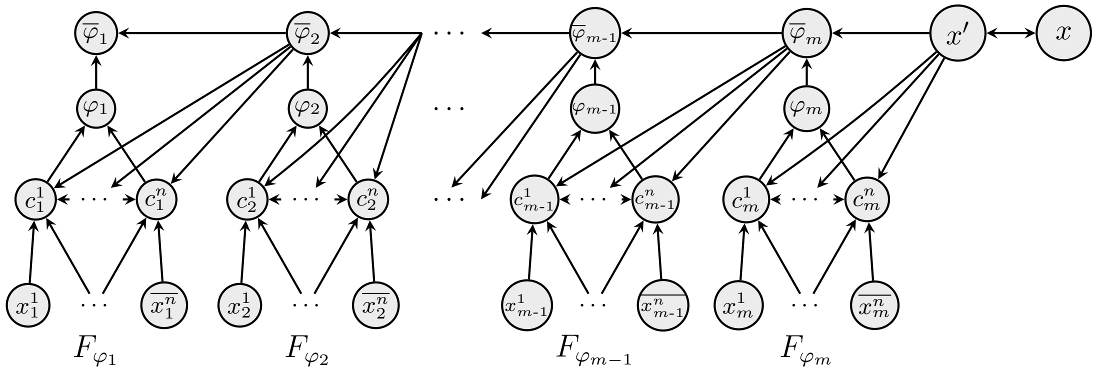


Figure 4.9: Illustration of the AF obtained from the reduction in the proof of Theorem 38.

C_i. F_m can be seen as the "starting framework", that has additional arguments x and x' (in symmetric attack) and x' adopting the role of attacking the "clause-arguments". A schematic illustration of F can be seen in Figure 4.9. Moreover, we define $G = (\{x, x'\}, \{(x, x'), (x', x)\})$ and $E = \{x\}$. In the following we show that $E \in stb(F *^D_{stb} G)$ if and only if k is odd.

Due to the splitting property [19] (we will see this in more detail in Section 5.1, Theorem 43), the stable extensions of F are composed by the union of stable extensions of its components, where the computation of $stb(F_{\varphi_i})$ has to take into account $stb(F_{\varphi_{i+1}})$. That is, $stb(F) = \{\{\alpha\} \cup \bigcup_{1 \leq i \leq m} E_i \mid \alpha \in \{x, x'\}, E_i \in stb(F'_{\varphi_i})\}$ where

- $F'_{\varphi_m} = F_{\varphi_m}$ if $\alpha = x$ and $F'_{\varphi_m} = F_{\varphi_m} - (C_m \cup \{\overline{\varphi}_m\})$ if $\alpha = x'$, and
- $F'_{\varphi_i} = F_{\varphi_i}$ if $\overline{\varphi}_{i+1} \notin E_{i+1}$ and $F'_{\varphi_i} = F_{\varphi_i} - (C_i \cup \{\overline{\varphi}_i\})$ if $\overline{\varphi}_{i+1} \in E_{i+1}$ for $1 \leq i < m$.

In words, F'_{φ_i} is just the AF F_{φ_i}, but without the arguments being attacked by accepted arguments $F'_{\varphi_{i+1}}$ (given by E_{i+1}). That is, if $\overline{\varphi}_{i+1} \notin E_{i+1}$ then $F'_{\varphi_i} = F_{\varphi_i}$ and otherwise F'_{φ_i} is F_{φ_i} without $C_i \cup \{\overline{\varphi}_i\}$.

Now recall that k is the highest index such that φ_k is satisfiable. Consider an i with $k < i \leq m$. If $F'_{\varphi_i} = F_{\varphi_i}$ then we know, by Lemma 30.1 and φ_i being unsatisfiable, that $\overline{\varphi}_i \in E_i$, hence $F'_{\varphi_{i-1}} = F_{\varphi_{i-1}} - (C_{i-1} \cup \{\overline{\varphi}_{i-1}\})$. On the other hand if $F'_{\varphi_i} = F_{\varphi_i} - (C_i \cup \{\overline{\varphi}_i\})$ then obviously $\overline{\varphi}_i \notin E_i$, hence $F'_{\varphi_{i-1}} = F_{\varphi_{i-1}}$. Now consider an i with $1 \leq i \leq k$. Again from Lemma 30.1 and φ_i being satisfiable, we get that there is some $E \in stb(F_{\varphi_i})$ with $\overline{\varphi}_i \notin E$. Therefore, by Lemma 30.2 and 30.3, for $\alpha \in \{x, x'\}$ the extension $S^*_\alpha = \{\alpha\} \cup \bigcup_{1 \leq i \leq m} E_i$ with $\overline{\varphi}_i \notin E_i$ for $1 \leq i \leq k$ is the one with the minimal distance to $\{\alpha\}$ among all elements of $stb(F)$ (recall the assumption that $|X_i| = |X_j|$ for all $1 \leq i, j \leq m$). Now if k is odd, we get, by the assumption (iii) that m is odd, that $m - k$ is even. Hence $\triangle_H(S^*_x, \{x\}) = \triangle_H(S^*_{x'}, \{x'\})$ and furthermore $stb(F *^D_{stb} G) = \{\{x\}, \{x'\}\}$, that is $E \in stb(F *^D_{stb} G)$. If, on the other hand, k is even, then $m - k$ is odd and, by Lemma 30.2 and 30.3, $d_H(S^*_x, \{x\}) = d_H(S^*_{x'}, \{x'\}) + 1$, hence $E \notin stb(F *^D_{stb} G) = \{\{x'\}\}$.

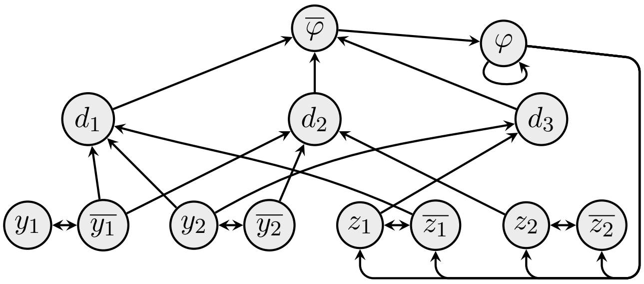


Figure 4.10: AF F_Φ for the QBF $\Phi = \exists y_1, y_2 \forall z_1, z_2 : (y_1 \wedge \neg y_2 \wedge z_1) \vee (y_1 \wedge y_2 \wedge \neg z_2) \vee (\neg y_2 \wedge \neg z_1)$.

□

The problem of verification (checking whether a given set of arguments is an extension) is coNP-complete for preferred semantics, while it is decidable in polynomial time (and even in logarithmic space) for stable semantics (cf. Table 2.1). As it turns out, this increase in complexity also carries over to revision under preferred semantics. To show this, we will make use of the following construction.

Definition 89. Given a 2-existential QBF $\Phi = \exists Y \forall Z \varphi(Y, Z)$ where φ is a DNF $\bigvee_{d \in D} d$ with each d a conjunction of literals from $X = Y \cup Z$, we define $F_\Phi = (A_\Phi, R_\Phi)$ as:

$$\begin{aligned} A_\Phi =& X \cup \overline{X} \cup D \cup \{\varphi, \overline{\varphi}\}, \\ R_\Phi =& \{(x, \overline{x}), (\overline{x}, x) \mid x \in X\} \cup \\ & \{(\overline{x}, d) \mid x \text{ occurs in } d\} \cup \{(x, d) \mid \neg x \text{ occurs in } d\} \cup \\ & \{(d, \overline{\varphi}) \mid d \in D\} \cup \{(\overline{\varphi}, \varphi), (\varphi, \varphi)\} \cup \{(\varphi, z) \mid z \in Z\}. \end{aligned}$$

The construction is illustrated on an exemplary 2-existential QBF Φ in Figure 4.10. Note that the attacks from arguments in $X \cup \overline{X}$ to arguments in D differ from Definition 88. Now the idea is that an argument $d \in D$ needs all arguments occurring as literals in d to be defended. The following lemma states this more formally.

Lemma 31. *Let $\Phi = \exists Y \forall Z \varphi(Y, Z)$ where φ is a DNF $\bigvee_{d \in D} d$. For each $d \in D$, $S \subseteq Y$ and $T \subseteq Z$ it holds that:*

- *$S \cup T \models d$ if and only if d is defended by $S \cup (\overline{Y \setminus S}) \cup T \cup (\overline{Z \setminus T})$;*
- *$S \cup T \not\models d$ if and only if d is attacked by $S \cup (\overline{Y \setminus S}) \cup T \cup (\overline{Z \setminus T})$.*

Proof. If $S \cup T \models d$, then the set of arguments attacking d is, according to Definition 89, contained in $\overline{S} \cup (Y \setminus S) \cup \overline{T} \cup (Z \setminus T)$. Therefore, it is not attacked and even defended by $S \cup (\overline{Y \setminus S}) \cup T \cup (\overline{Z \setminus T})$.

If $S \cup T \not\models d$, then there is some argument attacking d which is not contained in $\overline{S} \cup (\overline{Y \setminus S}) \cup \overline{T} \cup (\overline{Z \setminus T})$. Therefore, it is attacked and, consequently, not defended by $S \cup (\overline{Y \setminus S}) \cup T \cup (\overline{Z \setminus T})$. □

The following lemma shows similar properties as Lemma 30.

Lemma 32. *Consider the 2-existential QBF $\Phi = \exists Y \forall Z \varphi(Y, Z)$ where φ is a DNF $\bigvee_{d \in D} d$. It holds that:*

1. *Φ is true if and only if there exists $E \in prf(F_\Phi)$ such that $\overline{\varphi} \notin E$;*
2. *for each $E \in prf(F_\Phi)$ it holds that (a) $|E| = |Y| + |Z| + 1$ if $\overline{\varphi} \in E$ and (b) $|E| = |Y|$ if $\overline{\varphi} \notin E$;*
3. *for each $E \in prf(F_\Phi - \{\overline{\varphi}\})$ is holds that $|E| = |Y|$.*

Proof. 1. ($\Rightarrow$): Assume Φ is true. That is, there is some $S \subseteq Y$ such that for all $T \subseteq Z$ it holds that $\varphi(S, T)$ is true. We show that $E = S \cup (\overline{Y \setminus S}) \in prf(F_\Phi)$. First, E is easily checked to be admissible. Towards a contradiction, assume there is some $E' \in adm(F_\Phi)$ with $E' \supset E$. Further assume there is some $d \in D$ included in $E' \setminus E$. Due to the non-triviality of d there is at least one $z \in Z \cup \overline{Z}$ attacking d and, consequently, it must hold that $\overline{z} \in E'$. Then, due to φ attacking all $Z \cup \overline{Z}$, it must hold that $\overline{\varphi} \in E'$, a contradiction to conflict-freeness of E' since also $d \in D$. Knowing that $d \notin E'$ for all $d \in D$, assume that $\overline{\varphi} \in E'$. To this end $\overline{\varphi}$ has to be defended by E' from each $d \in D$. This means that there must be some $T \subseteq Z$ such that $T \cup (\overline{Z \setminus T}) \subseteq E'$ and each $d \in D$ is attacked by $S \cup (\overline{Y \setminus S}) \cup T \cup (\overline{Z \setminus T})$. But then, by Lemma 31, $S \cup T \not\models d$ for each $d \in D$, a contradiction to $\varphi(S, T)$ being true.

($\Leftarrow$): We show the contrapositive, that is, if Φ is false then all $E \in prf(F_\Phi)$ have $\overline{\varphi} \in E$. Observe that for any $S \subseteq Y$, $S \cup (\overline{Y \setminus S})$ is admissible in F_Φ, hence $S \cup (\overline{Y \setminus S})$ is contained in some preferred extension. Moreover, each preferred extension must contain $S \cup (\overline{Y \setminus S})$ for some $S \subseteq Y$. Consider an arbitrary $S \subseteq Y$. As, by assumption, Φ is false, there must be some $T \subseteq Z$ such that $\varphi(S, T)$ is false. Hence for every $d \in D$ it must hold that $S \cup T \not\models d$ and consequently, by Lemma 31, d is attacked by $X_S = S \cup (\overline{Y \setminus S}) \cup T \cup (\overline{Z \setminus T})$. Hence $X_S \cup \{\overline{\varphi}\}$ is admissible and, by attacking all other arguments, also preferred in F_Φ. Now assume there is an $E' \in prf(F_\Phi)$ with $S \subseteq E'$ and $\overline{\varphi} \notin E'$. By the latter no argument among $Z \cup \overline{Z}$ can be in E' as it cannot be defended from φ. Hence, to be incomparable to all the preferred extensions which do include $\overline{\varphi}$, E' must include some $d \in D$. But also this in not possible as by assumption there must be some $T \subseteq Z$ making $S \cup T \not\models d$, meaning, by Lemma 31, that d is attacked by $S \cup (\overline{Y \setminus S}) \cup T \cup (\overline{Z \setminus T})$. If it is attacked by $S \cup (\overline{Y \setminus S})$ then E' is not conflict-free; if it is attacked by $T \cup (\overline{Z \setminus T})$ then E' is not admissible. We conclude that all $E \in prf(F_\Phi)$ have $\overline{\varphi} \in E$.

2. Consider some $E \in prf(F_\Phi)$. (a) If $\overline{\varphi} \in E$ then $d \notin E$ for all $d \in D$, hence a maximal conflict-free selection of arguments among $Y \cup \overline{Y} \cup Z \cup \overline{Z}$ must be included in E, therefore $S \cup (\overline{Y \setminus S}) \cup T \cup (\overline{Z \setminus T}) \subseteq E$ for some $S \subseteq Y$ and $T \subseteq Z$. Hence $|E| = |Y| + |Z| + 1$. (b) If $\overline{\varphi} \notin E$ then no argument among $Z \cup \overline{Z}$ can be defended by E. Moreover, as φ does not contain monoms which are trivial for Y, it follows by Lemma 31 that no $d \in D$ can be defended by E. On the other hand, E must include a maximal conflict-free selection of arguments among $Y \cup \overline{Y}$, hence $|E| = |Y|$.

3. Let $F'_\Phi = F_\Phi - \{\overline{\varphi}\}$ and observe that the self-attacking argument φ is unattacked in F'_Φ. Hence none of the arguments $Z \cup \overline{Z}$ can be defended. Moreover, as φ does not contain monoms which are trivial for Y, each argument $d \in D$ is attacked by $Z \cup \overline{Z}$ and can therefore also not be defended. It follows that the preferred extensions of F'_Φ are given by $S \cup (\overline{Y \setminus S})$ for each $S \subseteq Y$, each containing $|Y|$ arguments.

□

We are now ready to show Θ_3^{P}-completeness of the considered problem under preferred semantics.

Theorem 39. *Given AFs $F, G \in AF_{\mathfrak{A}}$ and $E \subseteq \mathfrak{A}$, deciding whether $E \in prf(F *^D_{prf} G)$ is Θ_3^{P}-complete.*

Proof. To show membership in Θ_3^{P} we sketch an algorithm that decides $E \in prf(F *^D_{prf} G)$ in polynomial time with $\mathcal{O}(\log m)$ calls to a Σ_2^{P} oracle, where $m = |A_F| + |A_G|$. First, we check whether $E \in prf(G)$ (in coNP); if $E \notin prf(G)$ we return with a negative answer. Otherwise, we continue with determining the minimal distance $z = \min\{\triangle_H(S,T) \mid S \in prf(F), T \in prf(G)\}$. Since $S \subseteq A_F$ (resp. $T \subseteq A_G$) for each $S \in prf(F)$ (resp. $T \in prf(G)$), it holds that $d \leq m$. Therefore it can be computed by binary search among $\{0, \dots, m\}$ with $\mathcal{O}(\log m)$ oracle calls to the following Σ_2^{P} procedure: Guess $S \subseteq A_F$, $T \subseteq A_G$ and check (in coNP) whether $S \in prf(F)$, $T \in prf(G)$ and $\triangle_H(S,T) < z$. Having obtained z, we finally call the Σ_2^{P}-oracle once again to check whether there is an $S \in prf(F)$ with $\triangle_H(S,E) = z$. If such an S does exist then $E \in prf(F *^D_{prf} G)$, otherwise $E \notin prf(F *^D_{prf} G)$.

To show Θ_3^{P}-hardness we give a polynomial-time reduction from the following Θ_3^{P}-complete problem:

Given: 2-existential QBFs $\Phi_1, \dots, \Phi_m$ such that
Φ_i being false implies Φ_{i+1} being false, for $1 \leq i < m$,
Decide: whether $k = \max\{1 \leq i \leq m \mid \Phi_i \text{ is true}\}$ is odd.

We use the following notation to identify the elements of QBFs: $\Phi_i = \exists Y_i \forall Z_i \varphi_i$ with φ_i being a propositional formula in DNF over atoms $X_i = Y_i + Z_i$. Without loss of generality we can assume that

(i) $X_i \cap X_j = \emptyset$ for all $1 \leq i, j \leq m$, $i \neq j$;

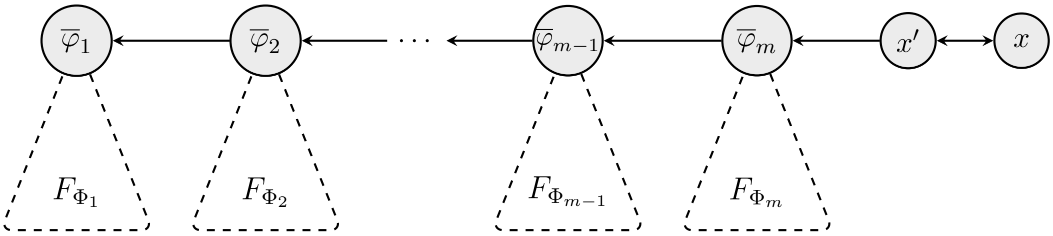


Figure 4.11: Illustration of the AF obtained from the reduction in the proof of Theorem 39.

(*ii*) $|Y_i| = |Y_j|$ and $|Z_i| = |Z_j|$ for all $1 \leq i, j \leq m$; and

(*iii*) m is odd.

Due to (*ii*) we will use $|Y|$ to denote $|Y_i|$ and $|Z|$ to denote $|Z_i|$ for any $1 \leq i \leq m$. Now for each $\Phi_i = \exists Y_i \forall Z_i \varphi_i$, let F_{Φ_i} be as given in Definition 89. We define $F = \bigcup_{1 \leq i \leq m} F_{\Phi_i} \cup F_i$ where:

$$\begin{aligned} F_i &= (\{\overline{\varphi}_i, \overline{\varphi}_{i+1}\}, \{(\overline{\varphi}_{i+1}, \overline{\varphi}_i)\}) && \text{for } 1 \leq i < m \\ F_m &= (\{\overline{\varphi}_m, x, x'\}, \{(x, x'), (x', x), (x', \overline{\varphi}_m)\})\,. \end{aligned}$$

Figure 4.11 depicts a schematic example of F. The subframeworks F_i can be regarded as "connecting frameworks", adding just an attack from $\overline{\varphi}_{i+1}$ to $\overline{\varphi}_i$. F_m is the "starting framework". Moreover, we define $G = (\{x, x'\}, \{(x, x'), (x', x)\})$ and $E = \{x\}$. We show that $E \in \mathit{prf}(F *^{D}_{\mathit{prf}} G)$ if and only if k is odd.

Due to the splitting property of preferred semantics [19] (again, Theorem 43 will give the details), the preferred extensions of F are composed as $\mathit{prf}(F) = \{\{\alpha\} \cup \bigcup_{1 \leq i \leq m} E_i \mid \alpha \in \{x, x'\}, E_i \in \mathit{prf}(F'_{\Phi_i})\}$, where:

- $F'_{\Phi_m} = F_{\Phi_m}$ if $\alpha = x$ and $F'_{\Phi_m} = (F_{\Phi_m} - \{\overline{\varphi}_m\})$ if $\alpha = x'$, and
- $F'_{\Phi_i} = F_{\Phi_i}$ if $\overline{\varphi}_{i+1} \notin E_{i+1}$ and $F'_{\Phi_i} = F_{\Phi - 1} - \{\overline{\varphi}_i\})$ if $\overline{\varphi}_{i+1} \in E_{i+1}$ for $1 \leq i < m$.

Recall that k is the highest index such that Φ_k is true. Due to Lemma 32 it holds that:

- $1 \leq i \leq k$: we have either $|E_i| = |Y|$ or $|E_i| = |Y| + |Z| + 1$;
- $k < i \leq m$: if $\alpha = x$ we have $|E_i| = |Y| + |Z| + 1$ for $i \in \{m, m-2, \dots\}$ and $|E_i| = |Y|$ for $i \in \{m-1, m-3, \dots\}$; otherwise we have $|E_i| = |Y|$ for $i \in \{m, m-2, \dots\}$ and $|E_i| = |Y| + |Z| + 1$ for $i \in \{m-1, m-3, \dots\}$.

Moreover, we get from Lemma 32 that each F_{Φ_i} with $1 \leq i \leq k$ has an extension $E^*_i \in \mathit{prf}(F_{\Phi_i})$ with $\overline{\varphi}_i \notin E^*_i$, hence $|E^*_i| = |Y|$. Let $S^*_\alpha \in \mathit{prf}(F)$ be now such that

$E_i = E_i^*$ for all $1 \leq i \leq k$. By the observations above and assumption (ii), S_α^* has minimal distance to $\{\alpha\}$ among all preferred extensions containing α, for $\alpha \in \{x, x'\}$.

If k is odd, we get, by the assumption (iii) that m is odd, that $m - k$ is even, hence $\triangle_H(S_\alpha^*, \{\alpha\}) = m|Y| + \frac{m-k}{2}(|Z| + 1) + 1$ for both $\alpha \in \{x, x'\}$. Therefore $prf(F *_{prf}^D G) = \{\{x\}, \{x'\}\}$, i.e. $E \in prf(F *_{prf}^D G)$.

If k is even, then $m - k$ is odd. We get $\triangle_H(S_{x'}^*, \{x'\}) = m|Y| + \lfloor\frac{m-k}{2}\rfloor(|Z| + 1) + 1 < m|Y| + \lceil\frac{m-k}{2}\rceil(|Z| + 1) + 1 = \triangle_H(S_x^*, \{x\})$, hence $E \notin prf(F *_{prf}^D G) = \{\{x'\}\}$. □

For the semi-stable semantics, we can use the fact that there is an exact and efficient translation from preferred to semi-stable semantics [109] to show that complexity from preferred semantics carries over.

Theorem 40. *Given AFs $F, G \in AF$ and $E \subseteq \mathfrak{A}$, deciding whether $E \in sem(F *_{sem}^D G)$ is Θ_3^P-complete.*

Proof. Membership in Θ_3^P is shown by the same algorithm as the one in Theorem 39, with checks wrt. semi-stable semantics instead of preferred semantics. As the verification problem is in coNP for both semi-stable and preferred semantics, the modified algorithm decides $E \in sem(F *_{sem}^D G)$ in polynomial time with $\mathcal{O}(\log m)$ calls to a Σ_2^P oracle, showing that the problem is in Θ_3^P.

Hardness is by reduction from the problem of, given AFs $F, G \in AF_\mathfrak{A}$ and $E \subseteq \mathfrak{A}$, deciding whether $E \in prf(F *_{sem}^D G)$, which was shown to be Θ_3^P-complete in Theorem 39. To this end we apply a translation from [109, Translation 1] to F and G to get, in polynomial time, $F' = (A_F \cup \{a' \mid a \in A_F\}, R_F \cup \{(a, a'), (a', a') \mid a \in A_F\})$ and $G' = (A_G \cup \{a' \mid a \in A_G\}, R_G \cup \{(a, a'), (a', a') \mid a \in A_G\})$. According to [109, Theorem 1] it holds that $sem(F') = prf(F)$ and $sem(G') = prf(G)$. It follows that $E \in sem(F' *_{sem}^D G')$ if and only if $E \in prf(F *_{prf}^D G)$, and the result follows. □

Finally, the problem of checking containment in the extensions of the result of Dalal revision under stage semantics is also in Θ_3^P, hardness of the problem has to be left subject to future work.

Theorem 41. *Given AFs $F, G \in AF$ and $E \subseteq \mathfrak{A}$, deciding whether $E \in stg(F *_{stg}^D G)$ is in Θ_3^P.*

Proof. Membership in Θ_3^P by the same algorithm as the one in Theorem 39, with checks wrt. stage semantics instead of preferred semantics, considering the fact that the verification problem for stage semantics is in coNP. □

Table 4.1 summarizes the complexity results presented in this section together with the result of [144] for revision of propositional formulas.

Table 4.1: Complexity of deciding, given $F, G \in AF_{\mathfrak{A}}$, $E \in \mathfrak{A}$, whether $E \in \sigma(F *_\sigma^D G)$.

σ	Mod	stb	prf	sem	stg
	Θ_2^{P}-c	Θ_2^{P}-c	Θ_3^{P}-c	Θ_3^{P}-c	in Θ_3^{P}

4.5 Discussion

In [89] also another variant of revision of AFs was studied. There, the AF is revised by a propositional formula, with the models of the formula expressing the information which should be incorporated. Hence the operators are of the form $\star : AF_{\mathfrak{A}} \times \mathsf{P}_{\mathfrak{A}} \mapsto AF_{\mathfrak{A}}$. This approach was preceded by the approach of Coste-Marquis et al. [76], with the crucial difference that there the outcome of the revision was a set of AFs. Requiring the result to be a single AF again leads to problems with limited expressiveness. We illustrate this on an example.

Example 53. Let σ be a semantics and F an AF such that $\sigma(F) = \{\{a, b, c\}\}$. Dalal's approach generates the following preorder $\preceq_F^\sigma$:

$$\{a, b, c\} \prec_F^\sigma \{a, b\} \approx_F^\sigma \{a, c\} \approx_F^\sigma \{b, c\} \prec_F^\sigma \{a\} \approx_F^\sigma \{b\} \approx_F^\sigma \{c\} \prec_F^\sigma \emptyset.$$

Now consider the formula $\varphi = \neg(a \wedge b \wedge c)$ by which we want to revise F. We obtain that $\min([\varphi], \preceq) = \{\{a, b\}, \{a, c\}, \{b, c\}\}$. Observe, now, that $\{\{a, b\}, \{a, c\}, \{b, c\}\}$ is not conflict-sensitive, and therefore not realizable under any of the semantics under consideration (cf. Theorem 1). Hence, we run into the same problems as in Example 43, as we have no means to express the desired result. $\Diamond$

The problem was solved in [89] by restricting the rankings to be σ-compliant, a similar notion as given in Definition 87, but requiring the minimal elements of every possible extension-set to be σ-realizable. The focus on I-maximal semantics circumvents this problem, as the next example illustrates.

Example 54. Consider again the AF F from Example 53, with $\sigma(F) = \{\{a, b, c\}\}$, for instance $F = (\{a, b, c\}, \emptyset)$. The corresponding ranking obtained with Hamming distance was problematic when revising by a propositional formula, because the desired outcome of a revision operator could turn out to be $\{\{a, b\}, \{b, c\}, \{a, c\}\}$, which usually is not σ-realizable. We cannot, however, run into this problem when revising by an AF G, since the outcome of revision will, by definition, be a proper subset of $\sigma(G)$, namely $\min(\sigma(G), \preceq_F^\sigma)$. Due to the proper I-maximality of σ, any proper subset of $\sigma(G)$ is also σ-realizable. It follows that Dalal's operator and, for the same reason, any other standard revision operator, can be applied in this setting. $\Diamond$

In this chapter we have given impossibility results for certain semantics (Theorems 29 and 32) saying there is no rational revision operator due to the limited expressiveness of the semantics. The following theorem states this in a very general manner.

Theorem 42. *Let $\mathcal{F}$ be a formalism and σ a semantics of that formalism such that there are some $\mathsf{kb}_1, \mathsf{kb}_2 \in \mathcal{F}$ with $\sigma(\mathsf{kb}_1) \cap \sigma(\mathsf{kb}_2) \notin \Sigma^{\sigma}_{\mathcal{F}}$. Then there exists no operator $* : \mathcal{F} \times \mathcal{F} \mapsto \mathcal{F}$ satisfying $(A2_{\sigma})$.*

Finally we want to emphasize that the hybrid approach presented in Section 4.3.3 is also of interest for other knowledge representation formalisms. Whenever there are different semantics for a formalism, it can be desirable to do revision by combining them in the way preferred and admissible semantics were combined for revision of ADFs. In particular, in case one semantics always delivers a subset of the results of the other semantics, a hybrid approach can be more meaningful than considering only one of the semantics in isolation. As an example we identify classical models and answer sets of logic programs, where each answer set is always a classical model. We envisage a generalization of the hybrid approach to arbitrary formalisms for future work.

CHAPTER 5

Splitting

This chapter deals with splitting in argumentation formalisms. That is the question whether, or under which conditions, a semantics of a formalism allows for incremental computation of its results. With incremental computation we basically mean the following: (i) splitting the knowledge base into two parts; (ii) computing the results of the first part; (iii) propagating these results to the second part; (iv) computing the results of the second part; (v) putting the sub-results together to obtain the results for the overall knowledge base.

Splitting in knowledge representation formalisms was proposed and first studied by Lifschitz and Turner [145] in the context of logic programs. It has further been investigated for answer set programming [115, 139], default logic [194] and, most recently, abstract argumentation [19, 28, 29, 18].

As pointed out in the introduction, splitting is interesting both from a theoretical and practical point of view. On the one hand it gives insights on whether local evaluation of a semantics is possible, on the other hand splitting techniques can be useful for solving since it can boost the performance of the evaluation by dividing one large task into several smaller tasks.

The rest of the chapter is organized as follows. We will begin, in Section 5.1, by presenting the basic concepts of splitting for the case of AFs, mostly reviewing work of Baumann [19] and Baumann et al. [29]. Then we will show results for splitting of ADFs in Section 5.2. There we will first, in Section 5.2.1, deal with directional splitting, that is splitting along the lines of strongly connected components, and present splitting techniques for all ADF semantics we consider in this work. In Section 5.2.2 we will then study general splitting. The main results of this chapter have appeared in [147].

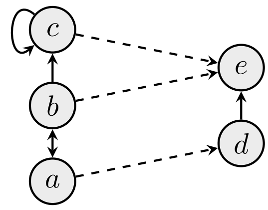


Figure 5.1: AF F and a possible splitting thereof.

5.1 Splitting of AFs

In this section we review the main results on splitting of AFs from [19] and [29]. We will be accurate in presentation, but use slightly different notation at times.

A splitting of an AF as defined in [19] is a partition into two AFs with disjoint sets of arguments, where the remaining attacks are restricted to a single direction. In other words, a splitting must not divide any strongly connected component of the AF.

Definition 90. Let F_1 and F_2 be AFs such that $A_{F_1} \cap A_{F_2} = \emptyset$. Moreover let $R_3 \subseteq A_{F_1} \times A_{F_2}$. The tuple (F_1, F_2, R_3) is called a *splitting* of the AF $F = (A_{F_1} \cup A_{F_2}, R_{F_1} \cup R_{F_2} \cup R_3)$.

Example 55. Consider the AF F depicted in Figure 5.1, also including the dashed edges. First observe that $prf(F) = \{\{a\}, \{b, d\}\}$. Let $F_1 = (\{a, b, c\}, \{(a, b), (b, a), (b, c), (c, c)\})$ and $F_2 = (\{d, e\}, \{(d, e)\})$. It holds that $(F_1, F_2, \{(a, d), (b, e), (c, e)\})$ is a splitting of F. (Note that there are also several other splittings of F.) The splitting is illustrated by the dashed attacks. Further observe that $prf(F_1) = \{\{a\}, \{b\}\}$ and $prf(F_2) = \{\{d\}\}$. As we cannot just put these sets together, $prf(F) \neq prf(F_1) \times prf(F_1) = \{\{a, d\}, \{b, d\}\}$, one has to find accurate ways of propagating the extension of the first sub-AF along the links of the splitting. $\Diamond$

The following definition of a reduct takes care of this propagation.

Definition 91. Let F be an AF, A' a set disjoint from A_F, $I, U \subseteq A'$, and $R \subseteq A' \times A_F$. The (I, U, R)-*reduct* of F is defined as $F^{I,U,R} = (A^{I,U,R}, R^{I,U,R})$ with

$$\begin{aligned} A^{I,U,R} =& \{a \in A_F \mid \nexists b \in I : (b, a) \in R\}, \text{ and} \\ R^{I,U,R} =& \{(a, b) \in R_F \mid a, b \in A^{I,U,R}\} \cup \\ & \{(a, a) \mid a \in A^{I,U,R}, \exists b \in U : (b, a) \in R\}. \end{aligned}$$

The intuitive reading of Definition 91 is that I and U represent the partial evaluation of the partial AF preceding F, and R contains the attacks connecting the preceding AF with F. While I can be regarded as the arguments in the extension, U contains arguments which are neither in the extension nor attacked by the extension. Arguments in F which are, via R, attacked by I now get removed in the (I, U, R)-reduct and arguments which are, via R, attacked by U, become self-attacking in $F^{I,U,R}$.

The splitting theorem is now formulated as follows.

Theorem 43 ([19]). *Let $\sigma \in \{stb, adm, prf, com, grd\}$. Further let F be an AF and (F_1, F_2, R_3) be a splitting of F. The following holds:*

1. *If $E_1 \in \sigma(F_1)$ and $E_2 \in \sigma(F_2^{E_1,(A_{F_1} \setminus (E_1)^+_{F_1}),R_3})$ then $(E_1 \cup E_2) \in \sigma(F)$.*

2. *If $E \in \sigma(F)$ then $(E \cap A_1) \in \sigma(F_1)$ and $(E \cap A_2) \in \sigma(F_2^{E_1,(A_{F_1} \setminus (E_1)^+_{F_1}),R_3})$.*

With this theorem at hand, we can now incrementally compute the extensions of an AF, given that it is not composed of only a single strongly connected component.

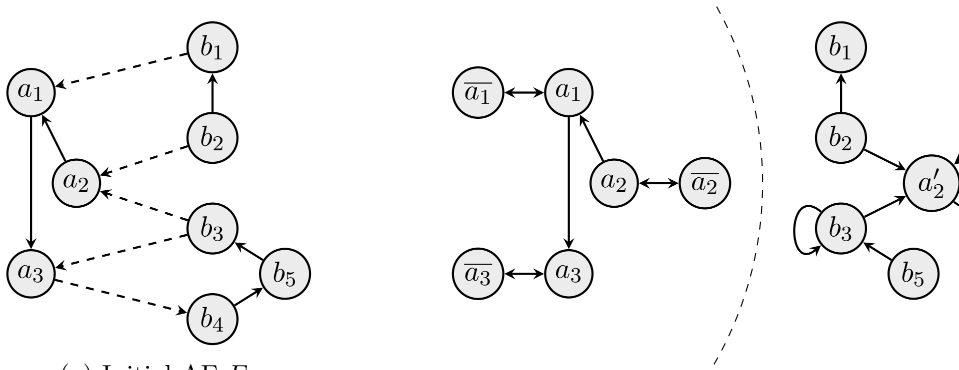
Example 56. Again consider the AF F depicted in Figure 5.1. We illustrate the evaluation F under the preferred semantics applying Theorem 43 with the splitting $(F_1, F_2, \{(a,d),(b,e),(c,e)\})$ as identified in Example 55. We begin with F_1 and get *prf*$(F_1) = \{\{a\},\{b\}\}$. We first consider $\{a\}$ and get, observing that $A_{F_1} \setminus \{a\}^+_{F_1} = \{c\}$, the reduct $F_2^{\{a\},\{c\},\{(a,d),(b,e),(c,e)\}} = (\{e\},\{(e,e)\})$. This AF has $\emptyset$ as only preferred extension. Consequently we get $\{a\} \in$ *prf*(F). For $\{b\}$ we get, observing that $A_{F_1} \setminus \{b\}^+_{F_1} = \emptyset$, the reduct $F_2^{\{b\},\emptyset,\{(a,d),(b,e),(c,e)\}} = (\{d\},\emptyset)$. This AF has $\{d\}$ as only preferred extension, hence $\{b,d\} \in$ *prf*(F). We conclude that *prf*$(F) = \{\{a\},\{b,d\}\}$. $\diamondsuit$

The splitting theorem is not only relevant for incremental computation in a static setting as illustrated in Example 56. It also allows to use precomputed extensions of an AF when it undergoes change. That is, however, restricted to changes which are called weak expansions [23], i.e. changes which are not among existing arguments and do not include attacks on existing arguments.

Parametrized splitting, as introduced by Baumann et al. [29], omits the restriction on the direction of attacks and allows arbitrary splittings of AFs. It is, however, only studied for stable semantics. A so-called quasi-splitting is identified just by a set of arguments, representing the partial AF to be evaluated first. In addition to propagating the results of the first sub-AF along the links of the splitting, also the first sub-AF has to be modified in order to anticipate the effects from ingoing links. The following definition covers all necessary notions needed for parametrized splitting.

Definition 92. Let F be an AF. A set $S \subseteq A_F$ is a *quasi-splitting* of F. We define $F_1^S = F|_S$ and $F_2^S = F|_{(A_F \setminus S)}$. Further let $R_S^{\rightarrow} = \{(a,b) \in R_F \mid a \in S, b \in (A_F \setminus S)\}$ and $R_S^{\leftarrow} = \{(a,b) \in R_F \mid a \in (A_F \setminus S), b \in S\}$. The modified AF $\left[F_1^S\right] = (A_1, R_1)$ is defined as

$$A_1 = S \cup \{\overline{a} \mid \exists b \in (A_F \setminus S) : (b,a) \in R_S^{\leftarrow}\},$$
$$R_1 = (R \cap (S \times S)) \cup \{(\overline{a},a),(a,\overline{a}) \mid \exists b \in (A_F \setminus S) : (b,a) \in R_S^{\leftarrow}\}.$$



(a) Initial AF F.

(b) $[F_1^S]$ and $[F_2^S]_E$ for $S = \{a_1, a_2, a_3\}$ and $E = \{\overline{a_1}, \overline{a_2}, a_3\}$.

Figure 5.2: Parametrized splitting illustrated.

Given some $E \in \left[F_1^S\right]$, let $U_E = \{a \in (S \setminus E) \mid a \notin (S \cap E)^+_F\}$. The modified AF $\left[F_2^S\right]_E = (A_2, R_2)$ is defined as

$$\begin{aligned} A_2 &= \{a \in (A_F \setminus S) \mid \nexists b \in E : (b, a) \in R_S^{\rightarrow}\} \cup \{a' \mid a \in U_E\}, \\ R_2 &= (R \cap (A_2 \times A_2)) \cup \{(a', a'), (b, a') \mid a \in U_E, (b, a) \in R_S^{\leftarrow}\} \\ &\qquad\qquad\qquad\qquad \cup \{(c, c) \mid \exists a \in E : (c, a) \in R_S^{\leftarrow}\}. \end{aligned}$$

For each argument a in F_1^S that is attacked from F_2^S, a copy $\overline{a}$ is added to $[F_1^S]$. Given a stable extension E of $[F_1^S]$ and wanting to obtain a stable extension of F we have to make sure that (i) if $\overline{a} \in E$ and a is not attacked by any other element in $(S \cap E)$ then a is attacked by an argument in F_2^S, and (ii) if $b \in E$ but it is attacked by some argument c in F_2^S then c is not included in extension of F. The construction of $[F_2^S]$ takes care of this by (i) adding self-attacking arguments a' which are attacked by the attackers of a in F_2^S, (ii) adding a self-attack to arguments such as c. We illustrate this on an example taken from [29].

Example 57. We illustrate the idea of parametrized splitting in Figure 5.2. Consider the AF F depicted in Figure 5.2a, also including the dashed part. We are interested in the quasi-splitting $S = \{a_1, a_2, a_3\}$. The dashed attacks are contained in the sets $R_S^{\rightarrow} = \{(a_3, b_4)\}$ and $R_S^{\leftarrow} = \{(b_1, a_1), (b_2, a_2), (b_3, a_3)\}$. Now the left-hand side of Figure 5.2a depicts the modified AF $[F_1^S]$, including new arguments $\overline{a_1}, \overline{a_2}, \overline{a_3}$ as each argument a_1, a_2, a_3 receives at least one attack from F_2^S. Consider $E = \{\overline{a_1}, \overline{a_2}, a_3\} \in stb([F_1^S])$. The modified AF $[F_2^S]_E$ is depicted on the right-hand side of Figure 5.2b. Since $a_2 \notin E$ and $a_2 \notin (S \cap E)^+_F = \{a_3\}^+_F = \{a_3, b_4\}$, $[F_2^S]_E$ includes the new, self-attacking argument a_2', which is attacked by b_3 and b_2, the attackers of a_2. Moreover, $a_2 \in E$ and $b_3 \rightarrow_F a_2$, hence b_3 is self-attacking in $[F_2^S]_E$. Finally, b_4 is attacked by $a_3 \in E$, hence b_4 is not included in $[F_2^S]_E$. Evaluating $[F_2^S]_E$, we get $E' = \{b_2, b_5\}$ as only stable extension. Putting things together we get $(E \cap S) \cup E' = \{a_3, b_2, b_5\} \in stb(F)$. Repeating this process for the other stable extensions of $[F_1^S]$ we are able to enumerate all stable extensions of F. $\diamondsuit$

Theorem 44 ([29])**.** *Let F be an AF and let S be a quasi-splitting of F. The following holds:*

1. *If $E_1 \in stb([F_1^S])$ and $E_2 \in stb([F_2^S]_{E_1})$, then $(E_1 \cap S) \cup E_2 \in stb(F)$.*
2. *If $E \in stb(F)$ then there is some $E_1 \in stb([F_1^S])$ with $(E_1 \cap S) = (E \cap S)$ and $(E \cap (A_F \setminus S)) \in stb([F_2^S]_{E_1})$.*

Parametrized splitting was only studied for stable semantics and is open for the other semantics. The result for "proper" splitting, as in Theorem 43, applies to stable, admissible, preferred, complete and grounded semantics. It does not hold for naive, semi-stable, stage, and ideal semantics. Counterexamples for each of these semantics can be found in [22].

5.2 Splitting of ADFs

In this section we study splitting of ADFs. We will first consider *directional splitting*, i.e. partitions of ADFs such that links between the parts are only in one direction, similar to "proper" splitting of AFs. Then we will deal with *general splitting*, giving up the condition on the links directions, a notion similar to parametrized splitting of AFs.

5.2.1 Directional Splitting

The definition of the various semantics of ADFs already suggests that not every decomposition of an ADF can be treated equivalently. In this section we focus on *directional splitting*, that is, given an ADF D, a partition of the graph underlying D into two disjoint subgraphs G_1 and G_2 such that there are no links from G_2 to G_1 in D. In other words it is splitting "along the lines" of the strongly connected components of an ADF.

Definition 93. Let $G_1 = (A_1, L_1)$ and $G_2 = (A_2, L_2)$ be directed graphs such that $A_1 \cap A_2 = \emptyset$. Moreover let $L_3 \subseteq A_1 \times A_2$. We call the tuple (G_1, G_2, L_3) a *directional splitting* of an ADF $D = (A_1 \cup A_2, L_1 \cup L_2 \cup L_3, C)$, and say that D is directionally split along L_3.

Figure 5.3 illustrates the two possible directional splittings of an exemplary ADF. Note that any other splitting of this frameworks contains links in both directions between the subgraphs and is therefore not directional.

In terms of notation, recall that, similar as for AFs, for an ADF D and a set of arguments $S \subseteq A_D$, $D|_S$ is the restriction of D to S, i.e. $\{\langle a, \varphi_a \rangle \mid a \in S\}$, under the condition that no argument $a \in S$ receives a link from $A_D \setminus S$. Moreover, $v|_S$ denotes the restriction of interpretation v to the arguments S, and $v_1 \cup v_2$ the interpretation obtained from conjoining v_1 and v_2, given that they are over disjoint sets of arguments (cf. Definition 4).

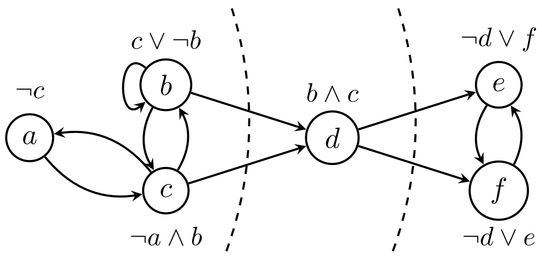


Figure 5.3: Possible directional splittings of the depicted ADF.

Definition 94. Let $G_1 = (A_1, L_1)$ and $G_2 = (A_2, L_2)$ be directed graphs such that (G_1, G_2, L_3) is a directional splitting of the ADF D. Further let v be an interpretation of $D|_{A_1}$. The *v-reduct* of D is defined as

$$D^v = \{\langle a, \varphi_a[b/\top : b \in v^{\mathbf{t}}][b/\bot : b \in v^{\mathbf{f}}][c/x_c : c \in v^{\mathbf{u}}]\rangle \mid a \in A_2\} \cup \{\langle x_c, \neg x_c\rangle \mid c \in v^{\mathbf{u}}\},$$

where x_c is a newly introduced argument for each $c \in v^{\mathbf{u}}$.

When v is an interpretation of the first part of the ADF, the v-reduct of D propagates v to the second part of the ADF by replacing occurrences of arguments in acceptance conditions (i) by the constants $\top$ (resp. $\bot$) if they are mapped to $\mathbf{t}$ (resp. $\mathbf{f}$) by v, and (ii) by new arguments if they are mapped to $\mathbf{u}$.

Compared to the reduct used for splitting of AFs (cf. Definition 91) we cannot just add a self-attack to arguments as this has no immediate meaning in ADFs. We have to make use of an additional arguments instead. Moreover, the v-reduct does not remove arguments, but replaces occurrences of arguments in acceptance conditions.

By slight abuse of notation we will, for arguments a, b, and an interpretation v with $v(b) \in \{\mathbf{t}, \mathbf{f}\}$, write $\varphi_a[b/v(b)]$ instead of $\varphi_a[b/\top]$ (resp. $\varphi_a[b/\bot]$).

The idea of directional splitting is to propagate truth values assigned to arguments by an interpretation of the first part along the links to the second part. The following example illustrates this.

Example 58. Consider the ADF D depicted on the left-hand side of Figure 5.4. Here the dotted borderline suggests the splitting (G_1, G_2, L_3) of D with $L_3 = \{(b, d), (c, d)\}$. The right part of the figure depicts the ADFs resulting from the splitting, that is, $D_1 = D|_{\{a,b,c\}}$ and D^{v_1}, where $v_1 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{u}, c \mapsto \mathbf{f}\}$ is a possible (complete) interpretation of D_1. In the acceptance condition φ_d the atom c ($v_1(c) = \mathbf{f}$) is replaced by the propositional constant $\bot$ and b ($v_1(b) = \mathbf{u}$) by the newly introduced argument x_b, which itself has acceptance condition $\varphi_{x_b} = \neg x_b$.

Note that the illustrated splitting is not the only directional splitting of D. The other possible choices are to split D along $\{(a, c)\}$, $\{(b, c), (b, d)\}$, or $\{(a, c), (b, c), (b, d)\}$. $\diamondsuit$

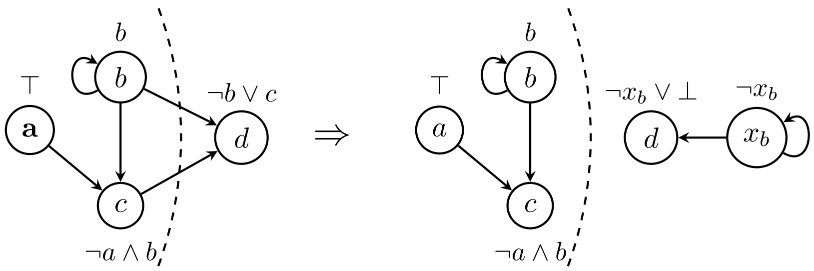


Figure 5.4: Directional splitting of the ADF D on the left into the ADFs $D|_{A_1}$ and D^{v_1}, where $A_1 = \{a, b, c\}$ and $v_1 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{u}, c \mapsto \mathbf{f}\}$ is a complete interpretation of D_1.

With the definition of a v-reduct at hand, we are ready to formulate our first results on directional splitting under the two-valued models (mod_3) of ADFs.

Theorem 45. *Let $G_1 = (A_1, L_1)$ and $G_2 = (A_2, L_2)$ be directed graphs such that (G_1, G_2, L_3) is a directional splitting of the ADF D. Further let $D_1 = D|_{A_1}$. The following holds:*

1. *If $v_1 \in mod_3(D_1)$ and $v_2 \in mod_3(D^{v_1})$, then $(v_1 \cup v_2) \in mod_3(D)$.*
2. *If $v \in mod_3(D)$, then $v|_{A_1} \in mod_3(D_1)$ and $v|_{A_2} \in mod_3(D^{v|_{A_1}})$.*

Proof. First note that for a two-valued interpretation v the v-reduct does not contain additional arguments but is built by solely replacing atoms by truth values in the acceptance conditions, i.e. $D^v = \{\langle a, \varphi_a[b/v(b) : b \in A_1]\rangle \mid a \in A_2\}$.

(1) Let $v_1 \in mod_3(D_1)$ and $v_2 \in mod_3(D^{v_1})$. Consider some $a \in A_1$. By assumption, $v_1(a) = v_1(\varphi_a)$. Since $par_D(a) \subseteq A_1$, it follows that $v_1(\varphi_a) = (v_1 \cup v_2)(\varphi_a)$. Hence $(v_1 \cup v_2)(a) = (v_1 \cup v_2)(\varphi_a)$. Now consider some $a \in A_2$. From $v_2 \in mod_3(D^{v_1})$ and the fact that v_1 is two-valued we know that $v_2(a) = v_2(\varphi_a[x/v_1(x) : x \in A_1])$. We can invert the substitution and get $v_2(a) = (v_1 \cup v_2)(\varphi_a)$. Hence $(v_1 \cup v_2)(a) = (v_1 \cup v_2)(\varphi_a)$ and the result for mod_3 follows.

(2) Let $v \in mod_3(D)$ and $v_1 = v|_{A_1}$, $v_2 = v|_{A_2}$. Again, $v_1 \in mod_3(D_1)$ is by $par_D(a) \subseteq A_1$ for each $a \in A_1$. For $a \in A_2$, it is clear that $v|_{A_2}(a) = v(a)$. Moreover, we substitute the A_1-part of v into φ_a to get $v(\varphi_a) = v|_{A_2}(\varphi_a[b/v(b) : b \in A_1]) = v|_{A_2}(\varphi_a^{D^{v|_{A_1}}})$.[1] Hence $v|_{A_2}(a) = v|_{A_2}(\varphi_a^{D^{v|_{A_1}}})$, i.e. $v|_{A_2} \in mod_3(D^{v|_{A_1}})$. □

As mentioned in the beginning of the proof, a simplified version of the v-reduct does the job for splitting under the model semantics. Since only two-valued interpretations are involved, there is no replacement of **u**-values and no additional arguments are introduced.

[1] Recall that we sometimes denote the acceptance condition of argument a in ADF D by φ_a^D to avoid ambiguities.

For the other semantics, the following lemma will be useful to prove splitting results.

Lemma 33. *Let $G_1 = (A_1, L_1)$ and $G_2 = (A_2, L_2)$ be directed graphs such that (G_1, G_2, L_3) is a directional splitting of the ADF D, and further $A = A_1 \cup A_2$ and $D_1 = D|_{A_1}$. The following holds:*

1. *If v_1 is an interpretation of D_1 and v_2 is an admissible interpretation of D^{v_1}, then*
$$(\Gamma_{D_1}(v_1) \cup \Gamma_{D^{v_1}}(v_2))|_A = \Gamma_D\left((v_1 \cup v_2)|_A\right).$$
2. *If v is an interpretation of D, then*
$$\Gamma_D(v) = \Gamma_{D_1}(v|_{A_1}) \cup \left(\Gamma_{D^{v|_{A_1}}}(v')\right)\Big|_{A_2}$$
where $v' = v|_{A_2} \cup \{x_c \mapsto \mathbf{u} \mid c \in (v|_{A_1})^{\mathbf{u}}\}$.

Proof. (1) First, as a side note, observe that the restriction of interpretations to A is due to the fact that v_2, being an interpretation of D^{v_1}, can include mappings for arguments not contained in D (namely those x_c where $v_1(c) = \mathbf{u}$).

We need to show that for each $a \in A_1$, $\Gamma_{D_1}(v_1)(a) = \Gamma_D((v_1 \cup v_2)|_A)(a)$ and for each $a \in A_2$, $\Gamma_{D^{v_1}}(v_2)(a) = \Gamma_D((v_1 \cup v_2)|_A)(a)$. The former is trivial by the fact that for each $a \in A_1$, $\varphi_a^D = \varphi_a^{D_1}$ and $par_D(a) \subseteq A_1$. The latter is by the following chain of equalities, letting $a \in A_2$ and $v_1^* = v_1|_{(v_1^{\mathbf{t}} \cup v_1^{\mathbf{f}})}$:

$$\begin{aligned}
(\Gamma_{D_1}(v_1) \cup \Gamma_{D^{v_1}}(v_2))(a) &= \Gamma_{D^{v_1}}(v_2)(a) && (5.1)\\
&= \bigsqcap_{w \in [v_2]_2} w(\varphi_a^{D^{v_1}}) && (5.2)\\
&= \bigsqcap_{w \in [v_2]_2} w(\varphi_a[b/v_1(b) : b \in (v_1^{\mathbf{t}} \cup v_1^{\mathbf{f}})][c/x_c : c \in v_1^{\mathbf{u}}]) && (5.3)\\
&= \bigsqcap_{w \in [v_2 \cup v_1^*]_2} w(\varphi_a[c/x_c : c \in v_1^{\mathbf{u}}]) && (5.4)\\
&= \bigsqcap_{w \in [v_2 \cup v_1]_2} w(\varphi_a) && (5.5)\\
&= \bigsqcap_{w \in [(v_1 \cup v_2)|_A]_2} w(\varphi_a) && (5.6)\\
&= \Gamma_D((v_1 \cup v_2)|_A)(a) && (5.7)
\end{aligned}$$

(5.1) holds since D_1 and D^{v_1} have disjoint sets of arguments and $a \in A_2$. We reach (5.4) by application of definitions and syntactic transformations. (5.4) = (5.5) holds by the fact that since $\varphi_{x_c} = \neg x_c$ and $v_2 \in adm_3(D^{v_1})$ necessarily $v_2(x_c) = \mathbf{u}$ and by definition of D^{v_1}, $v_1(c) = \mathbf{u}$. Finally (5.5) = (5.6) is by $par_D(a) \subseteq A$ for any $a \in A_2$.

(2) Again, for $a \in A_1$, $\Gamma_D(v)(a) = \Gamma_{D_1}(v|_{A_1})(a)$ is immediate by $\varphi_a^D = \varphi_a^{D_1}$ and $par_D(a) \subseteq A_1$. For $a \in A_2$, let $v' = v|_{A_2} \cup \{x_c \mapsto \mathbf{u} \mid c \in (v|_{A_1})^{\mathbf{u}}\}$. We get

$$\begin{aligned}
\Gamma_D(v)(a) &= \bigsqcap_{w \in [v]_2} w(\varphi_a) && (5.8)\\
&= \bigsqcap_{w \in [v']_2} w(\varphi_a[b/v(b) : b \in (A_1 \setminus v^{\mathbf{u}})][c/x_c : c \in (A_1 \cap v^{\mathbf{u}})]) && (5.9)\\
&= \bigsqcap_{w \in [v']_2} w(\varphi_a^{D^{v|_{A_1}}}) && (5.10)\\
&= \Gamma_{D^{v|_{A_1}}}(v')(a) && (5.11)
\end{aligned}$$

From (5.8) to (5.9), $\mathbf{t}$ and $\mathbf{f}$ values in $v|_{A_1}$ are substituted directly into φ_a and atoms with $\mathbf{u}$ values are replaced by other atoms which are $\mathbf{u}$ in v' by definition, thus (5.8) = (5.9). The rest is by definition. □

In the general case of three-valued interpretations the v-reduct (cf. Definition 94) involves the introduction of additional arguments, hence the following equalities only hold under projection on the respective arguments.

Theorem 46. *Let $\sigma \in \{adm_3, prf_3, com_3, grd_3\}$, and $G_1 = (A_1, L_1)$ and $G_2 = (A_2, L_2)$ be directed graphs such that (G_1, G_2, L_3) is a directional splitting of the ADF D. Further let $D_1 = D|_{A_1}$. The following holds:*

1. *If $v_1 \in \sigma(D_1)$ and $v_2 \in \sigma(D^{v_1})$, then $(v_1 \cup v_2)|_A \in \sigma(D)$.*
2. *If $v \in \sigma(D)$, then $v|_{A_1} \in \sigma(D_1)$ and $\exists v_2 \in \sigma(D^{v|_{A_1}})$ such that $v_2|_{A_2} = v|_{A_2}$.*

Proof. (1) Let $v_1 \in \sigma(D_1)$ and $v_2 \in \sigma(D^{v_1})$. Further define $v = (v_1 \cup v_2)|_A$. By $\sigma(D^{v_1}) \subseteq adm_3(D^{v_1})$ (cf. Proposition 4), we can follow by Lemma 33.1 that $(\Gamma_{D_1}(v_1) \cup \Gamma_{D^{v_1}}(v_2))|_A = \Gamma_D(v)$. We get that $v = \Gamma_D(v)$, hence the result for adm_3 and com_3 follows. For grd_3 it remains to show that v is the least fixpoint of Γ_D wrt. $\leq_i$. To this end assume there is a lower fixpoint $v' = \Gamma_D(v')$, $v' <_i v$. If $v'|_{A_1} <_i v_1$ then $v'|_{A_1}$ is a fixpoint of Γ_{D_1} by Lemma 33.2, a contradiction to $v_1 \in grd_3(D_1)$. If $v'|_{A_1} = v_1$ and $v'|_{A_2} <_i v_2|_{A_2}$ then $v'' = v'|_{A_2} \cup \{x_c \mapsto \mathbf{u} \mid c \in (v'|_{A_1})^{\mathbf{u}}\}$ is, again by by Lemma 33.2, a fixpoint of $\Gamma_{D^{v_1}}$, a contradiction to $v_2 \in grd_3(D^{v_1})$. Likewise, for prf_3, it remains to show that v is a greatest fixpoint of Γ_D. Assuming that there is some $v' = \Gamma_D(v')$ with $v' >_i v$ gives us, by Lemma 33.2, that $v'|_{A_1} = \Gamma_{D_1}(v'|_{A_1})$ and $v'|_{A_2} = \left(\Gamma_{D^{v'|_{A_1}}}(v'')\right)\Big|_{A_2}$. Now by $v' >_i v$ either $v'|_{A_1} >_i v_1$ or $v'|_{A_2} >_i v_2$. We get a contradiction either to $v_1 \in prf_3(D_1)$ or to $v_2 \in prf_3(D^{v_1})$, in the latter case recalling that $v_2(x_c) = \mathbf{u}$ for each $c \in (v|_{A_1})^{\mathbf{u}} \subseteq (v'|_{A_1})^{\mathbf{u}}$.

(2) Let $v \in \sigma(D)$. From Lemma 33.2 it follows that $v|_{A_1} = \Gamma_{D_1}(v|_{A_1})$ and $v' = \left(\Gamma_{D^{v|_{A_1}}}(v')\right)\Big|_{A_2}$ with $v' = v|_{A_2} \cup \{x_c \mapsto \mathbf{u} \mid c \in (v|_{A_1})^{\mathbf{u}}\}$. Hence the result follows for adm_3 and com_3. For grd_3 assume, towards a contradiction, that there is some $w <_i v|_{A_1}$

with $w = \Gamma_{D_1}(w)$. But then, by Lemma 33.1, $(w \cup v')|_A = \Gamma_D((w \cup v')|_A)$, contradicting $v \in \mathit{grd}_3(D)$ (as $(w \cup v')|_A <_i v$). Assume, on the other hand, that there is some $w <_i v'$ with $w = \Gamma_{D^{v|_{A_1}}}(w)$. Also then we get $(v|_{A_1} \cup w)|_A = \Gamma_D((v|_{A_1} \cup w)|_A)$ from Lemma 33.1, contradicting $v \in \mathit{grd}_3(D)$. For prf_3, assume, towards a contradiction, that there is some $w >_i v|_{A_1}$ with $w = \Gamma_{D_1}(w)$. But then, by Lemma 33.1, $(w \cup v')|_A = \Gamma_D((w \cup v')|_A)$, contradicting $v \in \mathit{prf}_3(D)$ (as $(w \cup v')|_A >_i v$). Assume, on the other hand, that there is some $w >_i v'$ with $w = \Gamma_{D^{v|_{A_1}}}(w)$. Also then we get $(v|_{A_1} \cup w)|_A = \Gamma_D((v|_{A_1} \cup w)|_A)$ from Lemma 33.1, contradicting $v \in \mathit{prf}_3(D)$. □

The following example illustrates the result in Theorem 46 under the complete semantics.

Example 59. The ADF D on the left-hand side of Figure 5.4 has

$$\begin{aligned}\mathit{com}_3(D) = \{&\{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}, d \mapsto \mathbf{f}\},\\ &\{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{f}, d \mapsto \mathbf{t}\},\\ &\{a \mapsto \mathbf{t}, b \mapsto \mathbf{u}, c \mapsto \mathbf{f}, d \mapsto \mathbf{u}\}\}.\end{aligned}$$

Now consider the splitting $((\{a,b,c\}, \{(a,c),(b,c),(b,b)\}), (\{d\}, \emptyset), \{(b,d),(c,d)\})$ of D. The complete interpretations of $D_1 = D|_{\{a,b,c\}}$ are $\mathit{com}_3(D_1) = \{v_1, v_2, v_3\}$ with

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

The v_1-reduct D^{v_1} is depicted on the right-hand side of Figure 5.4. The only complete interpretation of D^{v_1} is $v_4 = \{d \mapsto \mathbf{u}, x_b \mapsto \mathbf{u}\}$ and indeed $(v_1 \cup v_4)|_{A_D} \in \mathit{com}_3(D)$. Moreover, we get the v_2-reduct $D^{v_3} = \{\langle d, \neg\top \vee \bot\rangle\}$, and hence $\mathit{com}_3(D^{v_2}) = \{v_5\}$ with $v_5 = \{d \mapsto \mathbf{f}\}$ (note that $\neg\top \vee \bot \equiv \bot$). Finally, we get the v_3-reduct $D^{v_3} = \{\langle d, \neg\bot \vee \bot\rangle\}$, and hence $\mathit{com}_3(D^{v_3}) = \{v_6\}$ with $v_6 = \{d \mapsto \mathbf{t}\}$. Putting things together, we get $\mathit{com}_3(D) = \{(v_1 \cup v_4)|_{A_D}, (v_1 \cup v_5)|_{A_D}, (v_1 \cup v_6)|_{A_D}\}$, which is in accordance with above. ◊

It remains to show the result for stable semantics. There we have to make use of the results for two-valued models and grounded semantics. Recall that, for evaluation of an ADF D under stable semantics, another reduct, D^{-v}, is involved: an interpretation v is a stable model of D if it is a two-valued model and $v^{\mathbf{t}} = w^{\mathbf{t}}$ for $w \in \mathit{grd}_3(D^{-v})$.

Theorem 47. *Let $G_1 = (A_1, L_1)$ and $G_2 = (A_2, L_2)$ be directed graphs such that (G_1, G_2, L_3) is a directional splitting of the ADF D. Further let $D_1 = D|_{A_1}$. The following holds:*

1. *If $v_1 \in \mathit{stb}_3(D_1)$ and $v_2 \in \mathit{stb}_3(D^{v_1})$, then $(v_1 \cup v_2) \in \mathit{stb}_3(D)$.*
2. *If $v \in \mathit{stb}_3(D)$, then $v|_{A_1} \in \mathit{stb}_3(D_1)$ and $v|_{A_2} \in \mathit{stb}_3(D^{v|_{A_1}})$.*

Proof. (1) Let $v_1 \in \mathit{stb}_3(D_1)$ and $v_2 \in \mathit{stb}_3(D^{v_1})$ and define $v = v_1 \cup v_2$. From Theorem 45.1 we get that $v \in \mathit{mod}_3(D)$. Moreover, we know $v_1^{\mathbf{t}} = w_1^{\mathbf{t}}$ for $w_1 \in \mathit{grd}_3({D_1}^{-v_1})$ and $v_2^{\mathbf{t}} = w_2^{\mathbf{t}}$ for $w_2 \in \mathit{grd}_3((D^{v_1})^{-v_2})$. Let $G_1' = (v_1^{\mathbf{t}}, L \cap (v_1^{\mathbf{t}} \times v_1^{\mathbf{t}}))$, $G_2' = (v_2^{\mathbf{t}}, L \cap (v_2^{\mathbf{t}} \times v_2^{\mathbf{t}}))$ and observe that $(G_1', G_2', \{(a,b) \in L_3 \mid v_1(a) = \mathbf{t}\})$ is a directional splitting of D^{-v} and $(D^{v_1})^{-v_2}$ is the w_1-reduct of D^{-v}: the syntactic requirements are easily checked, keeping in mind that the v_1-reduct of D does not introduce new arguments since it is two-valued. It remains to show that the acceptance conditions coincide. For $a \in v_1^{\mathbf{t}}$ we have $\varphi_a^{{D_1}^{-v_1}} = \varphi_a[b/\bot : b \in v_1^{\mathbf{f}}] = \varphi_a[b/\bot : b \in v^{\mathbf{f}}] = \varphi_a^{D^{-v}}$. For $a \in v_2^{\mathbf{t}}$ we have

$$\varphi_a^{(D^{v_1})^{-v_2}} = \varphi_a[b/v_1(b) : b \in (v_1^{\mathbf{t}} \cup v_1^{\mathbf{f}})][b/\bot : b \in v_2^{\mathbf{f}}] \tag{5.12}$$
$$= \varphi_a[b/\bot : b \in (v_1^{\mathbf{f}} \cup v_2^{\mathbf{f}})][b/v_1(b) : b \in v_1^{\mathbf{t}}] \tag{5.13}$$
$$= \varphi_a[b/\bot : b \in v^{\mathbf{f}}][b/v_1(b) : b \in (w_1^{\mathbf{t}} \cup w_1^{\mathbf{f}})] \tag{5.14}$$
$$= \varphi_a^{(D^{-v})^{w_1}}, \tag{5.15}$$

where (5.12)=(5.13) holds since the order of replacements can be chosen arbitrarily, and (5.13)=(5.14) is by the facts that $v_1^{\mathbf{t}} = w_1^{\mathbf{t}}$ and all $b \in v_1^{\mathbf{f}}$ are already replaced by $\bot$ in the first replacement, hence they can be added without any effect in the second.

Hence, recalling that $w_1 \in \mathit{grd}_3({D_1}^{-v_1})$ and $w_2 \in \mathit{grd}_3((D^{v_1})^{-v_2})$, it follows by Theorem 46.1 that $(w_1 \cup w_2) \in \mathit{grd}_3(D^{-v})$. Since $v^{\mathbf{t}} = (w_1 \cup w_2)^{\mathbf{t}}$ we conclude that $v \in \mathit{stb}_3(D)$.

(2) Consider some $v \in \mathit{stb}_3(D)$ and let $v_1 = v|_{A_1}$ and $v_2 = v|_{A_2}$. From Theorem 45.2 we know that $v_1 \in \mathit{mod}_3(D_1)$ and $v_2 \in \mathit{mod}_3(D^{v_1})$. It remains to show that $v_1^{\mathbf{t}} = w_1^{\mathbf{t}}$ for $w_1 \in \mathit{grd}_3({D_1}^{-v_1})$ and $v_2^{\mathbf{t}} = w_2^{\mathbf{t}}$ for $w_2 \in \mathit{grd}_3((D^{v_1})^{-v_2})$. We know that $v^{\mathbf{t}} = w^{\mathbf{t}}$ for $w \in \mathit{grd}_3(D^{-v})$. Now let $G_1' = (v_1^{\mathbf{t}}, L \cap (v_1^{\mathbf{t}} \times v_1^{\mathbf{t}}))$, $G_2' = (v_2^{\mathbf{t}}, L \cap (v_2^{\mathbf{t}} \times v_2^{\mathbf{t}}))$, and $L_3' = \{(a,b) \in L_3 \mid v(a) = \mathbf{t}\}$. Observe that (G_1', G_2', L_3') is a directional splitting of D^{-v} and $(D^{v_1})^{-v_2}$ is the w_1-reduct of D^{-v}: Let $a \in A_1$. First, $a \in v_1^{\mathbf{t}}$ iff $a \in A_{{D_1}^{-v_1}}$. Moreover, we get $\varphi_a^{D^{-v}} = \varphi_a[b/\bot : b \in v^{\mathbf{f}}] = \varphi_a[b/\bot : b \in v_1^{\mathbf{f}}] = \varphi_a^{{D_1}^{-v_1}}$ by $\mathit{par}_D(a) \subseteq A_1$. Hence, by Theorem 46.2 we get that $w|_{A_1} = w_1 \in \mathit{grd}_3({D_1}^{-v_1})$. By $v^{\mathbf{t}} = w^{\mathbf{t}}$ also $v_1^{\mathbf{t}} = w_1^{\mathbf{t}}$, hence $v_1 \in \mathit{stb}_3(D_1)$. Let $a \in A_2$. Again, $a \in v_2^{\mathbf{t}}$ iff $a \in A_{(D^{v_1})^{-v_2}}$, since v_1 is two-valued and therefore no new arguments are introduced by the v_1-reduct of D. For the acceptance condition, we get

$$\varphi_a^{(D^{-v})^{w_1}} = \varphi_a[b/\bot : b \in v^{\mathbf{f}}][b/v_1(b) : b \in (w_1^{\mathbf{t}} \cup w_1^{\mathbf{f}})] \tag{5.16}$$
$$= \varphi_a[b/\bot : b \in v^{\mathbf{f}}][b/v_1(b) : b \in w_1^{\mathbf{t}}] \tag{5.17}$$
$$= \varphi_a[b/\bot : b \in (v_1^{\mathbf{f}} \cup v_2^{\mathbf{f}})][b/v_1(b) : b \in v_1^{\mathbf{t}}] \tag{5.18}$$
$$= \varphi_a[b/v_1(b) : b \in (v_1^{\mathbf{t}} \cup v_1^{\mathbf{f}})][b/\bot : b \in v_2^{\mathbf{f}}] \tag{5.19}$$
$$= \varphi_a^{(D^{v_1})^{-v_2}} \tag{5.20}$$

meaning that the w_1-reduct of D^{-v} is just $(D^{v_1})^{-v_2}$. Hence it follows from Theorem 46.2 that there is some $w_2 \in \mathit{grd}_3((D^{-v})^{w_1})$, and also $w_2 \in \mathit{grd}_3((D^{v_1})^{-v_2})$, with $w_2|_{v_2^{\mathbf{t}}} = w|_{v_2^{\mathbf{t}}}$. Since $w^{\mathbf{t}} = v^{\mathbf{t}}$ it follows that $w_2^{\mathbf{t}} = v_2^{\mathbf{t}}$. Therefore $v_2 \in \mathit{stb}_3(D^{v_1})$. ☐

We have established a uniform method of directional splitting for all ADF semantics considered in this work. The concept of a v-reduct (v being an interpretation of the sub-ADF evaluated first) propagates the values of v along the links of the splitting into the acceptance conditions of the second sub-ADF such that (i) arguments mapped to $\mathbf{t}$ (resp. $\mathbf{f}$) by v are replaced by $\top$ (resp. $\bot$) and (ii) arguments mapped to $\mathbf{u}$ lead to the introduction of a new (self-attacking) argument and are replaced by this new argument in the acceptance conditions. Note that arguments mapped to $\mathbf{u}$ could also be kept, with a new acceptance condition, sparing the introduction of a new argument (this is similarly done in [120]). However, our design choice is to have a strict separation between the two sub-ADFs and therefore abstain from taking arguments along the splitting. Finally recall that for the ADF semantics which always deliver two-valued interpretations (i.e. mod_3 and stb_3) there are no additional arguments involved in the splitting.

Incremental computation of interpretations. In the following we show directional splitting at work for the evaluation of the ADF in Figure 5.3. Let us call this ADF D. We will exemplify the evaluation under preferred semantics using the directional splitting method. First, observe that the preferred interpretations D are given as follows:

$$\begin{aligned} prf_3(D) = \{&\{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, d \mapsto \mathbf{t}, e \mapsto \mathbf{t}, f \mapsto \mathbf{t}\},\\ &\{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, d \mapsto \mathbf{t}, e \mapsto \mathbf{f}, f \mapsto \mathbf{f}\},\\ &\{a \mapsto \mathbf{t}, b \mapsto \mathbf{u}, c \mapsto \mathbf{f}, d \mapsto \mathbf{f}, e \mapsto \mathbf{t}, f \mapsto \mathbf{t}\}\}. \end{aligned}$$

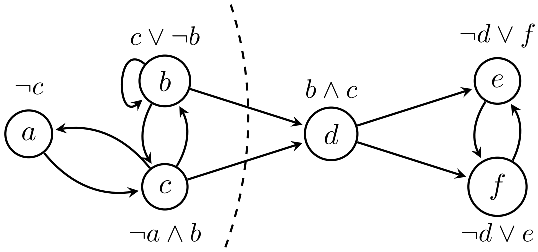
The incremental computation of $prf_3(D)$ using the techniques of directional splitting presented in this section is illustrated in Figures 5.5 and 5.6. We begin with detecting the strongly connected components of D and decide to split after the first SCC, i.e. apply the following splitting: $((A_1, L_1), (A_2, L_2), \{(d, e), (d, f)\})$ with $A_1 = \{a, b, c\}$, $L_1 = \{(a, c), (c, a), (b, b), (b, c), (c, b)\}$, $A_2 = \{d, e, f\}$, and $L_2 = \{(d, e), (d, f), (e, f), (f, e)\}$ (cf. Figure 5.5a). In the next step we compute the preferred interpretations of $D_1 = D|_{A_1}$ (cf. Figure 5.5b). We get $prf_3(D_1) = \{v_1, v_2\}$ with

$$\begin{aligned} v_1 &= \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}\}, \text{ and}\\ v_2 &= \{a \mapsto \mathbf{t}, b \mapsto \mathbf{u}, c \mapsto \mathbf{f}\}. \end{aligned}$$

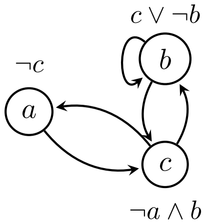
To continue the evaluation, we pick a preferred interpretation of D_1, say v_1, and propagate its valuations by constructing the v_1-reduct of D. The resulting ADF D^{v_1} is depicted in Figure 5.5c together with the illustration of its directional splitting $((A_3, L_3), (A_4, L_4), \{(d, e), (d, f)\})$ with $A_3 = \{d\}$, $L_3 = \emptyset$, $A_4 = \{e, f\}$, and $L_4 = \{(e, f), (f, e)\}$. The first ADF resulting from this splitting, $D_2 = D^{v_1}|_{A_3}$ (cf. Figure 5.5d), is easily evaluated to have

$$v_3 = \{d \mapsto \mathbf{t}\}$$

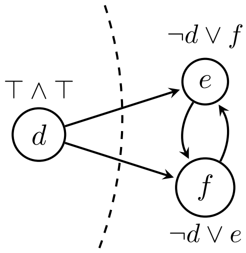
as its only preferred interpretation. We propagate v_3 along the splitting to obtain the v_3-reduct of D^{v_1}, depicted in Figure 5.6a. The evaluation of $(D^{v_1})^{v_3}$ results in $prf_3 = \{v_4, v_5\}$



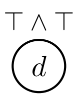
(a) ADF D, to be split along $\{(b, d), (c, d)\}$.



(b) $D_1 = D|_{A_1}$ with $A_1 = \{a, b, c\}$, having $prf_3(D_1) = \{v_1, v_2\}$, where $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}\}$ and $v_2 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{u}, c \mapsto \mathbf{f}\}$.

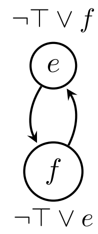


(c) D^{v_1} for $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}\}$, to be split along $\{(d, e), (d, f)\}$.

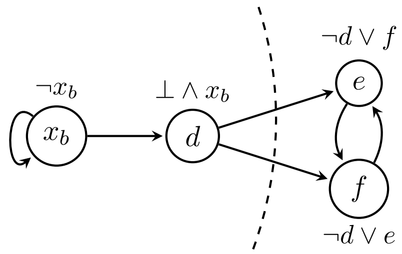


(d) $D_2 = D^{v_1}|_{A_3}$ with $A_3 = \{d\}$, having $prf_3(D_2) = \{v_3\}$, where $v_3 = \{d \mapsto \mathbf{t}\}$.

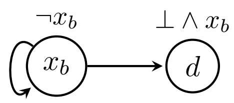
Figure 5.5: Splitting in action (1/2).



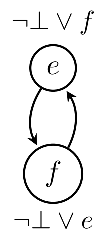
(a) $(D^{v_1})^{v_3}$ for $v_3 = \{d \mapsto \mathbf{t}\}$, having $prf_3((D^{v_1})^{v_3}) = \{v_4, v_5\}$, where $v_4 = \{e \mapsto \mathbf{t}, f \mapsto \mathbf{t}\}$ and $v_5 = \{e \mapsto \mathbf{f}, f \mapsto \mathbf{f}\}$.



(b) D^{v_2} for $v_2 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{u}, c \mapsto \mathbf{f}\}$, to be split along $\{(d, e), (d, f)\}$.



(c) $D_3 = D^{v_2}|_{A_5}$ with $A_5 = \{x_b, d\}$, having $prf_3(D_3) = \{v_6\}$, where $v_6 = \{x_b \mapsto \mathbf{u}, d \mapsto \mathbf{f}\}$.



(d) $(D^{v_2})^{v_6}$ for $v_6 = \{x_b \mapsto \mathbf{u}, d \mapsto \mathbf{f}\}$, having $prf_3((D^{v_2})^{v_6}) = \{v_4\}$, where $v_4 = \{e \mapsto \mathbf{t}, f \mapsto \mathbf{t}\}$.

Figure 5.6: Splitting in action (2/2).

with

$$v_4 = \{e \mapsto \mathbf{t}, f \mapsto \mathbf{t}\}, \text{ and}$$
$$v_5 = \{e \mapsto \mathbf{f}, f \mapsto \mathbf{f}\}.$$

We have obtained two preferred interpretations of D, namely

$$(v_1 \cup v_3 \cup v_4)|_{A_D} = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, d \mapsto \mathbf{t}, e \mapsto \mathbf{t}, f \mapsto \mathbf{t}\}, \text{ and}$$
$$(v_1 \cup v_3 \cup v_5)|_{A_D} = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, d \mapsto \mathbf{t}, e \mapsto \mathbf{f}, f \mapsto \mathbf{f}\}.$$

As the reduction, until now, did not involve introducing new arguments, the projection to A_D is not strictly necessary for these interpretations.

For the splitting of D^{v_1}, there was only one preferred interpretation of $D_2 = D^{v_1}|_{A_3}$, hence we are done here and can go back to the splitting of D. There, we have another preferred interpretation v_2 of $D_1 = D|_{A_1}$, for which we build the v_2-reduct of D. The resulting ADF D^{v_2} is depicted in Figure 5.6b. Note that, since $v_2(b) = \mathbf{u}$, the reduct involves a new argument x_b, having acceptance condition $\varphi_{x_b} = \neg x_b$. We again apply a directional splitting on D^{v_2}, namely $((A_5, L_5), (A_6, L_6), \{(d, e), (d, f)\})$ with $A_5 = \{x_b, d\}$, $L_5 = \{(x_b, x_b), (x_b, d)\}$, $A_6 = A_4$, and $L_6 = L_4$. We obtain $D_3 = D^{v_2}|_{A_5}$ (cf. Figure 5.6c) as the ADF to be evaluated first, having $\mathit{prf}_3(D_3) = \{v_6\}$ with

$$v_6 = \{x_b \mapsto \mathbf{u}, d \mapsto \mathbf{f}\}.$$

Finally, given that, we construct the v_6-reduct of D^{v_2} (depicted in Figure 5.6d) and obtain $\mathit{prf}_3((D^{v_2})^{v_6}) = \{v_4\}$. Hence, we compose the last preferred interpretation of D as follows:

$$(v_2 \cup v_6 \cup v_4)|_{A_D} = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{u}, c \mapsto \mathbf{f}, d \mapsto \mathbf{f}, e \mapsto \mathbf{t}, f \mapsto \mathbf{t}\}.$$

Note that, here, the projection on A_D is necessary, since x_b is not an argument of D.

To conclude, we have incrementally evaluated the ADF D from Figure 5.3 and obtained the preferred interpretations as compositions of the preferred interpretations of the partial ADF obtained along the way:

$$\begin{aligned}\mathit{prf}_3(D) = \{&(v_1 \cup v_3 \cup v_4)|_{A_D},\\ &(v_1 \cup v_3 \cup v_5)|_{A_D},\\ &(v_2 \cup v_6 \cup v_4)|_{A_D}\}.\end{aligned}$$

Two remarks are in order about the incremental computation we have just presented.

- The ADF does not necessarily have to be split immediately after the initial SCC(s). The split points are up to the design of the algorithm that implements the concept of directional splitting. For instance, we could have just considered the splitting of D along the links $\{(d,e),(d,f)\}$, and evaluated the ADF among arguments $\{a,b,c,d\}$ at once. In fact, we did not apply every possible splitting along the way, since the ADF D^{v_2} (cf. Figure 5.6b) could have been split along $\{(x_b,d)\}$. As it turns out, it is a wise decision not to apply this splitting, as it would result, after evaluation of the ADF $\{\langle x_b,\neg x_b\rangle\}$ with $\{x_b \mapsto \mathbf{u}\}$ as single preferred interpretation, in the same ADF as D^{v_2} modulo renaming again.

- Simplification of the acceptance conditions of the reduct can reveal redundant links. This is because the propagation of valuations, i.e. the replacement of variables by truth constants, can alter the influence of other variables. For instance, considering the v_6-reduct of D^{v_2} (cf. Figure 5.6d), we observe that $\varphi_e^{(D^{v_2})^{v_6}} = \neg\bot \vee f \equiv \top$ and $\varphi_f^{(D^{v_2})^{v_6}} = \neg\bot \vee e \equiv \top$, hence $(D^{v_2})^{v_6}$ is equivalent to $\{\langle e,\top\rangle,\langle f,\top\rangle\}$. This means that the links (e,f) and (f,e), which are attacking in D, are redundant in $(D^{v_2})^{v_6}$.

Precomputation of core arguments. Splitting can also be used, under certain circumstances, to react more efficiently to changes to an ADF. This is particularly important in dynamic settings where the reactive capabilities of an ADF system are of interest.

In order to do so, we have to identify a core of arguments which (i) have no ingoing links from arguments outside of the core and (ii) will not be affected by further changes to the ADF. Given such a core and positive results for directional splitting, we have to compute the interpretations for the ADF restricted to the core only once and use these results for the evaluation of the remaining ADF every time a change happens.

As an example consider the ADF D depicted in Figure 5.7. We regard the arguments a,b,c,d as the core arguments, illustrated by the dashed circle, and assume further changes to occur only outside of this core. First of all we showcase the evaluation of the ADF, as it is, under preferred semantics with the application of directional splitting. We begin with the ADFs restricted to the core arguments and get $\mathit{prf}_3(D|_{\{a,b,c,d\}}) = \{v_1,v_2\}$ with

$$v_1 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, d \mapsto \mathbf{f}\}, \text{ and}$$
$$v_2 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{f}, c \mapsto \mathbf{u}, d \mapsto \mathbf{t}\}.$$

Beginning with v_1, we construct the reduct and get, with simplification of acceptance conditions, $D^{v_1} = \{\langle e,\neg f\rangle,\langle f,\neg e\rangle\{\langle i,\top\rangle,\langle j,\neg i\rangle\{\langle m,\top\rangle\}$. It can be seen that the v_1-reduct is composed of three components. We have $\mathit{prf}_3(D^{v_1}) = \{v_3,v_4\}$ with

$$v_3 = \{e \mapsto \mathbf{t}, f \mapsto \mathbf{f}, i \mapsto \mathbf{t}, j \mapsto \mathbf{f}, m \mapsto \mathbf{t}\}, \text{ and}$$
$$v_4 = \{e \mapsto \mathbf{f}, f \mapsto \mathbf{t}, i \mapsto \mathbf{t}, j \mapsto \mathbf{f}, m \mapsto \mathbf{t}\}.$$

Likewise, for v_2 we get $D^{v_2} = \{\langle e,\bot\rangle,\langle f,\bot\rangle\} \cup \{\langle i,\bot\rangle,\langle j,\bot\rangle\} \cup \{\langle x_c,x_c\rangle,\langle m,x_c\rangle\}$, and furthermore $\mathit{prf}_3(D^{v_2}) = \{v_5\}$ with

$$v_5 = \{e \mapsto \mathbf{f}, f \mapsto \mathbf{f}, i \mapsto \mathbf{f}, j \mapsto \mathbf{f}, x_c \mapsto \mathbf{u}, m \mapsto \mathbf{u}\}.$$

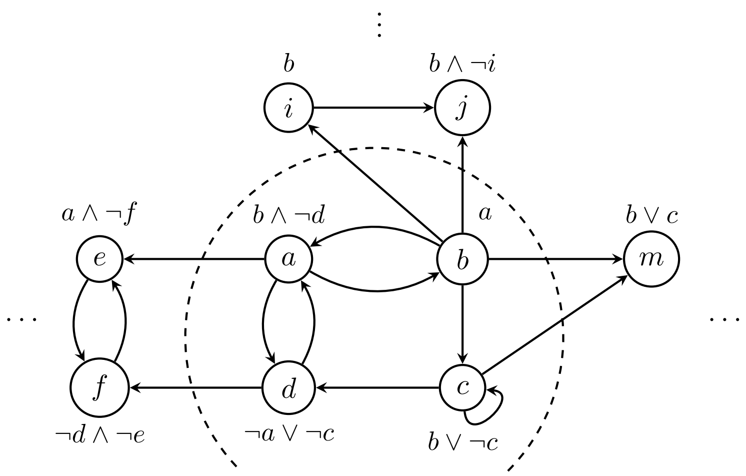


Figure 5.7: ADF illustrating the possibility of precomputing the semantics of a core part of the ADF.

We conclude that $\mathit{prf}_3(D) = (v_1 \cup v_3, v_1 \cup v_4, (v_2 \cup v_5)|_{A_D})$.

More important than the evaluation of the ADF as depicted in Figure 5.7 is the fact that due to the assumption that arguments $\{a, b, c, d\}$ will not be affected by future changes, we can store $\{v_1, v_2\}$ as the preferred interpretations of this core and use it to build the reduct whenever a change happens. Moreover, the incremental computation leads to further decomposition of the non-core ADF, given that potential additional arguments do not connect the components.

Note that the restrictions on the core are rather limiting. While it can be safe to assume that certain arguments and their relations will not undergo any change in the future (consider, for instance, a discussion, where a certain topic is agreed upon among participants), it seems too restrictive allow only outgoing links from the core. General splitting, as presented in the following section, tries to overcome this limitation.

5.2.2 General Splitting

In the previous section we have dealt with splittings of ADFs along the lines of strongly connected components. However, the graph induced by an ADF may not be sparse enough to be suitable for directional splitting. Therefore we introduce the notion of *general splitting*, which imposes no restrictions on the decomposition of the ADF. It follows the same idea as parametrized splitting for AFs presented in Section 5.1. The increased modeling power of ADFs will allow us, however, to propose a method which is conceptually simpler than parametrized splitting. That is, we will define a transformation (L-elimination), which makes an ADF amenable for directional splitting. Then we can make use of the results of Section 5.2.1.

In the following we give some preliminary results on general splitting of ADFs. This is identified just by a subset of the arguments of an ADF, which shall represent the first part of the ADF we want to split.

Definition 95. Given an ADF D we call a set $S \subseteq A_D$ a *general splitting* of D.

First we consider two-valued models. Here we can clear the way for directional splitting by transforming a given ADF while preserving equality with respect to two-valued models.

Definition 96. Given an ADF D, let $L \subseteq L_D$ be a subset of links in D. We define $L^- = \{b \mid (b,a) \in L\}$. Moreover, the L-*elimination* of D is defined as

$$\begin{aligned} D^L = & \{\langle a, \varphi_a[b/x_b : b \in L^-]\rangle \mid a \in A_D\} \cup \\ & \{\langle x_b, x_b\rangle \mid b \in L^-\} \cup \\ & \{\langle \omega(D^L), \neg\left(\bigwedge_{b\in L^-}(b \leftrightarrow x_b)\right) \wedge \neg\omega(D^L)\rangle\}, \end{aligned}$$

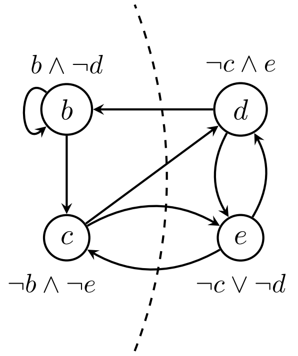
where $\omega(D^L)$ and x_b for each $b \in L^-$ are newly introduced arguments.

As the name suggests, an L-elimination of D removes the links L from D. In addition, it adds some additional arguments to preserve the two-valued models of D. While this transformation is of some interest of its own, it can be used as a preparatory measure to make an ADF amenable for directional splitting.

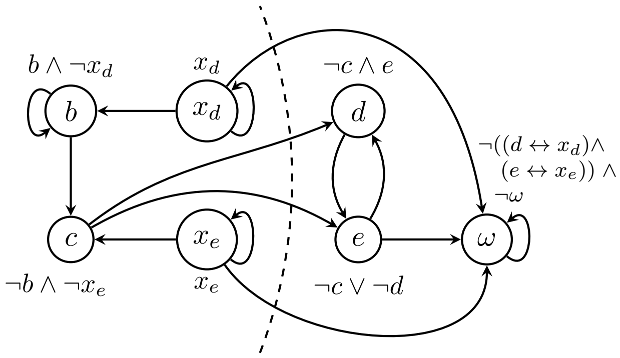
Example 60. The concept of an L-elimination is illustrated in Figure 5.8. We start from the ADF depicted in Figure 5.8a, say D, which has $mod_3(D) = \{b \mapsto \mathbf{f}, c \mapsto \mathbf{f}, d \mapsto \mathbf{t}, e \mapsto \mathbf{t}\}$. Note that D is composed of a single strongly connected component, hence there is no directional splitting of D. We envisage the removal of the links (d,b) and (e,c). Therefore let $L = \{(d,b),(e,c)\}$. The L-elimination is depicted in Figure 5.8b. As $L^- = \{d,e\}$, D^L contains the additional, self-supporting arguments x_d and x_e as well as the argument $\omega(D^L)$ (abbreviated as ω in Figure 5.8b). Now observe that $mod_3(D^L) = \{b \mapsto \mathbf{f}, c \mapsto \mathbf{f}, x_d \mapsto \mathbf{t}, x_e \mapsto \mathbf{t}, d \mapsto \mathbf{t}, e \mapsto \mathbf{t}, w \mapsto \mathbf{f}\}$, i.e. $mod_3(D) = mod_3(D^L)\Big|_{A_D}$. Proposition 36 will show that this correspondence holds in general. $\lozenge$

Compared to parametrized splitting of AFs (cf. Definition 92), the L-elimination shares similar aspects with the modified AF $[F_1^S]$. For each arguments which is attacked from the second part, an additional argument is added. This is now, however, self-supporting and can stand in relation to the corresponding argument in another form than attack. A remarkable difference is that we can encode all the conditions, which had to be explicitly propagated for parametrized splitting of AFs, now into the acceptance condition of the new argument ω. In this way will be able, after evaluation of the first part, to simply apply directional splitting and read off the desired result.

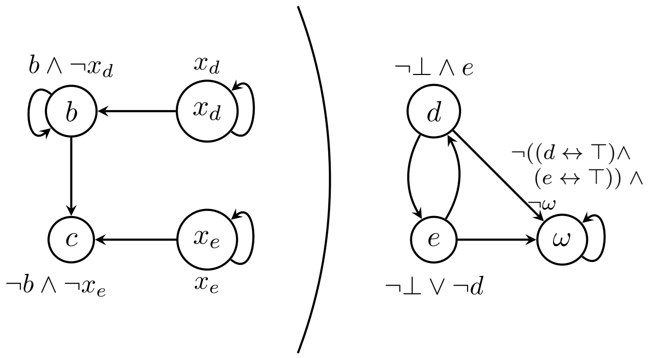
First, we show that new arguments x_b are proper copies of b in a semantical sense.



(a) ADF D composed of a single SCC.



(b) D^L with $L = \{(d, b), (e, c)\}$.



(c) Directional splitting of D^L along $\{(c, d), (c, e), (x_d, \omega), (x_e, \omega)\}$. On the left is $D^L\big|_{\{b,c,x_d,x_e\}}$ and on the right is $(D^L)^{v_1}$ for $v_1 = \{b \mapsto \mathbf{f}, c \mapsto \mathbf{f}, x_d \mapsto \mathbf{t}, x_e \mapsto \mathbf{t}\}$.

Figure 5.8: General splitting $S = \{b, c\}$ of the ADF D.

Lemma 34. *Let D be an ADF and $L \subseteq L_D$. For any model $v \in mod_3(D^L)$ and for any $b \in L^-$, it holds that $v(b) = v(x_b)$.*

Proof. Let $v \in mod_3(D^L)$ and $b \in L^-$ and assume towards a contradiction that $v(b) \neq v(x_b)$. Hence $\neg(\bigwedge_{b \in L^-}(b \leftrightarrow x_b))$ surely evaluates to true. Therefore we cannot find a valid mapping for $\omega(D^L)$, since if $v(\omega(D^L)) = \mathbf{t}$ then $v(\varphi_{\omega(D^L)}) = \mathbf{f}$ and if $v(\omega(D^L)) = \mathbf{f}$ then $v(\varphi_{\omega(D^L)}) = \mathbf{t}$. We end up in a contradiction to $v \in mod_3(D^L)$. □

Now we can show that the L-elimination indeed preserves equality of two-valued models under projection:

Proposition 36. *Given an ADF D, for any $L \subseteq L_D$, the following holds:*

1. *If $v \in mod_3(D^L)$, then $v|_{A_D} \in mod_3(D)$.*
2. *If $v \in mod_3(D)$, then $\exists v' \in mod_3(D^L) : v = v'|_{A_D}$.*

Proof. (1) Let $v \in mod_3(D^L)$ and $a \in A_D$. We need to show, knowing that $v(a) = v(\varphi_a[b/x_b : b \in L^-])$, that $v|_{A_D}(a) = v|_{A_D}(\varphi_a)$. By Lemma 34 it must hold for each $b \in L^-$ that $v(b) = v(x_b)$, hence $v(a) = v(\varphi_a)$ and since $par_D(a) \subseteq A_D$, we conclude that $v|_{A_D}(a) = v|_{A_D}(\varphi_a)$.

(2) Consider some $v \in mod_3(D)$ and let $v' = v \cup \{x_b \mapsto v(b) \mid b \in L^-\} \cup \{\omega(D^L) \mapsto \mathbf{f}\}$. Obviously $v'(x_b) = v'(\varphi_{x_b})$. Now consider some $a \in A_D$. It holds that $v(a) = v'(a) = v(\varphi_a)$. In order to ensure $v'(\varphi_{\omega(D^L)}) = v'(\omega(D^L)) = \mathbf{f}$, it must hold that $v'(b) = v'(x_b)$ for all $b \in L^-$. Hence $v'(a) = v'(\varphi_a[b/x_b : b \in L^-])$, showing that $v' \in mod_3(D^L)$. □

This allows us to apply directional splitting under two-valued models along any desired partition of arguments after a suitable transformation. First the transformation eliminating all links of one direction preserves equality of two-valued models (cf. Proposition 36), and second the computation of two-valued models can be carried out in two stages by directionally splitting the ADF (cf. Theorem 45).

Theorem 48. *Given an ADF D, let $S \subseteq A_D$ be a general splitting of D. Further let $L = \{(b,a) \in L_D \mid a \in S, b \in (A_D \setminus S)\}$ and $X = \{x_b \mid b \in L^-\}$. The following holds:*

1. *If $v_1 \in mod_3(D^L\big|_{(S \cup X)})$ and $v_2 \in mod_3\left((D^L)^{v_1}\right)$, then $(v_1 \cup v_2)|_{A_D} \in mod_3(D)$.*
2. *If $v \in mod_3(D)$, then $\exists v_1, v_2$ s.t. $(v_1 \cup v_2)|_{A_D} = v$ and $v_1 \in mod_3\left(D^L\big|_{(S \cup X)}\right)$ and $v_2 \in mod_3\left((D^L)^{v_1}\right)$.*

Proof. (1) Let $v_1 \in mod_3\left(D^L\big|_{(S \cup X)}\right)$ and $v_2 \in mod_3\left((D^L)^{v_1}\right)$. It can be seen that $(((S \cup X), L_1), ((A_D \setminus S) \cup \{\omega(D^L)\}, L_2), \{(a,b) \in L_D \mid a \in S, b \in (A_D \setminus S)\})$ is a directional splitting of D^L. Therefore it follows from Theorem 45.1 that $(v_1 \cup v_2) \in mod_3(D^L)$. Moreover, we get by Proposition 36.1 that $(v_1 \cup v_2)|_{A_D} \in mod_3(D)$.

(2) Let $v \in mod_3(D)$. By Proposition 36.2 it follows that there is a $v' \in mod_3(D^L)$ such that $v'|_{A_D} = v$ Applying directional splitting along $\{(a,b) \in L_D \mid a \in S, b \in (A_D \setminus S)\}$ it follows by Theorem 45.2 that $v'|_{(S \cup X)} \in mod_3\left(D^L\big|_{(S \cup X)}\right)$ and $v'|_{((A_D \setminus S) \cup \{\omega(D^L)\})} \in mod_3((D^L)^{v_1})$. By $v'|_{A_D} = v$ also $(v'|_{(S \cup X)} \cup v'|_{((A_D \setminus S) \cup \{\omega(D^L)\})})|_{A_D} = v$, hence the result follows. □

The following example illustrates the idea of Theorem 48.

Example 61. Again consider the ADF D in Figure 5.8a and its general splitting $S = \{b, c\}$. The L-elimination of D with $L = \{(b,a) \in L_D \mid a \in S, b \in (A_D \setminus S)\} = \{(d,b),(e,c)\}$ in Figure 5.8b was already discussed in Example 60. It paves the way for directional splitting of D^L along $\{(c,d),(c,e),(x_d,\omega),(x_e,\omega)\}$. The resulting ADF

$D^L\big|_{\{b,c,x_d,x_e\}}$ (left-hand side of Figure 5.8c) has a single two-valued model, namely $v_1 = \{b \mapsto \mathbf{f}, c \mapsto \mathbf{f}, x_d \mapsto \mathbf{t}, x_e \mapsto \mathbf{t}\}$. The v_1-reduct of D^L (right-hand side of Figure 5.8c) then has $mod_3((D^L)^{v_1}) = v_2 = \{d \mapsto \mathbf{t}, e \mapsto \mathbf{t}, \omega \mapsto \mathbf{f}\}$. It easy to see now that $mod_3(D) = \{(v_1 \cup v_2)|_{A_D}\}$. ◊

We now turn to general splitting under the admissible semantics. A transformation in the fashion of Definition 96 is not possible since we cannot force an interpretation to have equal truth values for two arguments in the three-valued setting, as we did with the acceptance condition of $\omega(D^L)$. Therefore we will have to apply local transformations on each of the sub-frameworks obtained by the splitting.

Definition 97. Given an ADF D, let $S \subseteq A_D$ be a general splitting of D. Further let $B = \{b \in (A_D \setminus S) \mid \exists a \in S : (b, a) \in L_D\}$. The *primary slice* of D wrt. S is defined as

$$\begin{aligned} D^S = &\{\langle a, \varphi_a[b/x_b : b \in B]\rangle \mid a \in S\} \cup \\ &\{\langle x_b, x_b\rangle \mid b \in B\}. \end{aligned}$$

where x_b for $b \in B$ are newly introduced arguments. Moreover, if v is an interpretation of D^S, the *extended v-reduct* of D wrt. S is defined as

$$D^{S,v} = D^v \cup \{\langle \omega(D^{S,v}), \bigwedge_{b \in B, v(x_b)=\mathbf{t}} b \quad \wedge \bigwedge_{b \in B, v(x_b)=\mathbf{f}} (\neg b)\rangle\}.$$

where the newly introduced $\omega(D^{S,v})$ is called *insurance argument* of $D^{S,v}$.

When S is a general splitting of some ADF D, all arguments not in S which have links to S are simulated in the primary slice of D by new, self-supporting arguments. This has the effect that these arguments can have an arbitrary truth value in an admissible interpretation. In the extended reduct of D, another additional argument, $\omega(D^{S,v})$, ensures that only "valid" interpretations survive the splitting. This construction can be regarded as a kind of guess-and-check-procedure, where evaluation of the primary slice guess the valuation of the arguments affected by the splitting, and evaluation of the extended v-reduct checks validity of the guess.

The following theorem shows how Definition 97 can be utilized to incrementally compute admissible extensions.

Theorem 49. *Given an ADF D and a general splitting $S \subseteq A_D$ thereof, let $B = \{b \in (A_D \setminus S) \mid \exists a \in S : (b, a) \in L_D\}$. The following holds:*

1. *If $v_1 \in adm_3(D^S)$, $v_2 \in adm_3(D^{S,v_1})$, and $v_2(\omega(D^{S,v_1})) = \mathbf{t}$, then $(v_1 \cup v_2)|_{A_D} \in adm_3(D)$.*
2. *If $v \in adm_3(D)$, then $\exists v_1, v_2$ s.t. $(v_1 \cup v_2)|_{A_D} = v$ and $v_1 \in adm_3(D^S)$ and $v_2 \in adm_3(D^{S,v_1})$ and $v_2(\omega(D^{S,v_1})) = \mathbf{t}$.*

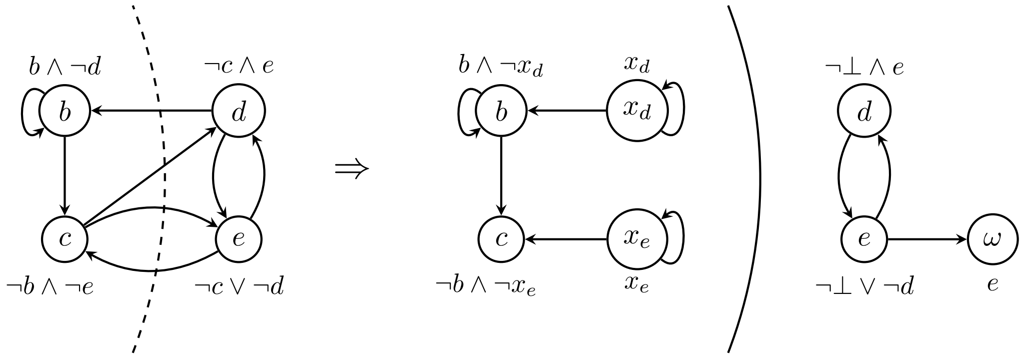


Figure 5.9: General splitting $S = \{b, c\}$ of the ADF D on the left under admissible semantics. The right side depicts the primary slice D^S of the splitting as well as the extended v_1-reduct wrt. S with $v_1 = \{b \mapsto \mathbf{u}, c \mapsto \mathbf{f}, x_d \mapsto \mathbf{u}, x_e \mapsto \mathbf{t}\}$.

Proof. (1) Let $v_1 \in adm_3(D^S)$ and $v_2 \in adm_3(D^{S,v_1})$ such that $v_2(\omega(D^{S,v_1})) = \mathbf{t}$. First observe that, by definition of $\varphi_{\omega(D^{S,v})}$ and $v_2(\omega(D^{S,v_1})) = \mathbf{t}$, for any $b \in B$ it holds that if $v_1(x_b) \neq \mathbf{u}$ then $v_1(x_b) = v_2(b)$.

Let $a \in S$. We know that $v_1(a) \leq_i \bigsqcap_{w \in [v_1]_2} w(\varphi_a[b/x_b : b \in B])$ from $v_1 \in adm_3(D^S)$. Consider an arbitrary $w \in [v_1]_2$. Further let $w' \in [v_1 \cup v_2]_2$ such that $w(c) = w'(c)$ for all $c \in A_{D^S}$. Since by the previous observation $v_1(x_b) \leq_i v_2(b)$ for all $b \in B$, it holds that $w'(x_b) = w'(b)$ for all $b \in B$. Therefore also $w(x_b) = w'(b)$, and hence $w(\varphi_a[b/x_b : b \in B]) = w'(\varphi_a)$. As $w \in [v_1]_2$ was chosen arbitrarily it follows that $\bigsqcap_{w \in [v_1]_2} w(\varphi_a[b/x_b : b \in B]) = \bigsqcap_{w \in [v_1 \cup v_2]_2} w(\varphi_a)$. By $par_D(a) \subseteq A_D$ it also holds that $v_1(a) = (v_1 \cup v_2)|_{A_D}(a)$. We conclude that $(v_1 \cup v_2)|_{A_D}(a) \leq_i \bigsqcap_{w \in [(v_1 \cup v_2)|_{A_D}]_2} w(\varphi_a)$.

For $a \in (A_D \setminus S)$, $(v_1 \cup v_2)|_{A_D}(a) \leq_i \Gamma_D((v_1 \cup v_2)|_{A_D})(a)$ follows by the same reasoning as in the proof of Theorem 46.

(2) Consider some $v \in adm_3(D)$. Let $v_1 = v|_S \cup \{x_b \mapsto v(b) \mid b \in B\}$ and $v_2 = v|_{(A_D \setminus S)} \cup \{x_c \mapsto \mathbf{u} \mid c \in (v|_S)^{\mathbf{u}}\} \cup \{\omega(D^{S,v_1}) \mapsto \mathbf{t}\}$. We argue that $v_1 \in adm_3(D^S)$ and $v_2 \in adm_3(D^{S,v_1})$. Since $\varphi_{x_b} = x_b$ it surely holds that $v_1(x_b) \leq_i \Gamma_{D^S}(v_1)(x_b)$. For any $a \in S$, $v_1(a) \leq_i \Gamma_{D^S}(v_1)(a)$ follows from the fact that $v_1(x_b) = v(b)$ by definition. Finally Theorem 46 implies that $v_2(a) \leq_i \Gamma_{D^{S,v_1}}(v_2)(a)$ for all $a \in (A_D \setminus S)$, hence the result follows. □

The following example illustrates Theorem 49 in practice.

Example 62. Let D be the ADF on the left side of Figure 5.9 and consider its general splitting $S = \{b, c\}$. We illustrate the computation of the admissible interpretations $v = \{b \mapsto \mathbf{u}, c \mapsto \mathbf{f}, d \mapsto \mathbf{u}, e \mapsto \mathbf{t}\}$ and $v' = \{b \mapsto \mathbf{u}, c \mapsto \mathbf{f}, d \mapsto \mathbf{t}, e \mapsto \mathbf{t}\}$ of D via the splitting S. First of all we consider the primary slice D^S and observe that $v_1 = \{b \mapsto \mathbf{u}, c \mapsto \mathbf{f}, x_d \mapsto \mathbf{u}, x_e \mapsto \mathbf{t}\}$ is an admissible interpretation thereof. Now the extended v_1-reduct D^{S,v_1} is depicted at the very right part of Figure 5.9. We observe that the admissible interpretations of D^{S,v_1} having $\omega(D^{S,v_1}) \mapsto \mathbf{t}$ are $v_2 = \{d \mapsto \mathbf{u}, e \mapsto \mathbf{t}, \omega(D^{S,v_1}) \mapsto \mathbf{t}\}$ and $v_2' = \{d \mapsto \mathbf{t}, e \mapsto \mathbf{t}, \omega(D^{S,v_1}) \mapsto \mathbf{t}\}$. Now it indeed holds that $v = (v_1 \cup v_2)|_{A_D}$ and $v' = (v_1 \cup v_2')|_{A_D}$.

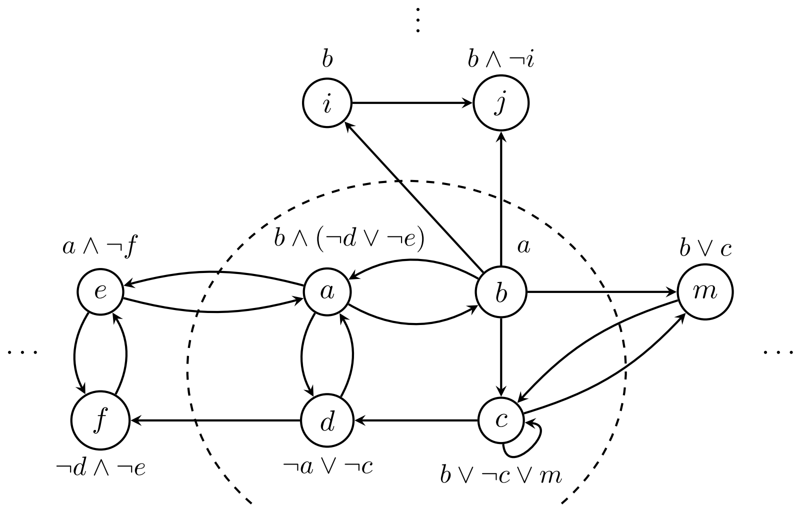
On the other hand consider the admissible interpretation $w_1 = \{b \mapsto \mathbf{t}, c \mapsto \mathbf{f}, x_d \mapsto \mathbf{f}, x_e \mapsto \mathbf{u}\}$ of D^S. We get $D^{S,w_1} = \{\langle d, \top \wedge e\rangle, \langle e, \top \vee \neg d\rangle, \langle \omega(D^{S,w_1}), \neg d\rangle\}$ and observe that there is no $w_2 \in \mathit{adm}_3(D^{S,w_1})$ with $w_2(\omega(D^{S,w_1})) = \mathbf{t}$. This is as expected since there is no $w \in \mathit{adm}_3(D)$ such that $w(b) = \mathbf{t}$ and $w(c) = \mathbf{f}$. $\Diamond$

Precomputation of core arguments. Given a set of arguments of which we know they will not be affected by future changes, we can apply general splitting to boost performance in a dynamic setting. For directional splitting this approach was rather limited, as the core arguments were not allowed to have incoming links. For general splitting we have no conditions on the neighborhood of the core. We can compute the interpretations of the L-elimination of the ADF restricted to core arguments with L being the incoming links of those arguments and store them for later use when we need to recompute the interpretations of the full ADF due to a change.

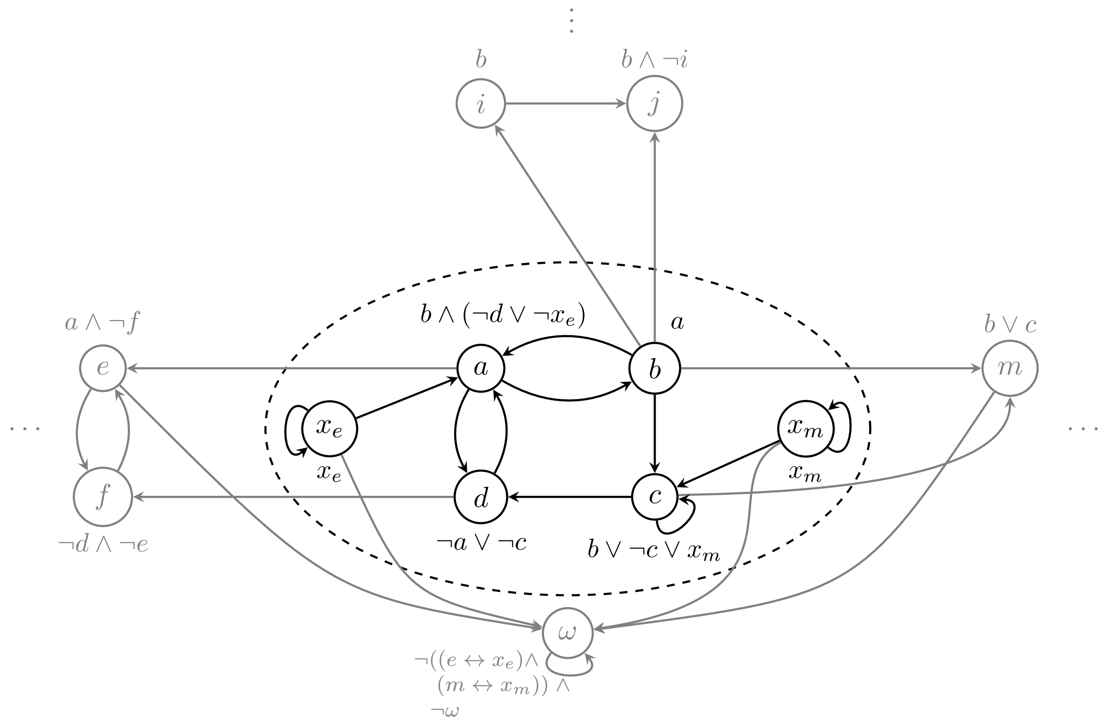
We exemplify this in Figure 5.10. The ADF D' depicted in Figure 5.10a is similar to the ADF D in Figure 5.7, but no with links from e to a and from m to c and hence also updated acceptance conditions for a and c. The core arguments are again given by $\{a, b, c, d\}$. Now for the precomputation of the core, we cannot, in contrast to directional splitting, just dismiss the arguments outside the core. Instead we construct the L-elimination of D' with $L = \{(e, a), (m, c)\}$ being the ingoing links of the core arguments. D'^L is depicted in Figure 5.10b. Due to the additional arguments involved in the elimination of links, we have increased the size of the core. Nevertheless, under the assumption that the core arguments will not be changed (the acceptance condition remains the same) we can compute the interpretations of D'^L once and use them for further evaluations of the ADF. Finally note that also the acceptance condition of $\omega(D'^L)$ (abbreviated as ω in Figure 5.10) is also fixed once the L-elimination is constructed.

We have presented results for general splitting for two-valued models and admissible semantics. In general these results do not carry over to the other semantics under consideration. Nevertheless a procedure for gradually computing the preferred interpretations is derivable from Theorem 49. This can be achieved by using the splitting procedure to determine the admissible interpretations and finally selecting the $\leq_i$-maximal elements.

Beyond the theoretical interest in splitting techniques, which gives insights on different semantics, the motivation of studying splitting is of practical nature, as we have shown in the examples along the way. An empirical evaluation of the effect of applying splitting is subject to future work.



(a) ADF D' where arguments a, b, c, d are identified as core arguments.



(b) L-elimination of D' with $L = \{(e, a), (m, c)\}$.

Figure 5.10: Precomputation of core arguments with general splitting.

CHAPTER 6

Discussion

6.1 Summary

In this thesis, we have contributed to the advancement of the study of abstract argumentation in several ways. In particular, we dealt with aspects of expressiveness and dynamics of argumentation semantics within the formalisms of Dung's abstract argumentation frameworks as well as Brewka and Woltran's abstract dialectical frameworks.

In the field of expressiveness we have first complemented the work on (general) realizability, as introduced in [146]. We have studied the signatures of two semantics, complete and resolution-based grounded, which had been dealt with erroneously or not at all before. As a remarkable side result, we have shown that the resolution-based grounded semantics, a semantics fulfilling many desirable abstract properties, is not capable of realizing simple extension-sets such as $\{\{a\}, \{b\}, \{c\}\}$. Moreover, we have provided a result that shows that every semantics which follows very basic principles cannot be more expressive than preferred or semi-stable semantics. In other words, preferred and semi-stable semantics are most expressive among all "reasonable" semantics. Then, we have studied closure properties of AF semantics, where complete semantics turned out to be the only semantics not closed under intersection. While of interest on their own, these results turned out to be essential for revision of AFs following the AGM approach. Finally, we have studied the quantitative diversity of the considered semantics, i.e. the maximum number of realizable extensions, and have shown that realizability is, assuming the input to be an explicit representation of the desired extension-set, decidable in polynomial time.

In the next section of the chapter on expressiveness we studied the class of AFs without rejected arguments: compact argumentation frameworks. As each semantics gives rise to its own class of compact AFs, we first gave a complete picture of the relations between these classes. Tackling the question to which extent rejected arguments contribute to the expressiveness of argumentation semantics we first presented the result showing that only for conflict-free and naive semantics we can translate every AF to an AF without rejected arguments. Then we studied the relations between compact signatures, i.e. the collections of extension-sets realizable by compact AFs, giving insights on the limits of global disagreement (a notion introduced in [50]) that can be modeled within compact AFs. There we obtained a picture which is significantly different to general signatures. For instance, while preferred and semi-stable semantics are strictly more expressive than stage semantics in the general setting, their expressiveness is pairwise incomparable in the compact setting.

Moreover, we studied realizability in the setting of input-output AFs as introduced by Baroni et al. [17], which differs from general realizability in two aspects: on the one hand, it is about enforcing a set of extensions for any input rather than a single set of extensions; on the other hand, one can exploit non-output arguments that are not seen outside a sub-AF for the realization. We have characterized all realizable two-valued I/O-specifications for the majority of semantics as well as all realizable three-valued I/O-specifications for preferred and grounded semantics.

After recalling results from Pührer [175] and Strass [188] on realizability in ADFs and clarifying the closure properties similar as for AFs, we presented an algorithm for realizability in which AFs, SETAFs, bipolar ADFs and general ADFs can be treated in a uniform way. The algorithm makes use of so-called propagators, by which it can be adapted to the different formalisms and semantics. We also presented an implementation of our framework in answer set programming and obtained several novel expressiveness results using our implementation.

In the field of dynamics we have first presented a generic solution to the problem of revision of AFs, which applies to the broad class of I-maximal semantics. We have considered the setting where revision is done with respect to another AF representing the new information, and is again resulting in a single AF. We have given a representation result showing a one-to-one correspondence between revision operators adhering to all AGM postulates (reformulated for AFs) and revision operators based on certain rankings on interpretations. Due to the limited expressiveness of AF semantics, a refined version of rankings compared to classical AGM revision as well as an additional postulate, adapted from previous work on Horn revision [84], had to be added in order to obtain the result. This result is significant as it allows any revision operator from the propositional setting to be applied in the AF context. We analyzed the computational complexity of Dalal's operator for AF revision, where hardness goes up to Θ_3^{P} for revision under preferred and semi-stable semantics. For the revision of ADFs we have characterized operators under preferred semantics and exemplified these results by a three-valued version of Dalal's operator. While admissible semantics yield only a single rational operator, we have proposed an alternative family of revision operators combining admissible and preferred semantics. Their representation by rankings is based on the *prf-adm*-compliance, a generalization of similar notions used previously [85, 84, 89].

Finally, we provided splitting results for ADFs. We showed that incremental computation is possible for all ADF semantics under consideration, under the condition that the splitting is done between strongly connected components. Since all operations involved in the splitting can be done efficiently, this lays the basis for optimization techniques for the evaluation of ADFs. Moreover, we have shown general splitting results for two-valued models and admissible semantics of ADFs. This is particularly interesting for pre-evaluation of parts of an ADF to be able to react efficiently to syntactic changes to the ADF.

6.2 Related Work

Expressiveness. Studying realizability and signatures of AFs has been initiated in [146]. It was inspired by previous work on the *intertranslatability* of AF semantics [109, 107]: a translation from semantics σ to semantics τ manipulates any given AF F into AF F' such that $\sigma(F)$ and $\tau(F')$ are related in a specific way. In case of an exact translation, that is if the translation always results in $\sigma(F) = \tau(F')$, it means that τ is at least as expressive as σ, i.e. $\Sigma^{\sigma}_{\mathrm{AF}} \subseteq \Sigma^{\tau}_{\mathrm{AF}}$ in signature terms. While the work on intertranslatability studies the expressiveness of semantics in relative terms as well as the efficiency of such translations, the work on realizability is concerned with exact characterization of the signatures of semantics.

A variant of realizability for preferred and semi-stable semantics was studied by Dyrkolbotn [112]. There, new arguments are not only allowed to be introduced, but the new arguments can also take on arbitrary acceptance statuses, i.e. the evaluation of the realizing AF is done with respect to projection on the original arguments. Under these assumptions, preferred and semi-stable semantics are shown to be able to express any extension-set. Compared to this approach, our results deal with additional arguments as follows: general realizability as presented in Section 3.2 allows additional arguments but disallows them to be part of any extensions; compact realizability as dealt with in Section 3.3.2 completely rules out the use of arguments which do not occur in the given set of extensions; finally, in the unifying approach (cf. Section 3.6) we uniformly use three-valued interpretations as the underlying model theory meaning that arguments cannot be “invisible” any more since the underlying vocabulary of arguments is always implicit in each interpretation.

Besides rejected arguments, *implicit conflicts* were identified as the other “hidden power” [31] of AFs. Two arguments are in implicit conflict with respect to a semantics σ in an AF F if they are never accepted together in an extension of F under σ, but also do not attack each other in F. Just as it is considered desirable to remove rejected arguments for computational tasks, the explication of AFs, i.e. adding attacks between arguments standing in implicit conflict, has been shown to be beneficial for solvers in first experiments. The question whether such an explication is possible in general, without changing the extensions, led to the *explicit conflict conjecture* [30], claiming that this is always possible for stable semantics (without the use of additional arguments), which we refuted in [31]. There, also the impact of rejected arguments on realizability was studied by characterizing *analytic signatures*:

$$\Sigma^{\sigma}_{\mathrm{XAF}} = \{\sigma(F) \mid F \text{ is an AF without implicit conflicts}\}.$$

Another aspect of realizability was tackled by studying *two-dimensional signatures* [103]. The two-dimensional signature of semantics σ and τ is the following collection of pairs:

$$\Sigma^{\sigma,\tau}_{\mathrm{AF}} = \{\langle\sigma(F), \tau(F)\rangle \mid F \text{ is an AF}\}.$$

Characterizations of these signatures give a deeper understanding of independence between semantics within a single AF. That is, whether the subset-relations presented in Proposition 2 are exhaustive or if there are other, more restrictive conditions on the relations between the sets of extensions obtained by the two semantics. Containment in the general signature as presented in Section 3.2 is always a necessary condition for containment in the two-dimensional signature: for any $\langle\mathbb{S}, \mathbb{T}\rangle \in \Sigma^{\sigma,\tau}_{\mathrm{AF}}$ it holds that $\mathbb{S} \in \Sigma^{\sigma}_{\mathrm{AF}}$ and $\mathbb{T} \in \Sigma^{\tau}_{\mathrm{AF}}$.

Realizability was put into action by Niskanen et al. [160] with an implementation of a slightly more general problem: the *synthesis* of AFs from examples. There, examples are, inspired from learning, either sets of arguments that shall be extensions, or sets of arguments that must not be extensions. The system then constructs an AF adhering to as many of these examples as possible.

In the work on the expressiveness of two-valued semantics of ADFs, Strass [189, 188] also considered the *representational succinctness* of ADFs, following previous work on expressiveness in knowledge representation [131]. That is, informally, how efficiently, in terms of size of the realizing ADF, interpretation-sets can be expressed. A question related to that, which is subject to future work, is whether the full expressive power of AFs can be achieved with polynomially many additional arguments, i.e. whether for each $\mathbb{S} \in \Sigma^{\sigma}_{\mathrm{AF}}$ there exists a realizing AF F ($\sigma(F) = \mathbb{S}$) such that $|A_F|$ is polynomial in $|Args_{\mathbb{S}}|$.

Dynamics. As indicated in the introduction, there has been a substantial amount of research in the dynamics of argumentation frameworks and the problems investigated and approaches that have been developed to address these differ considerably. In the following we describe those studies related to the revision of AFs, the problem we considered in Section 4.2, in more detail. Most of these works study revision of AFs in more restricted scenarios or from slightly different perspectives. Also, no general results regarding the complexity of revision of AFs are presented.

The focus of the work on *enforcing* by Baumann [20] is on the issue of modifying an AF such that a certain subset of arguments is contained in some extension (with respect to a semantics of interest) and, if so, what the number of minimal modifications is. This problem has also been approached by two implementations [78, 198]. On the other hand, Kontarinis et al. [142] propose a strategy in terms of rewriting rules to compute the minimal number of modifications to the attack relation of an AF to enforce a desired acceptance status of an argument. Booth et al. [51] give an AGM-like characterization of revision of AFs when certain logical constraints (expressing beliefs regarding the labellings of the AFs) are strengthened in order to incorporate newly held beliefs. But the focus is on determining certain fall back beliefs when the newly held beliefs are inconsistent with those held previously. How to compute the fall back beliefs is developed in detail for the complete semantics.

The work that was most influential to our work was the one on revision of AFs by Coste-Marquis et al. [76]. There, the authors are also aiming for AGM-style representation results for revision of AFs under minimal change in the extensions. However, the main difference between the work by Coste-Marquis et al. [76] and our approach are that (i) we consider the revising knowledge base to be an AF whereas it is in the form of a propositional formula in [76], and (ii) we require the revision to produce a single AF instead of a set of AF as the result. The price we pay is that in our solution the revised AF may have new arguments, while Coste-Marquis et al. [76] only need to modify the attack relation.

Baumann and Brewka [25] develop a monotonic logic ("Dung logic", based on the notion of strong equivalence [165]) to formalize reasoning about the dynamics of AFs under the different semantics, and rephrase the AGM postulates in this context. This approach is hence closer to the work of Delgrande et al. [85] on the AGM-revision of logic programs under the answer set semantics, which makes use of a standard monotonic model theory (based on the notion of SE-models [195]) for logic programs. Since, as they show, standard distance based revision operators do not work in their context, they develop an alternative syntactic-based revision operator returning a unique AF for the stable semantics. They also provide ideas for revision operators based on selection functions from a set of possible AFs for the other AF semantics they consider.

Moguillansky [156] developed a theory of remainder sets for abstract argumentation, thus yielding a more syntax-based approach to belief change in argumentation. Revision is defined via expansion and contraction and a representation result for the basic postulates of success, consistency, inclusion, vacuity, and core-retainment is shown. However, the postulates are reformulated with respect to acceptance of an argument rather than with respect to sets of extensions as done in our work. A similar approach to ours, focused on postulates and representation results, which also highlights the subtleties of instantiating the output as a single AF, concerns merging AFs in the presence of integrity constraints [86].

In [83] a very general theory for modeling dynamics of AFs is proposed. This theory makes it possible to express how an agent who has beliefs in the form of an argumentation system can interact on a target argumentation system that may represent the state of knowledge at a given stage of a debate. Here AFs (and the dynamics of AFs) are encoded within the general, tailor-made first order language YALLA.

Models of dynamics in structured argumentation can be found in [157] and [181]. The former offers a model building on [156] while the latter is a model for one of the most prominent formalisms for structured argumentation, ASPIC+ [155].

Finally, another notable connection between belief revision and argumentation are studies where argumentation techniques are used as a tool to perform change on non-argumentation knowledge bases. As an example, we point to [143], where deductive argumentation is used for selecting the parts of the new information that shall be accepted.

Gaggl and Strass [120] presented a decomposition schema for ADFs proceeding along an ADF's strongly connected components. It turns out to use similar propagation techniques as our results on directional splitting (cf. Section 5.2.1). Only arguments mapped to the truth value **u** are treated slightly different. By the decomposition schema they provide alternative characterizations of the semantics, while the motivation of splitting is more of computational nature.

6.3 Future Work

The presented work offers various directions of future work.

The results on expressiveness in Section 3 left some gaps here and there by leaving certain semantics open. For instance, the exact characterization of the general signature as well as the maximum number of realizable extensions have not been clarified for complete and resolution-based grounded semantics yet and are among the open problems in abstract argumentation as listed in [27]. Moreover, the expressiveness of the cf2 semantics [13] has not been studied yet.

For input-output realizability as studied in Section 3.4, there are two main issues to be tackled: first, we are interested in the construction of I/O-modules from compact I/O-specifications where the function is not explicitly stated but, for instance, described as a Boolean (or three-valued) circuit. Second, we want to find ways of minimizing the size of I/O-modules, which is particularly interesting for the task of summarization [17], i.e. replacing sub-AFs by simpler ones without changing the semantics of the remaining AF. Moreover, I/O-realizability for ADFs is still to be studied.

We plan to extend the unifying approach to realizability to further formalisms as well as to other semantics, such as naive-based semantics [120, 123]. Due to the modular design of the algorithm presented in Section 3.6, one only has to develop suitable propagators to be used by the algorithm. Moreover, the expressiveness of the family of extension-based semantics for ADFs [167, 168] is an open issue.

A user-friendly variant of ADFs was introduced by Brewka and Woltran [54] in the form of a framework for "graph-based argument processing with patterns of acceptance" (GRAPPA). In such a GRAPPA framework, links are labelled and the acceptability of arguments is specified by acceptance patterns based on these labels, which makes them independent from the actual neighboring arguments. The improved usability is witnessed by an implementation with a graphical user interface [135] as well as a mobile application for online discussion [176]. Future work should clarify whether the results on ADFs presented in this work carry over to GRAPPA and how they are affected by restrictions of the pattern language.

For the use of results on expressiveness to improve solvers, a further step has been done by two-valued signatures [103], which are particularly interesting if AFs are evaluated under more than one semantics at a time. Still, our aim is to find (easy-to-use) techniques to prune search space when parts of the results have been already found. Characterizations of signatures are essential in this respect. Moreover, while we have shown that checking compactness of AFs is computationally expensive [31], non-exhaustive ways of determining rejected arguments are still a promising subject of future research.

Also in the field of revision there are several directions for future work. Fist, we want to extend our results for revision of AFs by AFs to semantics which are not proper I-maximal. Moreover, the application to AFs and ADFs still has to be figured out for other revision operators from the literature such as the operators by Borgida [52], Winslett [199], Forbus [119], and Satoh [180], to name a few. Meaningful revision operators will also have to take the syntactic form of the framework into account. Here, a possibility is a two-step approach, where our abstract revision is the first step. Based on this result, a second step would revise the syntactic structure of the framework. Finally, we plan to apply our findings to other belief change operations. In particular, iterated belief revision [81, 140, 49] seems to have natural applications in the argumentation domain and we believe that the understanding of revision yielding a single AF (resp. ADF) is fundamental for this purpose.

On a more general level, we want to analyze whether our insights can be extended to a broader theory of belief change within fragments. In particular, the hybrid approach presented in Section 4.3.3 follows a concept that is by no means limited to ADFs.

Finally, parametrized splitting for AFs as well as general splitting for ADFs is still an open issue for most semantics.

Bibliography

[1] Latifa Al-Abdulkarim, Katie Atkinson, and Trevor J. M. Bench-Capon. Abstract dialectical frameworks for legal reasoning. In Rinke Hoekstra, editor, *Proceedings of the 27th Annual Conference on Legal Knowledge and Information Systems (JURIX 2014)*, volume 271 of *Frontiers in Artificial Intelligence and Applications*, pages 61–70. IOS Press, 2014.

[2] 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.

[3] Latifa Al-Abdulkarim, Katie Atkinson, and Trevor J. M. Bench-Capon. Accommodating change. *Artif. Intell. Law*, 24(4):409–427, 2016.

[4] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: partial meet contraction and revision functions. *J. Symb. Log.*, 50 (2):510–530, 1985.

[5] Leila Amgoud and Claudette Cayrol. On the acceptability of arguments in preference-based argumentation. In Gregory F. Cooper and Serafín Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI 1998)*, pages 1–7. Morgan Kaufmann, 1998.

[6] Leila Amgoud and Henri Prade. Using arguments for making and explaining decisions. *Artif. Intell.*, 173(3-4):413–436, 2009.

[7] Leila Amgoud and Srdjan Vesic. A new approach for preference-based argumentation frameworks. *Ann. Math. Artif. Intell.*, 63(2):149–183, 2011.

[8] Ofer Arieli. Distance-based paraconsistent logics. *Int. J. Approx. Reasoning*, 48(3): 766–783, 2008.

[9] Ofer Arieli. Conflict-tolerant semantics for argumentation frameworks. In Luis Fariñas del Cerro, Andreas Herzig, and Jérôme Mengin, editors, *Proceedings of the 13th European Conference on Logics in Artificial Intelligence (JELIA 2012)*, volume 7519 of *Lecture Notes in Computer Science*, pages 28–40. Springer, 2012.

[10] Sanjeev Arora and Boaz Barak. *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009. ISBN 978-0-521-42426-4.

[11] Pietro Baroni and Massimiliano Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artif. Intell.*, 171(10-15):675–700, 2007.

[12] Pietro Baroni and Massimiliano Giacomin. A systematic classification of argumentation frameworks where semantics agree. In Philippe Besnard, Sylvie Doutre, and Anthony Hunter, editors, *Proceedings of the 2nd International Conference on Computational Models of Argument (COMMA 2008)*, volume 172 of *Frontiers in Artificial Intelligence and Applications*, pages 37–48. IOS Press, 2008.

[13] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-recursiveness: a general schema for argumentation semantics. *Artif. Intell.*, 168(1-2):162–210, 2005.

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

[15] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. AFRA: Argumentation framework with recursive attacks. *Int. J. Approx. Reasoning*, 52(1):19–37, 2011.

[16] Pietro Baroni, Paul E. Dunne, and Massimiliano Giacomin. On the resolution-based family of abstract argumentation semantics and its grounded instance. *Artif. Intell.*, 175(3-4):791–813, 2011.

[17] Pietro Baroni, Guido Boella, Federico Cerutti, Massimiliano Giacomin, Leendert W. N. van der Torre, and Serena Villata. On the input/output behavior of argumentation frameworks. *Artif. Intell.*, 217:144–197, 2014.

[18] Pietro Baroni, Massimiliano Giacomin, and Beishui Liao. On topology-related properties of abstract argumentation semantics. A correction and extension to dynamics of argumentation systems: A division-based method. *Artif. Intell.*, 212:104–115, 2014.

[19] Ringo Baumann. Splitting an argumentation framework. In James P. Delgrande and Wolfgang Faber, editors, *Proceedings of the 11th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2011)*, volume 6645 of *Lecture Notes in Computer Science*, pages 40–53. Springer, 2011.

[20] Ringo Baumann. What does it take to enforce an argument? Minimal change in abstract argumentation. In Luc De Raedt, Christian Bessière, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter J. F. Lucas, editors, *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 127–132. IOS Press, 2012.

[21] Ringo Baumann. Context-free and context-sensitive kernels: update and deletion equivalence in abstract argumentation. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 63–68. IOS Press, 2014.

[22] Ringo Baumann. *Metalogical Contributions to the Nonmonotonic Theory of Abstract Argumentation.* PhD thesis, Leipzig University, 2014.

[23] Ringo Baumann and Gerhard Brewka. Expanding argumentation frameworks: Enforcing and monotonicity results. In Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Guillermo R. Simari, editors, *Computational Models of Argument: Proceedings of COMMA 2010*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 75–86. IOS Press, 2010.

[24] Ringo Baumann and Gerhard Brewka. Spectra in abstract argumentation: an analysis of minimal change. In Pedro Cabalar and Tran Cao Son, editors, *Proceedings of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2013)*, volume 8148 of *Lecture Notes in Computer Science*, pages 174–186. Springer, 2013.

[25] Ringo Baumann and Gerhard Brewka. AGM meets abstract argumentation: Expansion and revision for dung frameworks. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2734–2740. AAAI Press, 2015.

[26] Ringo Baumann and Hannes Strass. On the maximal and average numbers of stable extensions. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Theory and Applications of Formal Argumentation – 2nd International Workshop (TAFA 2013), Revised Selected papers*, volume 8306 of *Lecture Notes in Computer Science*, pages 111–126. Springer, 2014.

[27] Ringo Baumann and Hannes Strass. Open problems in abstract argumentation. 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 325–339. Springer, 2015.

[28] Ringo Baumann, Gerhard Brewka, and Renata Wong. Splitting argumentation frameworks: An empirical evaluation. In Sanjay Modgil, Nir Oren, and Francesca Toni, editors, *Theory and Applications of Formal Argumentation – 1st International Workshop (TAFA 2011), Revised Selected Papers*, volume 7132 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2011.

[29] Ringo Baumann, Gerhard Brewka, Wolfgang Dvořák, and Stefan Woltran. Parameterized splitting: A simple modification-based approach. In Esra Erdem, Joohyung Lee, Yuliya Lierler, and David Pearce, editors, *Correct Reasoning – Essays on Logic-Based AI in Honour of Vladimir Lifschitz*, volume 7265 of *Lecture Notes in Computer Science*, pages 57–71. Springer, 2012.

[30] Ringo Baumann, Wolfgang Dvořák, Thomas Linsbichler, Hannes Strass, and Stefan Woltran. Compact Argumentation Frameworks. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 69–74. IOS Press, 2014.

[31] Ringo Baumann, Wolfgang Dvořák, Thomas Linsbichler, Christof Spanring, Hannes Strass, and Stefan Woltran. On rejected arguments and implicit conflicts: The hidden power of argumentation semantics. *Artif. Intell.*, 241:244–284, 2016.

[32] Ringo Baumann, Thomas Linsbichler, and Stefan Woltran. Verifiability of argumentation semantics. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Proceedings of the 6th International Conference on Computational Models of Argument (COMMA 2016)*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 83–94. IOS Press, 2016.

[33] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.

[34] Trevor J. M. Bench-Capon, Henry Prakken, and Giovanni Sartor. Argumentation in legal reasoning. In Guillermo R. Simari and Iyad Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 363–382. Springer, 2009.

[35] Jamal Bentahar, Bernard Moulin, and Micheline Bélanger. A taxonomy of argumentation models used for knowledge representation. *Artif. Intell. Rev.*, 33(3): 211–259, 2010.

[36] Philippe Besnard and Sylvie Doutre. Checking the acceptability of a set of arguments. In James P. Delgrande and Torsten Schaub, editors, *Proceedings of the 10th International Workshop on Non-Monotonic Reasoning (NMR 2004)*, pages 59–64, 2004.

[37] Philippe Besnard and Sylvie Doutre. Characterization of semantics for argument systems. In Didier Dubois, Christopher A. Welty, and Mary-Anne Williams, editors, *Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 183–193. AAAI Press, 2004.

[38] Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. *Artif. Intell.*, 128(1-2):203–235, 2001.

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

[40] Pierre Bisquert, Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. Change in argumentation systems: exploring the interest of removing an argument. In Salem Benferhat and John Grant, editors, *Proceedings of the 5th International Conference on Scalable Uncertainty Management (SUM 2011)*, volume 6929 of *Lecture Notes in Computer Science*, pages 275–288. Springer, 2011.

[41] Pierre Bisquert, Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. Enforcement in argumentation is a kind of update. In Weiru Liu, V. S. Subrahmanian, and Jef Wijsen, editors, *Proceedings of the 7th International Conference on Scalable Uncertainty Management (SUM 2013)*, volume 8078 of *Lecture Notes in Computer Science*, pages 30–43. Springer, 2013.

[42] Stefano Bistarelli and Francesco Santini. Conarg: A constraint-based computational framework for argumentation systems. In Taghi M. Khoshgoftaar and Xingquan Zhu, editors, *Proceedings of the IEEE 23rd International Conference on Tools with Artificial Intelligence (ICTAI 2011)*, pages 605–612. IEEE Computer Society, 2011.

[43] Stefano Bistarelli, Fabio Rossi, and Francesco Santini. Conarg: A tool for classical and weighted argumentation. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Proceedings of the 6th International Conference on Computational Models of Argument (COMMA 2016)*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 463–464. IOS Press, 2016.

[44] Alexander Bochman. Abstract dialectical argumentation among close relatives. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Proceedings of the 6th International Conference on Computational Models of Argument (COMMA 2016)*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 127–138. IOS Press, 2016.

[45] Guido Boella, Souhila Kaci, and Leendert van der Torre. Dynamics in argumentation with single extensions: attack refinement and the grounded extension (extended version). In Peter McBurney, Iyad Rahwan, Simon Parsons, and Nicolas Maudet, editors, *Proceedings of the 6th International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2009), Revised Selected and Invited Papers*, volume 6057 of *Lecture Notes in Computer Science*, pages 150–159. Springer, 2009.

[46] 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 R. Simari, editors, *Computational Models of Argument: Proceedings of COMMA 2010*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 111–122. IOS Press, 2010.

[47] Andrei Bondarenko, Francesca Toni, and Robert A. Kowalski. An assumption-based framework for non-monotonic reasoning. In Luís Moniz Pereira and Anil Nerode, editors, *Proceedings of the 2nd International Workshop on Logic Programming and Non-monotonic Reasoning*, pages 171–189. MIT Press, 1993.

[48] 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, 2015.

[49] Richard Booth and Thomas Andreas Meyer. Admissible and restrained revision. *J. Artif. Intell. Res. (JAIR)*, 26:127–151, 2006.

[50] Richard Booth, Martin Caminada, Mikolaj Podlaszewski, and Iyad Rahwan. Quantifying disagreement in argument-based reasoning. In Wiebe van der Hoek, Lin Padgham, Vincent Conitzer, and Michael Winikoff, editors, *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 493–500. IFAAMAS, 2012.

[51] Richard Booth, Souhila Kaci, Tjitze Rienstra, and Leendert van der Torre. A logical theory about dynamics in abstract argumentation. In Weiru Liu, V. S. Subrahmanian, and Jef Wijsen, editors, *Proceedings of the 7th International Conference on Scalable Uncertainty Management (SUM 2013)*, volume 8078 of *Lecture Notes in Computer Science*, pages 148–161. Springer, 2013.

[52] Alexander Borgida. Language features for flexible handling of exceptions in information systems. *ACM Trans. Database Syst.*, 10(4):565–603, 1985.

[53] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR 2010)*, pages 102–111. AAAI Press, 2010.

[54] Gerhard Brewka and Stefan Woltran. GRAPPA: A semantical framework for graph-based argument processing. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 153–158. IOS Press, 2014.

[55] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes P. 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.

[56] Gerhard Brewka, Sylwia Polberg, and Stefan Woltran. Generalizations of Dung frameworks and their role in formal argumentation. *IEEE Intell. Syst.*, 29(1):30–38, 2014.

[57] 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 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*, pages 1077–1083. AAAI Press, 2017.

[58] Remi Brochenin, Thomas Linsbichler, Marco Maratea, Johannes P. Wallner, and Stefan Woltran. Abstract solvers for Dung's argumentation frameworks. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Theory and Applications of Formal Argumentation – 3rd International Workshop (TAFA 2015), Revised Selected Papers*, volume 9524 of *Lecture Notes in Computer Science*, pages 40–58. Springer, 2015.

[59] Katarzyna Budzynska, Mathilde Janier, Juyeon Kang, Chris Reed, Patrick Saint-Dizier, Manfred Stede, and Olena Yaskorska. Towards argument mining from dialogue. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 185–196. IOS Press, 2014.

[60] Elena Cabrio and Serena Villata. Abstract dialectical frameworks for text exploration. In Jaap van den Herik and Joaquim Filipe, editors, *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016)*, pages 85–95. SciTePress, 2016.

[61] Martin Caminada. Comparing two unique extension semantics for formal argumentation: Ideal and eager. In *Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2007)*, pages 81–87, 2007.

[62] Martin Caminada. Strong admissibility revisited. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 197–208. IOS Press, 2014.

[63] Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.

[64] Martin Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2):109–145, 2009. ISSN 0039-3215.

[65] Martin Caminada, Walter Alexandre Carnielli, and Paul E. Dunne. Semi-stable semantics. *J. Log. Comput.*, 22(5):1207–1254, 2012.

[66] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In Lluis Godo, editor, *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, volume 3571 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2005.

[67] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Bipolarity in argumentation graphs: Towards a better understanding. *Int. J. Approx. Reasoning*, 54(7): 876–899, 2013.

[68] Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. Change in abstract argumentation frameworks: adding an argument. *J. Artif. Intell. Res. (JAIR)*, 38:49–84, 2010.

[69] Federico Cerutti, Massimiliano Giacomin, and Mauro Vallati. Argsemsat: Solving argumentation problems using SAT. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 455–456. IOS Press, 2014.

[70] Federico Cerutti, Mauro Vallati, and Massimiliano Giacomin. Where are we now? State of the art and future trends of solvers for hard argumentation problems. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Proceedings of the 6th International Conference on Computational Models of Argument (COMMA 2016)*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 207–218. IOS Press, 2016.

[71] Günther Charwat, Wolfgang Dvořák, Sarah A. Gaggl, Johannes P. Wallner, and Stefan Woltran. Methods for solving reasoning problems in abstract argumentation – A survey. *Artif. Intell.*, 220:28–63, 2015.

[72] Andrea Cohen, Alejandro J. García, and Guillermo R. Simari. A structured argumentation system with backing and undercutting. *Eng. Appl. of AI*, 49: 149–166, 2016.

[73] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Symmetric argumentation frameworks. In Lluis Godo, editor, *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, volume 3571 of *Lecture Notes in Computer Science*, pages 317–328. Springer, 2005.

[74] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Prudent semantics for argumentation frameworks. In Andrew Lim, editor, *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2005)*, pages 568–572. IEEE Computer Society, 2005.

[75] Sylvie Coste-Marquis, Sébastien Konieczny, Pierre Marquis, and Mohand Akli Ouali. Weighted attacks in argumentation frameworks. In Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, editors, *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR 2012)*, pages 593–597. AAAI Press, 2012.

[76] Sylvie Coste-Marquis, Sébastien Konieczny, Jean-Guy Mailly, and Pierre Marquis. On the revision of argumentation systems: minimal change of arguments statuses. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 72–81. AAAI Press, 2014.

[77] Sylvie Coste-Marquis, Sébastien Konieczny, Jean-Guy Mailly, and Pierre Marquis. A translation-based approach for revision of argumentation frameworks. In Eduardo Fermé and João Leite, editors, *Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA 2014)*, volume 8761 of *Lecture Notes in Computer Science*, pages 397–411. Springer, 2014.

[78] Sylvie Coste-Marquis, Sébastien Konieczny, Jean-Guy Mailly, and Pierre Marquis. Extension enforcement in abstract argumentation as an optimization problem. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2876–2882. AAAI Press, 2015.

[79] Kristijonas Cyras and Francesca Toni. ABA+: assumption-based argumentation with preferences. In Chitta Baral, James P. Delgrande, and Frank Wolter, editors, *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*, pages 553–556. AAAI Press, 2016.

[80] Mukesh Dalal. Investigations into a theory of knowledge base revision. In Howard E. Shrobe, Tom M. Mitchell, and Reid G. Smith, editors, *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI 1988)*, pages 475–479. AAAI Press / The MIT Press, 1988.

[81] Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. *Artif. Intell.*, 89(1-2):1–29, 1997.

[82] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, 1962.

[83] Florence Dupin de Saint-Cyr, Pierre Bisquert, Claudette Cayrol, and Marie-Christine Lagasquie-Schiex. Argumentation update in YALLA (Yet Another Logic Language for Argumentation). *Int. J. Approx. Reasoning*, 75:57–92, 2016.

[84] James P. Delgrande and Pavlos Peppas. Belief revision in Horn theories. *Artif. Intell.*, 218:1–22, 2015.

[85] James P. Delgrande, Pavlos Peppas, and Stefan Woltran. AGM-style belief revision of logic programs under answer set semantics. In Pedro Cabalar and Tran Cao Son, editors, *Proceedings of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2013)*, volume 8148 of *Lecture Notes in Computer Science*, pages 264–276. Springer, 2013.

[86] Jérôme Delobelle, Adrian Haret, Sébastien Konieczny, Jean-Guy Mailly, Julien Rossit, and Stefan Woltran. Merging of abstract argumentation frameworks. In Chitta Baral, James P. Delgrande, and Frank Wolter, editors, *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*, pages 33–42. AAAI Press, 2016.

[87] Marc Denecker, Victor W. Marek, and Miroslaw Truszczynski. Ultimate approximation and its application in nonmonotonic knowledge representation systems. *Inf. Comput.*, 192(1):84–121, 2004.

[88] Martin Diller, Johannes P. Wallner, and Stefan Woltran. Reasoning in abstract dialectical frameworks using quantified boolean formulas. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 241–252. IOS Press, 2014.

[89] Martin Diller, Adrian Haret, Thomas Linsbichler, Stefan Rümmele, and Stefan Woltran. An extension-based approach to belief revision in abstract argumentation. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2926–2932. AAAI Press, 2015.

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

[91] Martin Diller, Adrian Haret, Thomas Linsbichler, Stefan Rümmele, and Stefan Woltran. An extension-based approach to belief revision in abstract argumentation. Technical Report DBAI-TR-2017-106, Technische Universität Wien, 2017. URL `http://www.dbai.tuwien.ac.at/research/report/dbai-tr-2017-106.pdf`.

[92] Yannis Dimopoulos and Alberto Torres. Graph theoretical structures in logic programs and default theories. *Theor. Comput. Sci.*, 170(1-2):209–244, 1996.

[93] Sylvie Doutre, Andreas Herzig, and Laurent Perrussel. A dynamic logic framework for abstract argumentation. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 62–71. AAAI Press, 2014.

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

[95] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. *Artif. Intell.*, 171(10-15):642–674, 2007.

[96] Paul E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artif. Intell.*, 171(10-15):701–729, 2007.

[97] Paul E. Dunne and Trevor J. M. Bench-Capon. Coherence in finite argument systems. *Artif. Intell.*, 141(1/2):187–203, 2002.

[98] Paul E. Dunne, Anthony Hunter, Peter McBurney, Simon Parsons, and Michael Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artif. Intell.*, 175(2):457–486, 2011.

[99] Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. In Christoph Beierle and Gabriele Kern-Isberner, editors, *Proceedings of the 4th Workshop on Dynamics of Knowledge and Belief (DKB 2013)*, pages 16–30, 2013.

[100] Paul E. Dunne, Wolfgang Dvořák, and Stefan Woltran. Parametric properties of ideal semantics. *Artif. Intell.*, 202:1–28, 2013.

[101] Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 72–81. AAAI Press, 2014.

[102] Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. *Artif. Intell.*, 228: 153–178, 2015.

[103] Paul E. Dunne, Christof Spanring, Thomas Linsbichler, and Stefan Woltran. Investigating the relationship between argumentation semantics via signatures. In Subbarao Kambhampati, editor, *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pages 1051–1057. IJCAI/AAAI Press, 2016.

[104] Wolfgang Dvořák, Matti Järvisalo, Johannes P. Wallner, and Stefan Woltran. Complexity-sensitive decision procedures for abstract argumentation. *Artif. Intell.*, 206:53–78, 2014.

[105] Wolfgang Dvořák. *Computational Aspects of Abstract Argumentation*. PhD thesis, Vienna University of Technology, 2012.

[106] Wolfgang Dvořák and Sarah A. Gaggl. Stage semantics and the SCC-recursive schema for argumentation semantics. *J. Log. Comput.*, 26(4):1149–1202, 2016.

[107] Wolfgang Dvořák and Christof Spanring. Comparing the expressiveness of argumentation semantics. *J. Log. Comput.*, 2016. In Press. Available online at http://dx.doi.org/10.1093/logcom/exw008.

[108] Wolfgang Dvořák and Stefan Woltran. Complexity of semi-stable and stage semantics in argumentation frameworks. *Inf. Process. Lett.*, 110(11):425–430, 2010.

[109] Wolfgang Dvořák and Stefan Woltran. On the intertranslatability of argumentation semantics. *J. Artif. Intell. Res. (JAIR)*, 41:445–475, 2011.

[110] Wolfgang Dvořák, Thomas Linsbichler, Emilia Oikarinen, and Stefan Woltran. Resolution-based grounded semantics revisited. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 269–280. IOS Press, 2014.

[111] Wolfgang Dvořák, Sarah A. Gaggl, Thomas Linsbichler, and Johannes P. Wallner. Reduction-based approaches to implement Modgil's Extended Argumentation 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 249–264. Springer, 2015.

[112] Sjur K. Dyrkolbotn. How to argue for anything: Enforcing arbitrary sets of labellings using afs. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 626–629. AAAI Press, 2014.

[113] Uwe Egly, Sarah A. Gaggl, and Stefan Woltran. Answer-set programming encodings for argumentation frameworks. *Argument & Computation*, 1(2):147–177, 2010.

[114] Thomas Eiter and Georg Gottlob. The complexity of logic-based abduction. *Journal of the ACM*, 42(1):3–42, 1995.

[115] Thomas Eiter, Georg Gottlob, and Heikki Mannila. Disjunctive datalog. *ACM Trans. Database Syst.*, 22(3):364–418, 1997.

[116] Stefan Ellmauthaler and Hannes Strass. The DIAMOND system for computing with abstract dialectical frameworks. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 233–240. IOS Press, 2014.

[117] Stefan Ellmauthaler and Hannes Strass. DIAMOND 3.0 - A native C++ implementation of DIAMOND. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Proceedings of the 6th International Conference on Computational Models of Argument (COMMA 2016)*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 471–472. IOS Press, 2016.

[118] Herbert B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 1972. ISBN 978-0-12-238450-9.

[119] Kenneth D. Forbus. Introducing actions into qualitative simulation. In N. S. Sridharan, editor, *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI 1989)*, pages 1273–1278. Morgan Kaufmann, 1989.

[120] Sarah A. Gaggl and Hannes Strass. Decomposing abstract dialectical frameworks. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 281–292. IOS Press, 2014.

[121] Sarah A. Gaggl and Stefan Woltran. The cf2 argumentation semantics revisited. *J. Log. Comput.*, 23(5):925–949, 2013.

[122] Sarah A. Gaggl, Norbert Manthey, Alessandro Ronca, Johannes P. Wallner, and Stefan Woltran. Improved answer-set programming encodings for abstract argumentation. *Theory and Practice of Logic Programming*, 15(4–5):434–448, 2015.

[123] Sarah A. Gaggl, Sebastian Rudolph, and Hannes Strass. On the computational complexity of naive-based semantics for abstract dialectical frameworks. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2985–2991. AAAI Press, 2015.

[124] Sarah A. Gaggl, Thomas Linsbichler, Marco Maratea, and Stefan Woltran. Introducing the second international competition on computational models of argumentation. In Matthias Thimm, Federico Cerutti, Hannes Strass, and Mauro Vallati, editors, *Proceedings of the 1st International Workshop on Systems and Algorithms for Formal Argumentation (SAFA 2016)*, volume 1672 of *CEUR Workshop Proceedings*, pages 4–9. CEUR-WS.org, 2016.

[125] Peter Gärdenfors. Epistemic importance and minimal changes of belief. *Australasian J. Philos.*, 62:136–157, 1984.

[126] Peter Gärdenfors. *Knowledge in flux. Modeling the dynamics of epistemic states*. Cambridge: MIT Press, 1988. ISBN 1-904-98789-3.

[127] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2012.

[128] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Clingo = ASP + control: Preliminary report. In Michael Leuschel and Tom Schrijvers, editors, *Technical Communications of the 30th International Conference on Logic Programming (ICLP 2014)*, volume 14(4-5), 2014.

[129] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In Robert A. Kowalski and Kenneth A. Bowen, editors, *Proceedings of the 5th International Conference and Symposium on Logic Programming*, pages 1070–1080. MIT Press, 1988.

[130] Massimiliano Giacomin, Thomas Linsbichler, and Stefan Woltran. On the functional completeness of argumentation semantics. In Chitta Baral, James P. Delgrande, and Frank Wolter, editors, *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*, pages 43–52. AAAI Press, 2016.

[131] Goran Gogic, Henry Kautz, Christos Papadimitriou, and Bart Selman. The comparative linguistics of knowledge representation. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995)*, pages 862–869. Morgan Kaufmann, 1995.

[132] Carla P. Gomes, Henry A. Kautz, Ashish Sabharwal, and Bart Selman. Satisfiability Solvers. In Frank van Harmelen, Vladimir Lifschitz, and Bruce W. Porter, editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 89–134. Elsevier, 2008.

[133] Davide Grossi and Sanjay Modgil. On the graded acceptability of arguments. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 868–874. AAAI Press, 2015.

[134] Richard W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

[135] Georg Heissenberger and Stefan Woltran. GrappaVis – A system for advanced graph-based argumentation. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Proceedings of the 6th International Conference on Computational Models of Argument (COMMA 2016)*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 473–474. IOS Press, 2016.

[136] Anthony Hunter. Computational persuasion with applications in behaviour change. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Proceedings of the 6th International Conference on Computational Models of Argument (COMMA 2016)*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 5–18. IOS Press, 2016.

[137] Anthony Hunter and Matthew Williams. Aggregation of clinical evidence using argumentation: A tutorial introduction. In Arjen Hommersom and Peter J. F. Lucas, editors, *Foundations of Biomedical Knowledge Representation – Methods and Applications*, volume 9521 of *Lecture Notes in Computer Science*, pages 317–337. Springer, 2015.

[138] Hadassa Jakobovits and Dirk Vermeir. Robust semantics for argumentation frameworks. *J. Log. Comput.*, 9(2):215–261, 1999.

[139] Tomi Janhunen, Emilia Oikarinen, Hans Tompits, and Stefan Woltran. Modularity aspects of disjunctive stable models. *J. Artif. Intell. Res. (JAIR)*, 35:813–857, 2009.

[140] Yi Jin and Michael Thielscher. Iterated belief revision, revised. *Artif. Intell.*, 171 (1):1–18, 2007.

[141] Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artif. Intell.*, 52(3):263–294, 1991.

[142] Dionysios Kontarinis, Elise Bonzon, Nicolas Maudet, Alan Perotti, Leon van der Torre, and Serena Villata. Rewriting rules for the computation of goal-oriented changes in an argumentation system. In João Leite, Tran Cao Son, Paolo Torroni, Leon van der Torre, and Stefan Woltran, editors, *Proceedings of the 14th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA XIV)*, volume 8143 of *Lecture Notes in Computer Science*, pages 51–68. Springer, 2013.

[143] Patrick Krümpelmann, Matthias Thimm, Marcelo A. Falappa, Alejandro J. García, Gabriele Kern-Isberner, and Guillermo R. Simari. Selective revision by deductive argumentation. In Sanjay Modgil, Nir Oren, and Francesca Toni, editors, *Theory and Applications of Formal Argumentation – 1st International Workshop (TAFA 2011), Revised Selected Papers*, volume 7132 of *Lecture Notes in Computer Science*, pages 147–162. Springer, 2012.

[144] Paolo Liberatore and Marco Schaerf. Belief revision and update: Complexity of model checking. *J. Comput. Syst. Sci.*, 62(1):43–72, 2001.

[145] Vladimir Lifschitz and Hudson Turner. Splitting a logic program. In Pascal Van Hentenryck, editor, *Proceedings of the 11th International Conference on Logic Programming (ICLP 1994)*, pages 23–37. MIT Press, 1994.

[146] Thomas Linsbichler. On the limits of expressiveness in abstract argumentation semantics: Realizability and Signatures. Master's thesis, TU Wien, 2013.

[147] Thomas Linsbichler. Splitting Abstract Dialectical Frameworks. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 357–368. IOS Press, 2014.

[148] Thomas Linsbichler and Stefan Woltran. Revision of abstract dialectical frameworks: Preliminary report. In Sarah Gaggl, Juan Carlos Nieves, and Hannes Strass, editors, *Proceedings of the 1st international Workshop on Argumentation in Logic Programming and Non-Monotonic Reasoning (Arg-LPNMR 2016)*, 2016.

[149] Thomas Linsbichler, Christof Spanring, and Stefan Woltran. The hidden power of abstract argumentation semantics. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Theory and Applications of Formal Argumentation – 3rd International Workshop (TAFA 2015), Revised Selected Papers*, volume 9524 of *Lecture Notes in Computer Science*, pages 146–162. Springer, 2015.

[150] Thomas Linsbichler, Jörg Pührer, and Hannes Strass. A uniform account of realizability in abstract argumentation. In Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen, editors, *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 252–260. IOS Press, 2016.

[151] Victor W. Marek and Mirosław Truszczyński. Stable models and an alternative logic programming paradigm. In Krzysztof R. Apt, Victor W. Marek, Mirosław Truszczyński, and David S. Warren, editors, *In The Logic Programming Paradigm: a 25-Year Perspective*, pages 375–398. Springer, 1999.

[152] Peter McBurney, Simon Parsons, and Iyad Rahwan, editors. *Argumentation in Multi-Agent Systems – 8th International Workshop, ArgMAS 2011, Revised Selected Papers*, volume 7543 of *Lecture Notes in Computer Science*, 2012. Springer. ISBN 978-3-642-33151-0.

[153] Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artif. Intell.*, 173(9-10):901–934, 2009.

[154] Sanjay Modgil and Trevor J. M. Bench-Capon. Metalevel argumentation. *J. Log. Comput.*, 21(6):959–1003, 2011.

[155] Sanjay Modgil and Henry Prakken. The $ASPIC^+$ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[156] Martín O. Moguillansky. A study of argument acceptability dynamics through core and remainder sets. In Marc Gyssens and Guillermo R. Simari, editors, *Proceedings of the 9th International Symposium on Foundations of Information and Knowledge Systems (FoIKS 2016)*, volume 9616 of *Lecture Notes in Computer Science*, pages 3–23. Springer, 2016.

[157] Martín O. Moguillansky and Guillermo R. Simari. A specialized set theoretic semantics for acceptability dynamics of arguments. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Proceedings of the 6th International Conference on Computational Models of Argument (COMMA 2016)*,

volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 391–402. IOS Press, 2016.

[158] Søren Holbech Nielsen and Simon Parsons. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In Nicolas Maudet, Simon Parsons, and Iyad Rahwan, editors, *Proceedings of the 3d International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006)*, volume 4766 of *Lecture Notes in Computer Science*, pages 54–73. Springer, 2006.

[159] Ilkka Niemelä. Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence*, 25(3–4): 241–273, 1999.

[160] Andreas Niskanen, Johannes P. Wallner, and Matti Järvisalo. Synthesizing argumentation frameworks from examples. In Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen, editors, *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 551–559. IOS Press, 2016.

[161] Andreas Niskanen, Johannes P. Wallner, and Matti Järvisalo. Optimal status enforcement in abstract argumentation. In Subbarao Kambhampati, editor, *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pages 1216–1222. IJCAI/AAAI Press, 2016.

[162] Samer Nofal, Katie Atkinson, and Paul E. Dunne. Algorithms for decision problems in argument systems under preferred semantics. *Artif. Intell.*, 207:23–51, 2014.

[163] Farid Nouioua. AFs with necessities: Further semantics and labelling characterization. In Weiru Liu, V. S. Subrahmanian, and Jef Wijsen, editors, *Proceedings of the 7th International Conference on Scalable Uncertainty Management (SUM 2013)*, volume 8078 of *Lecture Notes in Computer Science*, pages 120–133. Springer, 2013.

[164] Farid Nouioua and Éric Würbel. Removed set-based revision of abstract argumentation frameworks. In *Proceedings of the 26th International Conference on Tools with Artificial Intelligence (ICTAI 2014)*, pages 784–791. IEEE Computer Society, 2014.

[165] Emilia Oikarinen and Stefan Woltran. Characterizing strong equivalence for argumentation frameworks. *Artif. Intell.*, 175(14-15):1985–2009, 2011.

[166] Christos H. Papadimitriou. *Computational complexity.* Addison-Wesley, 1994. ISBN 978-0-201-53082-7.

[167] Sylwia Polberg. Extension-based semantics of abstract dialectical frameworks. In Ulle Endriss and João Leite, editors, *Proceedings of the 7th European Starting AI Researcher Symposium (STAIRS 2014)*, volume 264 of *Frontiers in Artificial Intelligence and Applications*, pages 240–249. IOS Press, 2014.

[168] Sylwia Polberg. Revisiting extension-based semantics of abstract dialectical frameworks. Technical Report DBAI-TR-2015-88, Technische Universität Wien, 2015. URL http://www.dbai.tuwien.ac.at/research/report/dbai-tr-2015-88.pdf.

[169] Sylwia Polberg. Understanding the abstract dialectical framework. In Loizos Michael and Antonis C. Kakas, editors, *Proceedings of the 15th European Conference on Logics in Artificial Intelligence (JELIA 2016)*, volume 10021 of *Lecture Notes in Computer Science*, pages 430–446, 2016.

[170] Sylwia Polberg. Intertranslatability of abstract argumentation frameworks. Technical Report DBAI-TR-2017-104, Technische Universität Wien, 2017. URL http://www.dbai.tuwien.ac.at/research/report/dbai-tr-2017-104.pdf.

[171] Sylwia Polberg and Nir Oren. Revisiting support in abstract argumentation systems. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 369–376. IOS Press, 2014.

[172] John L. Pollock. Defeasible reasoning. *Cognitive Science*, 11(4):481–518, 1987.

[173] Henry Prakken. An abstract framework for argumentation with structured arguments. *Argument & Computation*, 1(2):93–124, 2010.

[174] Henry Prakken. On support relations in abstract argumentation as abstractions of inferential relations. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 735–740. IOS Press, 2014.

[175] Jörg Pührer. Realizability of Three-Valued Semantics for Abstract Dialectical Frameworks. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 3171–3177. AAAI Press, 2015.

[176] Jörg Pührer. ArgueApply: A mobile app for argumentation. In Marcello Balduccini and Tomi Janhunen, editors, *Proceedings of the 14th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2017)*. Springer, 2017. To appear.

[177] Iyad Rahwan and Guillermo R. Simari, editors. *Argumentation in Artificial Intelligence*. Springer, 2009. ISBN 978-0-387-98197-0.

[178] Tjitze Rienstra. *Argumentation in Flux – Modelling Change in the Theory of Argumentation*. PhD thesis, University of Luxembourg, 2014.

[179] Chiaki Sakama. Counterfactual reasoning in argumentation frameworks. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 385–396. IOS Press, 2014.

[180] Ken Satoh. Nonmonotonic reasoning by minimal belief revision. In *Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS 1988)*, pages 455–462, 1988.

[181] Mark Snaith and Chris Reed. Argument revision. *J. Log. Comput.*, 2016. In Press. Available online at http://dx.doi.org/10.1093/logcom/exw028.

[182] Christof Spanring. Perfection in abstract argumentation. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Proceedings of the 6th International Conference on Computational Models of Argument (COMMA 2016)*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 439–446. IOS Press, 2016.

[183] Wolfgang Spohn. Ordinal conditional functions: A dynamics theory of epistemic states. In *Causation in decision, belief change and statistics, vol. 2*, pages 105–134. Kluwer Academic Publishers, 1988.

[184] Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time: Preliminary report. In Alfred V. Aho, Allan Borodin, Robert L. Constable, Robert W. Floyd, Michael A. Harrison, Richard M. Karp, and H. Raymond Strong, editors, *Proceedings of the 5th Annual ACM Symposium on Theory of Computing*, pages 1–9. ACM, 1973.

[185] Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artif. Intell.*, 205:39–70, 2013.

[186] Hannes Strass. Implementing instantiation of knowledge bases in argumentation frameworks. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 475–476. IOS Press, 2014.

[187] Hannes Strass. The relative expressiveness of abstract argumentation and logic programming. In Sven Koenig and Blai Bonet, editors, *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, pages 1625–1631. AAAI Press, 2015.

[188] Hannes Strass. Expressiveness of two-valued semantics for abstract dialectical frameworks. *J. Artif. Intell. Res. (JAIR)*, 54:193–231, 2015.

[189] Hannes Strass. Representational succinctness of abstract dialectical frameworks. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Proceedings of the 3rd International Workshop on Theory and Applications of Formal Argumentation (TAFA 2015)*, 2015.

[190] Hannes Strass and Johannes P. Wallner. Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. *Artif. Intell.*, 226:34–74, 2015.

[191] Matthias Thimm. Strategic argumentation in multi-agent systems. *KI*, 28(3): 159–168, 2014.

[192] Matthias Thimm, Serena Villata, Federico Cerutti, Nir Oren, Hannes Strass, and Mauro Vallati. Summary report of the first international competition on computational models of argumentation. *AI Magazine*, 37(1):102, 2016.

[193] Philippe Torres and Pierre Lopez. Overview and possible extensions of shaving techniques for job-shop problems. In *Proceedings of the 2nd International Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR 2000)*, pages 181–186, 2000.

[194] Hudson Turner. Splitting a default theory. In William J. Clancey and Daniel S. Weld, editors, *Proceedings of the 13th National Conference on Artificial Intelligence and 8th Innovative Applications of Artificial Intelligence Conference (AAAI/IAAI 1996)*, pages 645–651. AAAI Press / The MIT Press, 1996.

[195] Hudson Turner. Strong equivalence made easy: nested expressions and weight constraints. *Theory and Practice of Logic Programming*, 3(4-5):609–622, 2003.

[196] Bart Verheij. Two approaches to dialectical argumentation: admissible sets and argumentation stages. In John-Jules C. Meyer and Linda C. van der Gaag, editors, *Proceedings of the 8th Dutch Conference on Artificial Intelligence (NAIC 1996)*, pages 357–368, 1996.

[197] Klaus W. Wagner. Bounded query classes. *SIAM J. Comput.*, 19(5):833–846, 1990.

[198] Johannes P. Wallner, Andreas Niskanen, and Matti Järvisalo. Complexity results and algorithms for extension enforcement in abstract argumentation. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*, pages 1088–1094. AAAI Press, 2016.

[199] Marianne Winslett. *Updating Logical Databases.* Cambridge University Press, 1990. ISBN 0-521-37371-9.