FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Computational Aspects of Abstract Argumentation

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der technischen Wissenschaften

eingereicht von

Wolfgang Dvořák
Matrikelnummer e0325673

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Privatdoz. Stefan Woltran

Diese Dissertation haben begutachtet:

(Privatdoz. Stefan Woltran) (Prof. Paul E. Dunne)

Wien, 16.02.2012

(Wolfgang Dvořák)

Erklärung zur Verfassung der Arbeit

Wolfgang Dvořák
Hormayrgasse 58/6-7, 1170 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 - einschliešlich 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.

(Ort, Datum) (Unterschrift Verfasser)

Danksagung

Acknowledgements

I am grateful to all the people who supported me and my studies in the last years. As most of them are German speakers I decided to compose the comprehensive acknowledgements in German.

Danksagung

Hier möchte ich mich bei all den Personen bedanken die diese Arbeit erst ermöglicht haben und mein Leben in den letzten Jahren bereichert haben. An erster Stelle bei meiner Freundin Barbara für ihre Unterstützung, Liebe und Nachsicht in den letzten Jahren. Weiters möchte ich den folgenden Personen herzlich danken:

Stefan Woltran für die Betreuung dieser Arbeit, die Ermutigung zum Doktoratsstudium, anregende Diskussionen und die produktive Zusammenarbeit in den letzten Jahren; Reinhard Pichler für seine ansteckende Begeisterung für Komplexitätstheorie, die Betreuung meiner Diplomarbeit und die Ermutigung zu einem Doktoratsstudium; Katrin Seyr und Nysret Musliu dafür das sie mein Interesse an der wissenschaftlichen Arbeit am Institut für Informationssysteme geweckt haben;

meiner Familie für ihre Unterstützung; Meinen Eltern dafür das sie mir ein Studium meiner Wahl ermöglicht haben; Meinen Eltern und Großeltern dafür das sie meine Neugier für Wissenschaft geweckt haben; Meinem Bruder Daniel dafür das er mich als Erster für Mathematik begeistert hat; Meiner Schwester Uli dafür das sie mein Zwilling ist und kontinuierlich versucht mich für Sachen außerhalb der Mathematik & Informatik zu begeistern;

meinen Kolleginnen und Kollegen am Institut für Informationssysteme und unseren Gästen für die produktive Zusammenarbeit in freundschaftlicher Atmosphäre und inspirierende Diskussionen. Unter Ihnen (in zufällgier[1] Reihenfolge): Clemens Nopp, Paul E. Dunne, Sebastian Ordyniak, Christof Spanring, Gerhard Brewka, Reinhard Pichler, Stefan Szeider, Sarah Alice Gaggl, Matti Järvisalo, Stefan Woltran, Johannes Peter Wallner, Günther Charwat, Georg Gottlob, Ringo Baumann und Martin Caminada;

und meinen Studienkollegen, die freundschaftliche Zusammenarbeit war ein Grundstein für meine Freude am Studium und meinen Studienerfolg.

[1] Sortiert nach dem md5 -Hash Wert des Namen so wie er abgedruckt ist.

Abstract

This work is in the context of formal argumentation, a sub-field of Artificial Intelligence. Probably the most popular formalism in argumentation is *abstract argumentation* as introduced by Dung [42]. So called abstract argumentation frameworks abstract from the actual content of arguments and represent them as abstract entities and further abstract from the reasons of conflicts between arguments and represent them as a binary relation. Hence abstract argumentation frameworks can be simply interpreted as directed graphs. On this abstract level one can study the conflicts between arguments and identify coherent sets of arguments. There is a plethora of approaches when a set of arguments should be considered to be coherent, each of these approaches is called a semantics for abstract argumentation.

In every argumentation system, towards conclusions, at some point we have to identify coherent sets of arguments. Hence we identify this as an important computational issue which indeed can be studied on the abstract level. In this work we are doing a computational analysis of evaluating abstract argumentation frameworks with semantics proposed in the literature.

The first part of this work is devoted to a classical *complexity analysis* of the associated reasoning problems, using methods from classical complexity theory. We complement existing results and it turns out that most problems are computationally intractable, i.e. NP-hard and in some cases even harder.

In a second part we explore the range of tractable subclasses. We study *tractable fragments*, i.e. graph classes that allow for an efficient evaluation of the argumentation framework. We extend and complement existing results for acyclic, even-cycle free, symmetric and bipartite Argumentation Frameworks. Moreover we consider the graph parameters tree-width and clique-width to obtain *Fixed-Parameter Tractability* results. We call a problem fixed parameter tractable if it can be solved by an algorithm, with a run-time that may highly increase with the value of the parameter for a concrete instance but is polynomial in the size of the instance.

Finally we consider the intertranslatability of different argumentation semantics. We consider a semantic σ to be translatable to a semantics σ' if there is a (translation) function modifying frameworks such that semantics σ on the original framework is in certain correspondence to σ' on the modified framework.

Zusammenfassung

Diese Arbeit beschäftigt sich mit Formaler Argumentation, einem Teilgebiet der Künstlichen Intelligenz. Einer der erfolgreichsten Formalismen in der Formalen Argumentation sind sogenannte *Abstract Argumentation Frameworks*, die 1995 von Dung eingeführt wurden. Das Konzept der Abstract Argumentation Frameworks abstrahiert von dem konkreten Inhalt der Argumente zu abstrakten Entitäten und einer Konfliktrelation zwischen diesen Entitäten. Diese Frameworks kann man sich also als gerichtete Graphen vorstellen, wobei die Knoten des Graphen die Argumente repräsentieren während gerichtete Kanten Konflikte zwischen Argumenten darstellen. Auf dieser abstrakten Ebene kann man nun die Konflikte zwischen Argumenten studieren und kohärente Mengen von Argumenten identifizieren. Die Literatur kennt eine Vielzahl an unterschiedlichen Kriterien, sogenannte Semantiken, um solche kohärenten Mengen zu definieren.

Eine wichtige Aufgabe in computerunterstützten Argumentations-Systemen ist die Bestimmung von kohärenten Mengen von Argumenten oder allgemeiner die Auswertung des zugehörigen Abstract Argumenation Frameworks mit der entsprechenden Semantik. Der Focus dieser Arbeit liegt in der computationalen Analyse dieser Aufgaben für Abstract Argumenation Frameworks und die unterschiedlichen Semantiken aus der Literatur.

Der erste Teil dieser Arbeit ist eine *Komplexitätsanalyse* dieser Probleme für beliebige Abstract Argumentation Frameworks mit Methoden der klassischen Komplexitätstheorie. Wie sich herausstellt ist ein Großteil der betrachteten Probleme schwierig im Sinne der Komplexitätstheorie, d.h. die Probleme sind NP-schwer und teilweise noch schwerer.

Daher werden in einem zweiten Teil sogennante *tractable fragments* untersucht. Das heißt wir betrachten Abstract Argumentation Frameworks mit einer bestimmte Struktur und untersuchen ob diese mit weniger computionalen Aufwand ausgewertet werden können. Zu diesem Zweck betrachten wir die Graph-Klassen von azyklischen Graphen, Graphen ohne Zyklen gerader Länge, symmetrischen Graphen und bipartiten Graphen. Desweiteren untersuchen wir mehrere Graph-Parameter, welche strukturelle Eigenschaften von Abstract Argumentation Frameworks messen. Mit diesen Parametern wenden wir Methoden der *Parametrsierten Komplexitätstheorie* an, und zeigen, dass Abstract Argumentation Frameworks effizient ausgewertet werden können wenn nur der dazugehörige Graph-Parameter nicht zu groß ist.

Der dritte Teil dieser Arbeit beschäftigt sich mit *Übersetzbarkeit* von verschiedenen Semantiken für Abstract Argumentation Frameworks. Unter einer Übersetzung von einer Semantik A in eine Semantik B versteht man in diesem Zusammenhang eine Funktion die jedem beliebigen Abstract Argumentation Framework F ein Abstract Argumentation Framework G zuordnet sodass die kohärenten Mengen von F bezüglich Semantik A den kohärenten Mengen von G bezüglich Semantik B entsprechen.

Contents

CHAPTER 1

Introduction

1.1 Argumentation in Artificial Intelligence

In recent years, starting with a seminal paper by Dung [42], Argumentation has become one of the major fields in Artificial Intelligence (AI), which is mirrored by the fact that argumentation nowadays appears as keyword at every important AI conference. Moreover, a multitude of articles concerning argumentation in prestigious journals, a two annual dedicated conference on the Computational Models of Argument, a workshop at Theory and Applications of Formal Argumentation, and the recently founded journal Argument & Computation underlines the significance of this research direction.

Informally argumentation concerns building arguments, identifying conflicts between arguments and then selecting coherent set of arguments to finally obtain a decision or conclusion. This allows for a kind of defeasible reasoning and is indeed non-monotonic. The origin of an argument setting can be of very different nature. For instance it may be by a single agent evaluating her knowledge to make a decision or by a multi-agent scenario where several agents negotiate. Argumentation was connected to existing work on non-monotonic reasoning by the observation that many non-monotonic formalisms can realised as a kind of argumentation. Constructing arguments by as defeasible proofs in the formalism, identifying conflict between these defeasible proofs and finally resolving conflicts. Such a correspondence was given for instance for default logic [42], defeasible logic [75] and answer set programming [42]. For a survey on argumentation in artificial intelligence the interested reader is referred to [14].

1.1.1 Abstract Argumentation

The core of resolving argumentation can be formalised by abstract argumentation frameworks introduced by Dung [42]. Such frameworks consists of abstract entities representing arguments and edges, so called attacks, representing conflicts between different arguments. However abstract argumentation frameworks do neither take care of the arguments' statements nor where conflicts between arguments origin from. As these attacks have a direction one can interpret this frameworks as directed graphs. In abstract argumentation one works with this graph structure and tries to identify sets of coherent arguments. Different definitions for a set of arguments being coherent sets, so called semantics, have being proposed in the literature, each of them having slightly different intuitions and properties [12, 22, 23, 42, 103] (see [7, 10] for an overview). Abstract argumentation is nowadays one of the central formalisms in argumentation research. An important part of the research concerning semantics for abstract argumentation frameworks is dedicated to properties that semantics should satisfy and relations between semantics (see e.g. [6, 8, 10, 72, 94]).

As indicated before abstract argumentation does not stand alone but typically appears as one step in an entire argumentation process, which we will discuss next.

1.1.2 The Argumentation Process

Already in Dung's seminal paper the concept of abstract argumentation frameworks was stated together with several ways to instantiate such frameworks, i.e. ways to generate arguments and identifying conflicts. In the following we illustrate our abstract view on the argumentation process which is inspired by the work of Caminada and Amgoud [26].

We consider the overall argumentation process to consist of the following six steps:

1. Start with or build a knowledge base (KB)
2. Build arguments out of the KB
3. Identify conflicts between arguments
4. Abstract from the internal structure of the arguments
5. Resolve conflicts between arguments and select acceptable subsets of arguments
6. Draw conclusions (aggregate from the extensions)

For an illustration of these steps see Figure 1.1, where we sketch an instantiation of defeasible logic in the spirit of the ASPIC [26, 99] system.

In general, to have an argument we have first to collect knowledge, where typically some parts are defeasible. This can be either a set of logical rules / formulae or even the knowledge of an expert. In a second step we need rules how to build arguments based out of the knowledge base. Next we have to identify which arguments are in conflict with each other and whether this conflict has a direction or goes in both directions. After that we can abstract from the concrete contents of the arguments and just consider a framework of abstract arguments and conflicts between them. On this abstract layer we can resolve conflicts and identify coherent sets of arguments. Having these coherent sets of arguments at hand we can decide which arguments to accept and finally we can draw conclusions by again considering the contents of the accepted arguments.

However, in many instantiations of abstract argumentation frameworks these steps are not clearly separated and may interact with each other. For example when not instantiating from a logic knowledge base the construction of the argumentation framework might be by a "knowledge engineer" who forms arguments and identifies conflicts from the accessible knowledge.

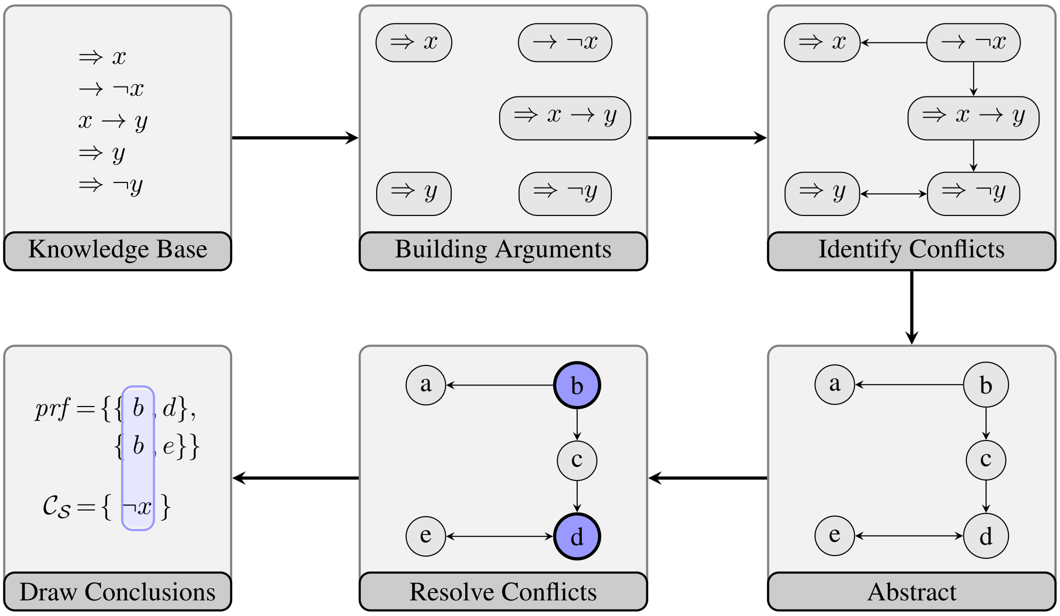


Figure 1.1: An illustration of the Argumentation Process

1.2 Computational Issues in Argumentation

As mentioned, evaluating arguments with respect to a semantics defining coherent sets of arguments is one of the core issues of each argumentation system. As these semantics operate on the abstract level, abstract argumentation frameworks are the right layer to study computational problems.

Typical tasks for such an argumentation engine are: computing one (or all) coherent set of arguments, deciding whether a specific argument is contained in one (or all) coherent set of arguments, and deciding whether a given set of arguments is coherent. The general complexity of abstract argumentation w.r.t. different semantics gained some interest in the literature [12, 31, 46, 48, 50]. However, there are still several open gaps which we attempt to close in this work.

As most of these problems turn out to be computationally intractable, that is NP-hard and even harder, one is also interested in identifying tractable cases. One approach is to only consider argumentation frameworks with a special kind of graph structure. Such investigations have been undertaken in the work of Coste-Marquis et al. [31] on symmetric frameworks, the work by Dunne and Bench-Capon [47] on even-cycle free frameworks, and an extensive study by Dunne [44]. However, all these studies are limited to certain semantics.

Another promising approach for tackling in general intractable problems comes from parameterized complexity theory (see e.g. [93]). The idea behind parameterized complexity is that the complexity of a problem typically does not merely depend on the entire size but on some specific properties of the instance which can be measured by a parameter. For instance, in graph theory there are several parameters measuring the structure of a graph, e.g. the parameter tree-width which intuitively measures how tree-like a graph is. Having such a parameter at hand one can do a more fine-grained complexity-analysis and in the good case there are algorithms with a runtime that might be exponential in the parameter but is only polynomial in the size of the instance. If then one only considers instances where this parameter is bounded by a fixed constant one has a polynomial time algorithm, we refer to such algorithms as fixed-parameter tractable. First investigations for fixed-parameter tractability in abstract argumentation have been undertaken in [44], where the graph parameter tree-width was considered. More recently Ordyniak and Szeider [95] also considered another parameterisation for abstract argumentation, namely the distance to known tractable graph classes.

A successful approach towards implementation of abstract argumentation systems is what we call the *reduction approach*. That is, instead of designing and implementing new algorithms for argumentation, one encodes argumentation problems in well studied formalisms with sophisticated solvers. There are several formalisms to which (abstract) argumentation reasoning has been reduced. Most prominently there are the following approaches to reduce the computation of extensions or specific argumentation reasoning tasks to: SAT-solving [17] and resp. to Quantified Boolean Formulae (QBF) [64]; Answerset-Programming (ASP) [65]; or a Constraint Satisfaction Problem (CSP) [1]. In this work we study the possibilities of using the reduction approach within abstract argumentation. That is, we study if and how reasoning with one argumentation semantics can be reduced, by modifying the argumentation framework, to reasoning with another semantics, for which we might have sophisticated tools.

1.3 Structure of the Thesis

This thesis is structured in six chapters, the first being this introduction. The remaining of this work is structured as follows:

- In Chapter 2 we discuss the necessary *background* for our further investigations. First, we introduce the concept of abstract argumentation frameworks and the different semantics together with known properties of them. As well we provide some novel results concerning semantics which we will make use of later on. Second, we briefly review the necessary background from classical computational complexity theory as well as from parameterized complexity theory. Finally, we discuss the background on direct graphs and graph-parameters together with Meta-theorems for parameterized complexity results.

- The *complexity analysis* in Chapter 3 starts with an overview of existing work and then contributes in the following directions:

  - In Section 3.2 we study tractable reasoning problems and classify them w.r.t. completeness for the class of polynomial time decidable problems. Firstly, we show several problems for the so called grounded semantics to be complete for polynomial time. Secondly, we extend these results to another semantics, the resolution-based grounded semantics. Finally, we show that several problems which are known to be solvable in polynomial time, can be actually solved in logarithmic space and are thus (under typical complexity-theoretic) assumptions not complete for the class of polynomial time.

  - In Section 3.3 we study the *complexity of semi-stable and stage semantics*. Starting from the complexity analysis for semi-stable semantics provided by Dunne and Caminada [50], we first complement their results by presenting matching lower bounds for credulous and skeptical reasoning. Secondly, we provide a complete complexity analysis for the related stage semantics.
  - In Section 3.4 we consider the *complexity of ideal reasoning*. That is we consider parameterized ideal semantics and give generic complexity results, i.e. upper and lower complexity bounds for several reasoning tasks, using the complexity classification of the base-semantics as parameter. Moreover we study all the semantics under our considerations as base-semantics and give exact complexity characterisations for the corresponding ideal semantics.

- Chapter 4 – *Towards Tractability* – addresses the issue of identifying tractable instances of argumentation problems. The contributions are organised in three subsections and are then finally summarised and discussed together with related work.
  - Section 4.1 studies *Tractable Fragments*, i.e. graph classes on which argumentation problems are tractable which are intractable in the general case. Four classes of argumentation frameworks, are considered, namely acyclic argumentation frameworks, argumentation frameworks which are free of even-length cycles, bipartite argumentation frameworks and symmetric argumentation frameworks. We review and complement existing results from the literature and extend them to all of our semantics. Finally we consider fragments that do not yield tractability but allow to solve argumentation problems, which are in general hard for the second level of the polynomial hierarchy, within the easier complexity classes NP or coNP.
  - Section 4.2 provides *Fixed-Parameter Tractability* results concerning the graph parameters tree-width and clique-width. That is we show that the reasoning problems under our considerations can be solved by algorithms whose worst-case runtime highly increases with the parameter but only linear with the size of the instances. To this end we first provide monadic second order logic encodings of the argumentation semantics and then apply the meta-theorems.
  - In Section 4.3 we prove *Fixed-Parameter Intractability* results for several parameters generalising the parameter tree-width for directed graphs. That is we prove that argumentation problems remains hard on argumentation frameworks where the parameter cycle-rank is bounded and then use a meta-theorem to extend these results to the parameters directed path-width, Kelly-width, DAG-width, and directed tree-width.

- In Chapter 5 we study the *intertranslatability of argumentation semantics*. By a translation from a semantics σ to a semantics σ' we understand a function modifying argumentation frameworks such that semantics σ on the original argumentation frameworks is in correspondence with semantics σ' on the modified argumentation frameworks. The contributions in this chapter are as follows.

  - In Section 5.1 we define properties for translations basically along the lines of Janhunen [79]. In particular, we consider here as desired properties efficiency (the translation can be computed in logarithmic space w.r.t. the given argumentation frameworks), modularity (the translation can be done independently for certain parts of the framework) and faithfulness (there should be a clear correspondence between the extensions of the translated argumentation frameworks and the original argumentation frameworks). However, we also consider some additional features which are needed to deal with some of the argumentation semantics (for instance, the admissible semantics always yields the empty set as one solution; thus filtering such an entire solution is necessary).
  - Section 5.2 contains our main results in this chapter, in particular we provide translations between grounded, stable, admissible, complete, preferred, semi-stable and stage stage when possible. We analyse these translations w.r.t. the introduced properties using as minimal desiderata efficiency and (a particular form of) faithfulness.
  - Section 5.3 then provides negative results, i.e. we show that certain translations between semantics are not possible. Some of these impossibility results make use of typical complexity-theoretic assumptions together with results from Chapter 3; others are genuine due to the different properties of the compared semantics.

- Finally, in Chapter 6 we *conclude* this work. We summarise and discus the results achieved and give an outlook to possible future research directions.

1.4 Publications

Parts of this thesis where previously published in scientific articles.

The complexity analysis of semi-stable and stage semantics presented in Section 3.3 was published in *Information Processing Letters* [54], and the complexity analysis of ideal reasoning has been partly presented at the prestigious *IJCAI'11* Conference [59]. Moreover, Propositions 6 & 8 in Chapter 3 are based on results published in [56]. However, we also complement the already published complexity analysis by novel results, e.g. by Proposition 7 giving a P-hardness result for resolution based grounded semantics and Theorem 26 studying the complexity of ideal reasoning with stable semantics. In Chapter 4, Section 4.1 presents novel results for tractable fragments, and Section 4.2 builds on observations presented at *KR'10* [57] and *COMMA'10* [61], and on the technical side on MSO encodings of stage and semi-stable semantics published in [54]. However, the given MSO_1 encoding of resolution-based grounded semantics (cf. Section 4.2.1) as well as the concrete encodings of the reasoning problems in Section 4.2.2 are novel. In Section 4.3 we present intractability results, which where (for the case of preferred semantics) previously presented at *KR'10* [57]. Finally the results presented in Chapter 5 were published in the *Journal of Artificial Intelligence Research* [56] (a short version was presented at *NONMON@30* [55]).

Let us briefly mention work of the author, that is closely related but not included in this thesis. Firstly, there is a paper presented at the *TAFA'11* workshop, which addresses computational properties of determining the justification status of an argument in a labeling-bases setting [53]. This paper strongly builds on complexity results for extension-based semantics, some of them presented in Chapter 3 of this thesis. Secondly, there is the work on dynamic programming algorithms for abstract argumentation that build on certain graph-parameters. These algorithms are based on fixed parameter tractability results presented in Section 4.2 of this thesis and where presented at *KR'10* [57] (for tree-width) and *COMMA'10* [58] (for clique-width). The first one also resulted in the system dynPARTIX[1] first presented at *INAP'11* [60]. Most recently there is the work on complexity-sensitive decision procedures for Abstract Argumentation which builds on complexity results of this thesis and will be presented at *KR'12* [62].

[1] http://www.dbai.tuwien.ac.at/research/project/argumentation/dynpartix/

CHAPTER 2

Background

In this chapter we briefly present the necessary background for the work presented in this thesis and give purposive links to related literature for deeper insights.

In Section 2.1 we consider the field of abstract argumentation: the concept of abstract argumentation frameworks as introduced by Dung [42]; the plethora of argumentation semantics, relevant properties of the semantics and relations between different semantics; and computational problems of interest in abstract argumentation.

In Section 2.2 we give an overview of complexity theoretic concepts we use in this work. In particular we introduce the complexity classes we will need later on and their relations. Moreover towards the hardness proofs in Chapter 3 and Chapter 4, we present complete problems for the relevant complexity classes. Finally we introduce the concept of fixed-parameter tractability and related notions.

In Section 2.3 we discuss different kind of graph parameters. To this end we first recall the necessary concepts from graph theory. Then we present the graph parameters tree-width, clique-width and cycle-rank and discuss their properties and relations. Moreover we introduce monadic second order logic and present useful meta-theorems for obtaining fixed-parameter tractability results w.r.t. tree-width and clique-width, which are based on monadic second order logic.

2.1 Abstract Argumentation

In this section we first introduce (abstract) argumentation frameworks as defined by Dung [42], give an overview of the most important semantics for such frameworks, discuss properties of and relations between these semantics and finally present the reasoning problems of interest.

We start with the formal definition of Dung's argumentation frameworks.

Definition 1. *An* argumentation framework (AF for short) *is a pair $F = (A, R)$ where A is a finite*[1] *non-empty*[2] *set of arguments and $R \subseteq A \times A$ is the attack relation. For a given AF*

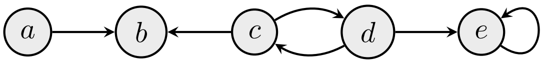
[1] In principle AFs can be infinite but in this work we restrict ourselfs to finite AFs.

[2] For technical reasons we only consider AFs with $A \neq \emptyset$.

$F = (A, R)$ we use A_F to denote the set A of its arguments and R_F to denote its attack relation R. If $(a, b) \in R$ we say that a attacks b.

As one can see AFs can naturally be represented as a directed graphs, interpreting the arguments as vertices and the attacks as edges (we will come back to this in Section 2.3). The following example illustrates an AF by using a standard graphical representation of directed graphs.

Example 1. *Consider the AF $F = (A, R)$, with $A = \{a, b, c, d, e\}$ and $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$. The graph representation of F is given as follows.*



For convenience we introduce some shorthands for denoting conflicts between arguments and sets of arguments:

Definition 2. *Given an AF $F = (A, R)$. For arguments $a, b \in A$ we may use the notation $a \rightarrowtail^R b$ instead of $(a, b) \in R$. For a set of arguments $S \subseteq A$ and an argument $a \in A$, we also write $S \rightarrowtail^R a$ (resp. $a \rightarrowtail^R S$) in case there exists an argument $b \in S$, such that $b \rightarrowtail^R a$ (resp. $a \rightarrowtail^R b$). In case no ambiguity arises, we may use $\rightarrowtail$ instead of $\rightarrowtail^R$.*

The main issue in argumentation scenarios is to identify in (some sense) coherent sets of arguments. Thus one has to fix a notion of coherence for argument sets, the so-called semantics of abstract argumentation.

2.1.1 Semantics

In the following we recapitulate the most popular semantics for abstract argumentation (see also [7, 10] for an overview), but first we formalize what we consider to be an (extension-based) semantics for abstract argumentation.

Definition 3. *An* extension-based semantics[3] *for abstract argumentation frameworks is a function σ mapping each AF F to a set of extensions $\sigma(F) \subseteq 2^{A_F}$. If for each F, $|\sigma(F)| = 1$ then we call σ a* unique status semantics, *otherwise* multiple status semantics. *For a unique status semantics we denote the unique extension of F as $E_\sigma(F)$.*

In the following we recall the most important semantics for abstract argumentation, starting with the semantics introduced by Dung [42] and the related concepts. A very basic notion for argumentation is *conflict-freeness* which underlies all of the semantics considered in this work. It mirrors the observation that if one argument attacks another then these arguments should not be accepted simultaneously.

[3] In the following we omit “extension-based“ and just talk about semantics, as we only deal with extension-based semantics in this work. Some work has be done on defining semantics via 3-valued labelings [27], but as these labelings can be computed easily from the corresponding extensions, labeling-based semantics are not of additional interest from the computational point of view. Nevertheless such argument labelings can be useful for an algorithmic purposes, see e.g. [57, 92, 104].

Definition 4. *Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is* conflict-free *in F, if there is no argument $a \in S$, such that $S \rightarrowtail a$. We denote the collection of sets which are conflict-free (in F) by $cf(F)$.*

The conflict-free sets of the AF in Example 1 are $\{\}$, $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{a, c\}$, $\{a, d\}$ and $\{b, d\}$. Next we consider maximal conflict-free sets, which are still not interesting as a semantics themselves, but can be used as basis for defining semantics and are thus often useful on a technical level.

Definition 5. *Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is a* naive set *in F, if S is a $\subseteq$-maximal conflict-free set for F, i.e. $S \in cf(F)$ and for each $T \in cf(F)$, $S \not\subset T$. We denote the collection of all naive sets of F by $naive(F)$.*

The naive sets in Example 1 are $\{a, c\}$, $\{a, d\}$ and $\{b, d\}$. One can see that every argument which is not in conflict with itself is in at least one naive extensions, which in most cases might be unwanted.

Another important concept for Dung's semantics is the notion of defense. The intuition of this concept is that one would accept an argument a that is attacked by an argument b only if there is an argument c that refutes b. Such an argument c disables b and thus defends a against the attack from b.

Definition 6. *Let $F = (A, R)$ be an AF. An argument $a \in A$ is* defended *by a set S if for each argument $b \in A$ with $(b, a) \in R$ we have that $S \rightarrowtail b$. Sometimes we may also say that the argument a is* acceptable *w.r.t. S.*

Based on the notion of conflict-freeness and defense, Dung defined admissible sets, i.e. conflict-free sets that defend all of their arguments.

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

For the AF in Example 1 we have the admissible sets $\{\}$, $\{a\}$, $\{c\}$, $\{d\}$, $\{a, c\}$ and $\{a, d\}$. Note that $\{b\}$ and $\{b, d\}$ are not admissible as b is not defended against the attack $(a, b) \in R$.

We observe that the empty set is always admissible, which makes admissible semantics inapplicable for cautious reasoning modes. A problem with admissible sets is that we might reject arguments without a reason, in particular we may reject arguments which are not attacked or even not involved in a conflict at all.

To overcome this one can consider different kind of maximal admissible sets which leads to complete, preferred and stable semantics. First let us consider a maximality where each argument defended by an extension is already contained in the extension (we will introduce this formally, later on, as reinstatement property).

Definition 8. *Let $F = (A, R)$ be an AF. A set S is a* complete extension *of F, if $S \in adm(F)$ and, for each $a \in A$ defended by S (in F), $a \in S$ holds. We denote the collection of all complete extensions of F by $com(F)$.*

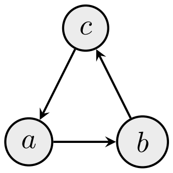


Figure 2.1: An AF without stable extension

The complete extensions for the AF in Example 1 are $\{a\}$, $\{a, c\}$ and $\{a, d\}$. Note that as a is not attacked by any other argument it is contained in each complete extension,

Another approach for admissibility based semantics is to consider $\subseteq$-maximal admissible sets.

Definition 9. *Let $F = (A, R)$ be an AF. A set S is a* preferred extension *of F, iff S is a $\subseteq$-maximal admissible set for F, i.e. $S \in adm(F)$ and for each $T \in adm(F)$, $S \not\subset T$. We denote the collection of all preferred extensions of F by $prf(F)$.*

For the AF in Example 1 we have that $\{a, c\}$ and $\{a, d\}$ are the only preferred extensions.

An even stronger maximality condition underlies stable semantics. Here we consider only admissible sets that attack all arguments not contained in the set.

Definition 10. *Let $F = (A, R)$ be an AF. A set S is a* stable extension *of F, iff S is a conflict-free set for F and for each $b \in A$ we have that $S \rightarrowtail b$. We denote the collection of all stable extensions of F by $stb(F)$.*

For the AF in Example 1 we have that $\{a, d\}$ is the only stable extensions. We mention that a conflict-free set that attacks all arguments that are not in the set is clearly an admissible set, as it attacks all possible attackers. The pitfall with stable semantics is that there exists AFs without an stable extension, e.g. the AF $(\{a, b, c\}, \{(a, b), (b, c), (c, a)\})$ illustrated in Figure 2.1.

Next we introduce an admissibility-based semantics which underlies a high level of skepticism.

Definition 11. *Let $F = (A, R)$ be an AF. A set S is the* grounded extension *of F, if S is the $\subseteq$-minimal complete extension. By a small abuse of notation we use $grd(F)$ to denote both, the set containing the grounded extension as single element and the grounded extension itself.*

For the AF F in Example 1 we have that $grd(F) = \{a\}$.

So far, we have introduced semantics along the lines of Dung's seminal paper [42]. We proceed with two semantics based on a different kind of maximality, i.e. the maximality of the so called range [103] of an extension.

Definition 12. *Given an AF $F = (A, R)$, for a set $S \subseteq A$, we define the* range of S *in F, denoted as S^+_R, as the set $S \cup \{b \mid S \rightarrowtail b\}$. We write S^+ instead of S^+_R if R is clear from the context and no ambiguity arises.*

Again consider our example AF and the sets $S = \{a, c\}$, $T = \{a, d\}$. The range S^+ of S is the set $\{a, b, c, d\}$ and the range of T is the set of all argument, i.e. $T^+ = \{a, b, c, d, e\}$, which mirrors the fact that T is a stable extension. Next we introduce some additional notation concerning the range of extensions and related concepts, which will be useful later on.

Definition 13. *Given an AF $F = (A, R)$, we will use the following notation:*

- *For a set $S \subseteq A$, $S_R^\oplus$ to denote $\{b \mid S \rightarrowtail b\}$* [4]
- *For a set $S \subseteq A$, $S_R^\ominus$ to denote $\{b \mid b \rightarrowtail S\}$*
- *For sets $S, T \subseteq A$ we write $S \leq_R^+ S'$ iff $S_R^+ \subseteq T_R^+$ and $S <_R^+ S'$ iff $S_R^+ \subset T_R^+$*

In case no ambiguity arises we will omit the subscript R in the above notations.

We are now ready to introduce semi-stable semantics, which where first introduced by Verheij [103] and later popularized by Caminada [22, 28].

Definition 14. *Let $F = (A, R)$ be an AF, and for a set $S \subseteq A$. A set S is a* semi-stable extension *of F, if S is a $\leq_R^+$-maximal admissible set for F, i.e. $S \in adm(F)$ and for each $T \in adm(F)$, $S \not\leq_R^+ T$. We denote the collection of all semi-stable extensions of F by $sem(F)$.*

If we consider conflict-free sets instead of admissible and apply $\leq_R^+$-maximality we get stage semantics [103].

Definition 15. *Let $F = (A, R)$ be an AF. A set S is a* stage extension *of F, if S is a $\leq_R^+$-maximal conflict-free set for F, i.e. $S \in cf(F)$ and for each $T \in cf(F)$, $S \not\leq_R^+ T$. We denote the collection of all stage extensions of F by $stg(F)$.*

Both semantics semi-stable and stage are motivated by the fact that there are AFs without an stable extension. It is easy to see that if an AF has an stable extension then stable, semi-stable and stage semantics coincide, e.g. for the AF F in Example 1 we have that $stb(F) = sem(F) = stg(F) = \{\{a, d\}\}$. Further, for each AF, both semi-stable and stage semantics propose at least one extension and thus somehow extend stable semantics to AFs where no stable extension exists, but in a different ways. Consider again the AF $F = (\{a, b, c\}, \{(a, b), (b, c), (c, a)\})$ (see Figure 2.1). We have that $sem(F) = \{\emptyset\}$ while $stg(F) = \{\{a\}, \{b\}, \{c\}\}$.

We proceed with two parametric approaches for defining semantics, that is using a arbitrary semantics and build a new semantics in a certain way. A popular parametric approach is the family of resolution-based semantics [12], with the resolution-based grounded semantics being its most popular instance. Towards a definition of resolution-based semantics we first need the concept of a resolution of an AF.

Definition 16. *Given AF $F = (A, R)$. A (full) resolution $\beta \subset R$ of F is a $\subset$-minimal set of attacks such that for each pair $\{(a, b), (b, a)\} \subseteq R$ with $a \neq b$ either $(a, b) \in \beta$ or $(b, a) \in \beta$. We denote the set of all resolutions of an AF F by $\gamma(F)$.*

[4] By definition we have that $S^+ = S \cup S^\oplus$.

Intuitively a resolution eliminates all symmetric conflicts in an AF, by deleting one of the attacks. On the basis of such resolved frameworks one can define resolution-based semantics.

Definition 17. *Given an AF $F = (A, R)$ and a semantics σ. We define the corresponding resolution based semantics σ^* as follows:*

$$\sigma^*(F) = \min_{\subseteq} \bigcup_{\beta \in \gamma(F)} \{\sigma((A, R \setminus \beta))\}$$

We note that the resolution-based grounded semantics grd^* satisfies many desirable properties [12] thus we will focus on this instantiation here. The following definition gives a simplified characterization for resolution-based grounded semantics.

Definition 18. *Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is a* resolution-based grounded extension *of F if*

1. *there exists a resolution β such that $grd((A, R \setminus \beta)) = S$ and*
2. *there is no resolution β' such that $grd((A, R \setminus \beta')) \subset S$.*

We denote the collection of all resolution-based grounded extensions of F by $resGr(F)$.

Next we consider parametrised ideal semantics. The concept of ideal semantics was original stated as an alternative for skeptical reasoning w.r.t. preferred semantics [43] and later also applied to semi-stable semantics [24]. Here we give a general definition of ideal semantics, abstracting from the approaches in [24, 43], which is applicable to arbitrary semantics.

Definition 19. *Let $F = (A, R)$ be an AF and σ a semantics. The* ideal sets *w.r.t.* base semantics *σ of F are those that satisfy the following constraints.*

I1. $S \in adm(F)$

I2. $S \subseteq \bigcap_{E \in \sigma(F)} E$ *if* $\sigma(F) \neq \emptyset$ *and* $S = \emptyset$ *otherwise.*

We say that S is an ideal extension *of F w.r.t. σ, if S is a $\subseteq$-maximal ideal set (of F) w.r.t. σ. We use σ^{idl} to denote the collection of ideal* sets *w.r.t. σ and σ^{ie} to denote the set of ideal extensions w.r.t. σ.*

For historical reasons we refer to prf^{idl}, prf^{ie} as standard-ideal semantics [43].

2.1.2 Properties

Here we consider basic properties of semantics as well as the relations between them. We start with Dung's famous fundamental lemma [42], which concerns the compatibility of admissible sets and acceptable arguments.

Lemma 1 (Fundamental Lemma). *Given an AF $F = (A, R)$, a set $S \in adm(F)$ and two arguments $a, b \in A$. If a, b are defended by S then*

$$
\begin{aligned}
cf(F) &= \{S \subseteq A \mid \forall x, y \in S, (x,y) \notin R\}\\
naive(F) &= \{S \in cf(F) \mid S \subset T \Rightarrow T \notin cf(F)\}\\
grd(F) &= \mathcal{F}^k(\emptyset),\ \text{for } k \text{ such that } \mathcal{F}^k(\emptyset) = \mathcal{F}^{k+1}(\emptyset)\\
adm(F) &= \{S \in cf(F) \mid S \subseteq \mathcal{F}_F(S)\}\\
com(F) &= \{S \in adm(F) \mid \mathcal{F}(S) \subseteq S\}\\
stbF &= \{S \in adm(F) \mid S^+ = A\}\\
prf(F) &= \{S \in adm(F) \mid S \subset T \Rightarrow T \notin adm(F)\}\\
sem(F) &= \{S \in adm(F) \mid S^+ \subset T^+ \Rightarrow T \notin adm(F)\}\\
stg(F) &= \{S \in cf(F) \mid S^+ \subset T^+ \Rightarrow T \notin cf(F)\}\\
resGr(F) &= \min_{\subseteq} \bigcup_{\beta \in \gamma(F)} \{grd((A, R \setminus \beta))\}
\end{aligned}
$$

Figure 2.2: Cheat-Sheet: Semantics for abstract argumentation (given AF $F = (A, R)$).

1. $S' = S \cup \{a\} \in adm(F)$ *and*
2. S' *defends* b.

One direct consequence of the fundamental lemma is that each admissible set is contained in a complete extension. Next we introduce a useful concept when dealing with admissibility-based argumentation semantics, the characteristic function.

Definition 20. *The* characteristic function $\mathcal{F}_F : 2^A \to 2^A$*, of an AF* $F = (A, R)$*, is defined as* $\mathcal{F}_F(S) = \{x \in A_F \mid x$ *is defended by* $S\}$.

Now we can give an alternative characterisation for the grounded extension, which allows for an efficient computation. We have that the grounded extension of an AF F is the least fixed-point of $\mathcal{F}_F$ [42] and further that the complete extensions are the conflict-free fixed-points of $\mathcal{F}_F$. The characteristic functions also allows for compacter characterisations of our semantics (see Figure 2.2).

Now let us consider basic properties of semantics. First, immediate by the definitions, we have that all semantics except stable semantics always propose at least one extension, although it might be just the emptyset. Further we have that some semantics are refinements of others in the sense that each extension w.r.t. one semantics is also an extension w.r.t. another semantics. We summarise these relations in the following proposition.

Proposition 1. *Given AF* $F = (A, R)$ *then the following* $\subseteq$*-relations hold:*

1. $stb(F) \subseteq sem(F), stb(F) \subseteq stg(F)$
2. $sem(F) \subseteq prf(F)$

3. $\mathit{prf}(F) \subseteq \mathit{com}(F)$

4. $\mathit{com}(F) \subseteq \mathit{adm}(F)$

5. $\mathit{adm}(F) \subseteq \mathit{cf}(F)$

6. $\mathit{stg}(F) \subseteq \mathit{naive}(F)$

7. $\mathit{naive}(F) \subseteq \mathit{cf}(F)$

8. $\mathit{resGr}(F) \subseteq \mathit{com}(F)$

9. $\mathit{grd}(F) \subseteq \mathit{com}(F)$

Proof. We prove each point separately:
(1) Consider $E \in \mathit{stb}(F)$. By definition E is conflict-free and as E attacks all arguments in $A \setminus E$ it is also admissible. Now we have that $E^+ = A$ and thus E is clearly $\leq^+$-maximal and thus $E \in \mathit{stg}(F)$ and $E \in \mathit{sem}(F)$.
(2) Towards a contradiction assume $E \in \mathit{sem}(F)$ and $E \notin \mathit{prf}(F)$. Then there exists $S \in \mathit{adm}(F)$ with $E \subset S$. As $E \subset S$ clearly also $E^+_R \subseteq S^+_R$ and there exists an argument $x \in S \setminus E$. We have that $E \cup \{x\} \subseteq S$ it is conflict-free and thus $x \notin E^+_R$. As clearly $x \in S^+_R$ we obtain $E <^+ S$, a contradiction.
(3) Assume $E \in \mathit{prf}(F)$ and $E \notin \mathit{com}(F)$ then there exists an argument $x \in A \setminus E$ and E defends x. But then by the fundamental lemma $E \cup \{x\} \in \mathit{adm}(F)$, a contradiction to the $\subseteq$-maximality of E.
(4) & (5) Immediate by the definition.
(6) Towards a contradiction assume $E \in \mathit{stg}(F)$ and $E \notin \mathit{naive}(F)$. Then there exists $S \in \mathit{cf}(F)$ with $E \subset S$. As $E \subset S$ clearly also $E^+_R \subseteq S^+_R$ and there exists an argument $x \in S \setminus E$. We have that $E \cup \{x\} \subseteq S$ it is conflict-free and thus $x \notin E^+_R$. As clearly $x \in S^+_R$ we obtain $E <^+ S$, a contradiction.

(7) Immediate by the definition.
(8) see [12].
(9) Immediate by the definition. □

Notice that the $\subseteq$-relations above are all that holds in general. For the remaining cases one can easily constructs AFs where the $\subseteq$-relation does not hold. We illustrate these $\subseteq$-relations in Figure 2.3. If two semantics are in $\subseteq$-relation we may also say that one semantics preserves the other one.

Definition 21. *Let σ and θ be semantics. If for all AFs $F = (A, R)$, $\sigma(F) \subseteq \theta(F)$, we call σ a θ-*preserving *semantics.*

We mention that that the preserving property is transitive, i.e. if σ is θ-preserving and θ is τ-preserving then also σ is τ-preserving. For instance, all semantics under our considerations are cf-preserving, but the cf, naive and the stg semantics are not adm-preserving. Moreover we have that sem is prf-preserving, and stg is naive-preserving, which will be important later

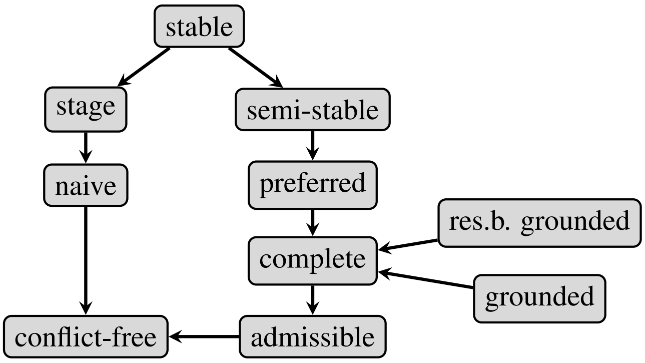


Figure 2.3: Relations between argumentation semantics: An arrow from a semantics σ to another semantics τ encodes that each σ-extension is also a τ-extension.

on when we consider the complexity of parametric ideal semantics. An overview of which semantics preserves which semantics is given in Figure 2.3.

Next we formalise the property separating complete extensions and admissible sets.

Definition 22. *A semantics σ satisfies the* reinstatement *property iff for every AF $F = (A, R)$ and $E \in \sigma(F)$ it holds that $\mathcal{F}_F(E) \subseteq E$.*

In other words for a semantics that satisfies reinstatement we have that all arguments which are acceptable w.r.t. an extension are already contained in the extension. Clearly complete semantics satisfies the reinstatement-property and thus also all com-preserving semantics satisfy the reinstatement-property (compare Figure 2.3). Furthermore, one can easily construct examples where the not com-preserving semantics under our considerations, i.e. cf, $naive$, stg, adm, do not satisfy the reinstatement-property.

Properties of Parametric Ideal Semantics

First of all, we show that for all reasonable base semantics σ, and indeed for all semantics under our considerations there is a unique ideal extension w.r.t. σ. To this end we need to following lemma showing that each conflict-free set has a unique $\subseteq$-maximal admissible subset.

Lemma 2. *Given an AF $F = (A, R)$ and a set $S \in cf(F)$ then there is a unique set $E \subseteq S$ such that $E \in adm(F)$ and for each $E' \subseteq S$ with $E \subset E'$ it holds that $E' \notin adm(F)$.*

Proof. It suffices to show that if $D \subseteq S, D \in adm(F)$ and $E \subseteq S, \in adm(F)$ then also $D \cup E \subseteq S, D \cup E \in adm(F)$, i.e. the admissible subsets of S are closed under set union. First note that $D \cup E \in cf(F)$ since $D \cup E \subseteq S$ and $S \in cf(F)$. It further holds that $D \cup E \in adm(F)$. We have that both D and E are admissible and thus any $a \in (D \cup E)^{\ominus}$ either belongs to $D^{\ominus}$ (and so is counterattacked by some $b \in D$) or is in $E^{\ominus}$ (and, in the same way, counterattacked by some $b \in E$). □

Having Lemma 2 at hand we can give our result.

Proposition 2. *If a semantics σ is cf-preserving, then each AF F has a unique ideal extension w.r.t. base semantics σ (σ^{ie} is a unique status semantics).*

Proof. First if the semantics does not propose an extension, by definition the emptyset is the unique ideal extension. Now let us assume there is at least extension E'. We have that $\bigcap_{E \in \sigma(F)} E \subseteq E'$ and as $E' \in cf(F)$ also $\bigcap_{E \in \sigma(F)} E$ is conflict-free. Finally by Lemma 2 we know that $\bigcap_{E \in \sigma(F)} E \subseteq E'$ has a unique $\subseteq$-maximal admissible subset. □

In following we use $E^{ie}_{\sigma}(F)$ to denote the unique ideal extension w.r.t. base-semantics σ.

Proposition 3. *If σ satisfies the reinstatement property and σ guarantees at least one extension, then σ^{ie} is comp-preserving.*

Proof. Let $F = (A, R)$ be an AF and $E \in \sigma^{ie}(F)$. By definition E is admissible and it remains to show that σ^{ie} satisfies the reinstatement property i.e. that every argument defended by E belongs to E,. Thus let $a \in A$ be an arbitrary argument which is defended by E. As E is part of every σ-extension we have that every σ-extension defends a and as σ satisfies the reinstatement property a is contained in every σ-extension. By the fact that a is skeptically accepted and defended by E we have that $S \cup \{a\}$ is an ideal set. But as E is already a maximal ideal set we get that $a \in E$, which completes our proof. □

Thus, if the base-semantics satisfies the reinstatement property (and proposes one extension) then the ideal extension is already a complete set. To see that ideal semantics with base-semantics stable does not satisfy reinstatement consider the AF $F = (\{a, b, c\}, \{(a, b), (b, c), (c, c)\})$ which has no stable extension. We then have that $E^{ie}_{stb}(F) = \emptyset$ while the argument a is not attacked at all.

Furthermore for a base-semantics σ without the reinstatement property, the existence of a complete set S which is also ideal w.r.t. σ is not guaranteed. For example, consider $naive$ (or stg) semantics and the AF F from above. This AF has two naive (resp. stage) extensions, namely $\{a\}$ and $\{b\}$, and thus none of the arguments is skeptically accepted. But the empty set is not a complete extension, since $\mathcal{F}_F(\emptyset) = \{a\}$.

Let us now consider the related case of using complete semantics as a base-semantics. We have that the ideal extension w.r.t. complete semantics coincides with the grounded extension.

Proposition 4. *For any AF (A, R), $E^{ie}_{comp}((A, R)) = E_{gr}((A, R)) = grd(A, R)$.*

Proof. It is known that the set of skeptically accepted arguments w.r.t. complete semantics coincides with the grounded extension. Further as the grounded extension is also an admissible set, the assertion follows. □

Next as the base semantics we consider here, except stable, always yield a unique ideal extension, we investigate now how ideal extensions of different base semantics are related to each other w.r.t. $\subseteq$-inclusion. Caminada [24] has already shown that $E^{ie}_{pr}((A, R)) \subseteq E^{ie}_{sem}((A, R))$[5] holds and that there exist frameworks (A, R), such that $E^{ie}_{pr}((A, R)) \subset E^{ie}_{sem}((A, R))$. With the

[5] sem^{ie} is called the eager semantics in [24].

following results we give a full analysis of the $\subseteq$ relations between the ideal extensions w.r.t. the base semantics considered in this paper.

Theorem 1. *For arbitrary AFs (A, R) the following $\subseteq$-relations hold:*

$$\begin{array}{ccccccc} E^{IE}_{comp}(F) & \subseteq & E^{IE}_{resGr}(F) & \subseteq & E^{IE}_{pr}(F) & \subseteq & E^{IE}_{sst}(F) \\ & & & & \rotatebox[origin=c]{90}{$\subseteq$} & & \\ & & & & E^{IE}_{naive}(F) & \subseteq & E^{IE}_{stage}(F) \end{array}$$

Furthermore, these are all $\subseteq$-relations that hold for arbitrary AFs.

Proof. Most of these $\subseteq$-relations can be proven by considering the $\subseteq$-relation of the corresponding sets of skeptically accepted arguments. The relation $E^{ie}_{pr}((A, R)) \subseteq E^{ie}_{sem}((A, R))$ is an immediate consequence of the fact that each extension $E \in sem((A, R))$ is also contained $prf((A, R))$; thus each argument skeptically accepted w.r.t. prf-semantics is also skeptically accepted w.r.t. sem-semantics. Similarly, we have that each stg-extension is also a $naive$-extension and thus $E^{ie}_{naive}((A, R)) \subseteq E^{ie}_{stage}((A, R))$.

To show, $E^{ie}_{resGr}((A, R)) \subseteq E^{ie}_{pr}((A, R))$, we use the fact that for each $E \in prf((A, R))$ there exists a full resolution β such that $E \in prf((A, R \setminus \beta))$. Such a resolution β can be constructed as follows. For each symmetric attack $(a, b), (b, a)$ with $a \in E$ put $(b, a) \in \beta$ and make an arbitrary choice for all the other symmetric attacks. Clearly $E^+_R = E^+_{R\setminus\beta}$ and thus E is still preferred in $(A, R \setminus \beta)$.

Further the grounded extension of $(A, R\setminus\beta)$ is contained in each $S \in prf((A, R\setminus\beta))$. Hence if an argument is not skeptically accepted w.r.t. prf-semantics, then it is also not skeptically accepted w.r.t. $resGr$-semantics. By the observation that the grounded extension is contained in every resolution-based grounded extension [12], we get that $E_{gr}((A, R)) = E^{ie}_{comp}((A, R)) \subseteq E^{ie}_{resGr}((A, R))$.

It remains to show $E^{ie}_{naive}((A, R)) \subseteq E^{ie}_{pr}((A, R))$. This relation is quite surprising and in accordance with this the proof is also more involved. That is, it makes use of a elaborate characterisation of E^{idl}_{naive} and E^{idl}_{pr} which we prove to be correct later on.

Let $B \subseteq E^{idl}_{naive}((A, R))$. Ww have the following characterisation of the skeptically accepted arguments w.r.t. $naive$ semantics: $A_{sa} = \{x : (x, x) \notin R, \{x\}^\ominus \cup \{x\}^\oplus \subseteq \{y : (y, y) \in R\}$ (see Proposition 12). As by definition $B \subseteq A_{sa}$, we get that all arguments $a \in B^\ominus$ are self-conflicting, i.e. $(a, a) \in R$. Such self-conflicting arguments are not credulously accepted w.r.t. prf-semantics; hence, as B is admissible by definition, by Proposition 10 (C1) we have that $B \in E^{idl}_{pr}((A, R))$. Now as each ideal set w.r.t. $naive$-semantics is also an ideal set w.r.t. prf-semantics, we get the desired $\subseteq$-relation for the ideal extensions.

It remains to show that there are no further $\subseteq$-relations between the considered ideal extension that holds for arbitrary AFs. In what follows, we use the frameworks as given in Figure 2.4, to show that these relations do not hold.

$$\begin{array}{rcl} E^{ie}_{sem}(.) \not\subseteq E^{ie}_{pr}(.) & \text{as} & E^{ie}_{sem}(AF_a) = \{a\} \text{ while } E^{ie}_{pr}(AF_a) = \emptyset \\ E^{ie}_{pr}(.) \not\subseteq E^{ie}_{resGr}(.) & \text{as} & E^{ie}_{pr}(AF_b) = \{a\} \text{ while } E^{ie}_{resGr}(AF_b) = \emptyset \\ E^{ie}_{resGr}(.) \not\subseteq E^{ie}_{comp}(.) & \text{as} & E^{ie}_{resGr}(AF_c) = \{d\} \text{ while } E^{ie}_{comp}(AF_c) = \emptyset \end{array}$$

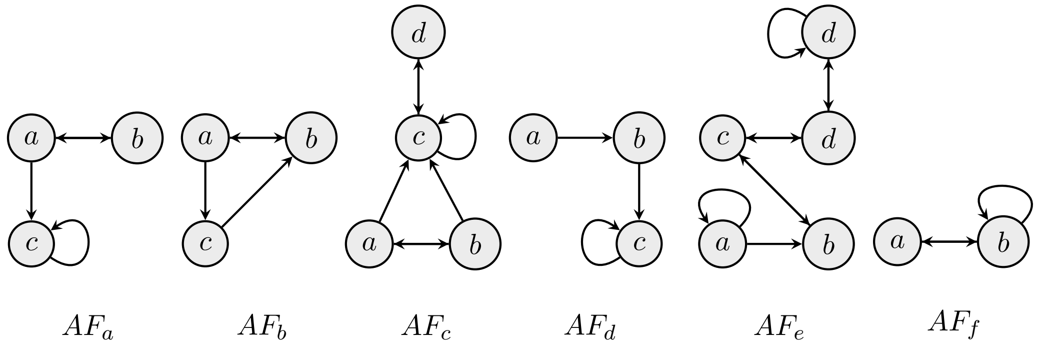


Figure 2.4: Counterexamples for possible ⊆-relations between ideal extensions.

$$\begin{array}{rcl}
E^{ie}_{stage}(.) \not\subseteq E^{ie}_{naive}(.) & \text{as} & E^{ie}_{stage}(AF_a) = \{a\} \text{ while } E^{ie}_{naive}(AF_a) = \emptyset \\
E^{ie}_{comp}(.) \not\subseteq E^{ie}_{stage}(.) & \text{as} & E^{ie}_{comp}(AF_d) = \{a\} \text{ while } E^{ie}_{stage}(AF_d) = \emptyset \\
E^{ie}_{naive}(.) \not\subseteq E^{ie}_{comp}(.) & \text{as} & E^{ie}_{naive}(AF_c) = \{a\} \text{ while } E^{ie}_{comp}(AF_c) = \emptyset \\
E^{ie}_{stage}(.) \not\subseteq E^{ie}_{sem}(.) & \text{as} & E^{ie}_{stage}(AF_e) = \{d\} \text{ while } E^{ie}_{sem}(AF_e) = \emptyset \\
E^{ie}_{pr}(.) \not\subseteq E^{ie}_{naive}(.) & \text{as} & E^{ie}_{pr}(AF_b) = \{a\} \text{ while } E^{ie}_{naive}(AF_b) = \emptyset \\
E^{ie}_{naive}(.) \not\subseteq E^{ie}_{resGr}(.) & \text{as} & E^{ie}_{naive}(AF_f) = \{a\} \text{ while } E^{ie}_{resGr}(AF_f) = \emptyset
\end{array}$$

The remaining cases follow by the transitivity of the ⊆-relation. □

Properties of AFs

Here we introduce important properties of AFs that refer to semantics.

The first considers stable semantics. As mentioned before there are AFs which propose no stable extensions, which can be interpreted as some kind of inconsistency in the AF w.r.t. stable semantics. This leads us to the following definition.

Definition 23. *We say that an $F = (A, R)$ is* stable-consistent *if it proposes at least one stable extension, i.e. $stb(F) \neq \emptyset$.*

In stable-consistent AFs we have that stable, semi-stable and stage semantics coincide. However as we will see later determining if an AF is stable-consistent is computationally hard.

Lemma 3. *Let F be a stable-consistent AF then $stb(F) = sem(F) = stg(F)$.*

Proof. We already know that $stb(F) \subseteq sem(F)$ (resp. $stb(F) \subseteq stg(F)$). It remains to show $stb(F) \supseteq sem(F)$ (resp. $stb(F) \supseteq stg(F)$). Given that there is a stable extension E with $E^+ = A$ we know that each range-maximal admissible (resp. conflict-free) set S satisfies $S^+ = A$ and thus is a stable extension. □

Next we consider AFs where preferred and stable semantics coincide.

Definition 24. *We say that an $F = (A, R)$ is* coherent *if stable and preferred semantics coincide, i.e. $prf(F) = stb(F)$.*

Again determining whether an AF is coherent is a computationally hard task – actually even harder than testing for stable-consistency [48]. Next we extend the traditional coherence term to semi-stable and stage semantics.

Lemma 4. *Let F be an coherent AF. Then F is stable-consistent and $prf(F) = stb(F) = sem(F) = stg(F)$.*

Proof. We have that arbitrary AFs propose at least one preferred extension and thus as F is coherent it proposes at least one stable extensions, i.e. F is stable-consistent. From the coherence we have that $prf(F) = stb(F)$ and from stable-consistency that $stb(F) = sem(F) = stg(F)$. □

Hence we have that an AF is coherent iff preferred, stable, semi-stable and stage coincide.

2.1.3 Computational Problems

We have that in general an argumentation semantics assigns several extensions to a single framework, but at the end of the day we want to make a conclusion about an argument (or a set of arguments). There are several ways to aggregate the acceptance status of an argument from the set of extensions, which mirrors different levels of scepticism. First it is quite clear that an argument which is in no extension at all should not be accepted, but for some (brave) people it might be alright to accept an argument if it appears in just one extension, this is what we will call *credulous reasoning*. On the other hand cautious people may demand that an argument is in all extensions, we refer to this as *skeptical reasoning*. Really cautious (or coward) people may not only require that the argument itself es skeptical accepted but also the arguments defending the argument, which gives rise to *ideal reasoning*. That is, we have three different reasoning modes for abstract argumentation[6], credulous, skeptical and ideal reasoning.

These reasoning modes give rise to the following computational problems for an argumentation semantics σ.

- *Credulous Acceptance $Cred_\sigma$*: Given AF $F = (A, R)$ and an argument $a \in A$. Is a contained in some $S \in \sigma(F)$?
- *Skeptical Acceptance $Skept_\sigma$*: Given AF $F = (A, R)$ and an argument $a \in A$. Is a contained in each $S \in \sigma(F)$?
- *Ideal Acceptance $Ideal_\sigma$*: Given AF $F = (A, R)$ and an argument $a \in A$. Is a contained in the ideal extension w.r.t. base semantics σ?

[6] Wu and Caminada [105] recently proposed a more fine-grained concept of a justification status of arguments for labeling-based semantics. The computational properties of their approach has been studied in [53], exploiting results for extension-bases semantics and the introduced reasoning modes.

If an AF has no stable extensions, according to our definition of skeptical acceptance, all arguments are skeptically accepted. This may be unwanted and hence one might consider a variation of the skeptical acceptance problem, let us call it $\mathit{Skept'}_{stb}$, asking whether an argument is contained in all extensions and there exists at least one extension.

Beside these reasoning problems there are also several other computational problems in the field of abstract argumentation. In this work we consider the three most prominent ones of them. First of all one might be interested in verifying given extensions, which may come from another agent or potential corrupted file, or simple as part of a (guess & check) reasoning algorithm.

- *Verification of an extension* Ver_σ: Given AF $F = (A, R)$ and a set of arguments $S \subseteq A$. Is $S \in \sigma(F)$?

Another task is deciding whether an AF provides any coherent conclusion. That can be deciding whether it has at least one extensions, in the case of stable semantics, or whether it has an extension different from the emptyset, for all the other semantics under our consideration.

- *Existence of an extension* Exists_σ: Given AF $F = (A, R)$. Is $\sigma(F) \neq \emptyset$?
- *Existence of a nonempty extension* $\mathit{Exists}_\sigma^{\neg\emptyset}$: Given AF $F = (A, R)$. Does there exist a set $S \neq \emptyset$ such that $S \in \sigma(F)$?

Now considering these problems w.r.t. the introduced semantics gives rise to a large number of problems. But as several semantics are closely related to each other some of these problems turn out to be just different formulations of the same problem. The following proposition identifies such cases.

Proposition 5. *The following holds:*

- $\mathit{Cred}_{cf} = \mathit{Cred}_{naive}$
- $\mathit{Cred}_{adm} = \mathit{Cred}_{com} = \mathit{Cred}_{prf}$
- $\mathit{Cred}_{grd} = \mathit{Skept}_{grd} = \mathit{Skept}_{com} = \mathit{Ideal}_{com} = \mathit{Ideal}_{grd}$
- $\mathit{Exists}_{stb} = \mathit{Exists}_{stb}^{\neg\emptyset}$ [7]
- $\mathit{Exists}_{adm}^{\neg\emptyset} = \mathit{Exists}_{com}^{\neg\emptyset} = \mathit{Exists}_{prf}^{\neg\emptyset} = \mathit{Exists}_{sem}^{\neg\emptyset}$
- $\mathit{Exists}_{cf}^{\neg\emptyset} = \mathit{Exists}_{naive}^{\neg\emptyset} = \mathit{Exists}_{stg}^{\neg\emptyset}$

Proof. $\mathit{Cred}_{cf} = \mathit{Cred}_{naive}$: Clearly an argument is contained in a conflict-free set iff it is contained in a $\subseteq$-maximal conflict-free set.

$\mathit{Cred}_{adm} = \mathit{Cred}_{com} = \mathit{Cred}_{prf}$: By Lemma 1 we have that each admissible set is contained in a complete extension, hence $\mathit{Cred}_{adm}(F, a) \Rightarrow \mathit{Cred}_{com}(F, a)$. As each complete extension is also admissible we further have that each complete extension is contained in some

[7] Provided that we only consider AFs with at least one argument.

preferred extension, thus $\mathit{Cred}_{\mathit{com}}(F, a) \Rightarrow \mathit{Cred}_{\mathit{prf}}(F, a)$. Finally we have that each preferred extension is an admissible set and thus $\mathit{Cred}_{\mathit{prf}}(F, a) \Rightarrow \mathit{Cred}_{\mathit{adm}}(F, a)$.

$\mathit{Cred}_{\mathit{grd}} = \mathit{Skept}_{\mathit{grd}} = \mathit{Skept}_{\mathit{com}} = \mathit{Ideal}_{\mathit{com}} = \mathit{Ideal}_{\mathit{grd}}$: First as grounded is a unique status semantics we clearly have that $\mathit{Cred}_{\mathit{grd}} = \mathit{Skept}_{\mathit{grd}}$. Further we have that the grounded extension is the $\subseteq$-minimal complete extension and thus coincides with the skeptical accepted arguments w.r.t. complete semantics, hence $\mathit{Skept}_{\mathit{grd}} = \mathit{Skept}_{\mathit{com}}$. Finally we have that the grounded extension (resp. skeptical accepted arguments w.r.t. com) is an admissible set and thus the ideal extension coincides with the grounded extension.

$\mathit{Exists}_{\mathit{stb}} = \mathit{Exists}^{\neg\emptyset}_{\mathit{stb}}$: We have that the emptyset is never a stable extension, as it does not attack anything and our AF contains at least one argument.

$\mathit{Exists}^{\neg\emptyset}_{\mathit{adm}} = \mathit{Exists}^{\neg\emptyset}_{\mathit{com}} = \mathit{Exists}^{\neg\emptyset}_{\mathit{prf}} = \mathit{Exists}^{\neg\emptyset}_{\mathit{sem}}$: By Lemma 1 we have that each admissible set is contained in a complete extensions, hence $\mathit{Exists}^{\neg\emptyset}_{\mathit{adm}}(F) \Rightarrow \mathit{Exists}^{\neg\emptyset}_{\mathit{com}}(F)$. As each complete extension is still admissible it is contained in some preferred extension, hence $\mathit{Exists}^{\neg\emptyset}_{\mathit{com}}(F) \Rightarrow \mathit{Exists}^{\neg\emptyset}_{\mathit{prf}}(F)$. If there exists a nonempty preferred extension we have an admissible set with nonempty range. Then each semi-stable extension has nonempty range and thus contains at least one argument, i.e. $\mathit{Exists}^{\neg\emptyset}_{\mathit{prf}}(F) \Rightarrow \mathit{Exists}^{\neg\emptyset}_{\mathit{sem}}(F)$. Finally we have that each semi-stable extension is admissible and therefore $\mathit{Exists}^{\neg\emptyset}_{\mathit{sem}}(F) \Rightarrow \mathit{Exists}^{\neg\emptyset}_{\mathit{adm}}(F)$.

$\mathit{Exists}^{\neg\emptyset}_{\mathit{cf}} = \mathit{Exists}^{\neg\emptyset}_{\mathit{naive}} = \mathit{Exists}^{\neg\emptyset}_{\mathit{stg}}$: By definition each conflict-free set is contained in a naive set, hence $\mathit{Exists}^{\neg\emptyset}_{\mathit{cf}}(F) \Rightarrow \mathit{Exists}^{\neg\emptyset}_{\mathit{naive}}(F)$. If there exists a nonempty naive set we have a conflict-free set with nonempty range. Then each stage extension has nonempty range and hence contains at least one argument, i.e. $\mathit{Exists}^{\neg\emptyset}_{\mathit{naive}}(F) \Rightarrow \mathit{Exists}^{\neg\emptyset}_{\mathit{stg}}(F)$. Finally we have that each stage extension is a conflict-free set and therefore $\mathit{Exists}^{\neg\emptyset}_{\mathit{stg}}(F) \Rightarrow \mathit{Exists}^{\neg\emptyset}_{\mathit{cf}}(F)$. □

We observe that we did not use any kind of (complexity theoretic) reduction in the above proposition. Hence two problems shown to be equivalent can be seen as different formulations of the same problem, that is an instance is valid for one problem iff it is for the other problem, without modifying anything in the instance.

2.2 Computational Complexity Theory

Here we briefly recapitulate the most important concepts of computational complexity theory (for a comprehensive introduction see e.g. [97]) and the complexity classes relevant for this work. We start with introducing different kind of abstract machine models, which underlie the definitions of our complexity classes.

2.2.1 Abstract Machines

> *Despite its weak and clumsy appearance, the Turing machine can simulate arbitrary algorithms with inconsequential loss of efficiency.* Papadimitriou [97]

One important concept for computability and complexity studies is the underlying machine model; here we consider Turing machines, dated back to Turing [102]. In the following we will distinguish deterministic, non-deterministic and oracle Turing machines.

We start with deterministic Turing machines. The intuition behind a Turing machine is a machine having a finite state register, several tapes (in the simplest case just one) with symbols written on it and a read/write head for each tape to handle the symbols on the tapes. In one computation step this machine reads all symbols under the read/write heads, may change them, may move the heads one step, and may change the state stored in the register. A program for such a machine defines for a given state and symbols read by the heads, how to modify the read symbols, how to move the cursors, and the next state of the machine.

The following definition gives a formalisation of this intuition:

Definition 25. *A (deterministic) k-string Turing machine (TM) is a tuple $M = \langle S, \Sigma, \delta, s \rangle$, with*

- *S a finite set of states*
- *Σ the alphabet of M - a finite set of symbols*
- *δ a transition function : $\delta : S \times \Sigma^k \mapsto S \cup \{\text{“yes”}, \text{“no”}\} \times \Sigma^k \times \{\rightarrow, \leftarrow, -\}^k$*
- *$s \in S$ the initial state*

The transition function δ plays the role of the program of the Turing machine. A computation of such a machine starts in the initial state and an initial configuration of the tapes and read/write-heads, encoding the input. In the following we call such a combination of a state and a tape configuration, the configuration of a machine. In each computation step the machine modifies the configuration according to the function δ and eventually holds if one of the halting states “yes”, “no” is reached. In the first case we say the machine accepts the input otherwise we say it rejects the input.

While this is a quite simple concept it is commonly believed that every effectively calculable function can be computed with a Turing machine, this is referred to as Church-Turing thesis. Thus Turing machines are the appropriate tool to classify computable functions and as it turns out also for classifying algorithms w.r.t. computational costs. To this end we introduce generalisations of Turing machines, namely non-deterministic Turing machines and oracle Turing machines.

Definition 26. *A non-deterministic k-string Turing machine (NTM) is a tuple, $N = \langle S, \Sigma, \delta, s \rangle$ with:*

- *S a finite set of states*
- *Σ the alphabet of M - a finite set of symbols*
- *δ a transition relation: $\delta \subseteq (S \times \Sigma^k) \times \left[S \cup \{\text{“yes”}, \text{“no”}\} \times \Sigma^k \times \{\rightarrow, \leftarrow, -\}^k\right]$*
- *$s \in S$ the initial state*

The main difference between deterministic and non-deterministic machines is that: for a deterministic machine each configuration has at most one possible computation step, given by the function δ, while for non-deterministic machines each configuration has several possible computation steps, given by the relation δ. Hence when starting with some input configuration we get exactly one computation for deterministic machines and in general several for non-deterministic machines. We say that a non-deterministic machine accepts an input if at least one of the possible computations accepts it and respective that it rejects an input only if all possible computations rejects it.

Next let us consider so-called oracle machines. By a $\mathcal{C}$-oracle machine we mean a Turing machine which can access an oracle that decides a given (sub)-problem $\mathcal{C}$ within one step.

Definition 27. *For a language $\mathcal{L}$, an $\mathcal{L}$-*oracle Turing machine *is a (non-deterministic) k-string Turing machine with an designated query string and three special states $q_?$, q_{yes} and q_{no}. The state $q_?$ is excluded from the function (resp. relation) δ. The transition step for a configuration with state $q_?$ is handled by the $\mathcal{L}$-oracle. We have that the state changes to q_{yes} if the current string on the query string is in $\mathcal{L}$ and q_{no} otherwise. The strings as well as the heads positions are not changed in this step.*

Towards complexity measures we define time and space used by a computation on a Turing machine. Let us start with time. For a deterministic computation we say that the computation time, the time required by the algorithm, is simple the number of computation steps needed, starting from the initial configuration, until a halting state is reached. For non-deterministic computations we define the computation time as the minimum time of all accepting computations if such a computation exits and as the maximum time of all rejecting computations otherwise. These definitions pass over to oracle machines. We mention that we count an oracle call just as one step, that is the computation time for deciding whether the query string is in $\mathcal{L}$ or not, does not add to the computation time of the oracle-machines computation.

To define the space used by Turing machine algorithms we consider a special kind of k-string machine, namely these with an designated input and a designated output string. The intuition behind this is that the space occupied by input or output should not be considered as space used by the algorithm. On the other side we want to be sure that the algorithm does not hide computational costs in the input or output. Thus we have the following restrictions: The machine is not allowed to write on the input string (read only head), is not allowed to read from the output string (write only head) and in the initial configuration all strings except the input string are empty. The space used by a deterministic algorithm on a string is the number of string positions visited by the read/write-heads. The space used by a deterministic algorithm is the sum over the space used on the read/write strings. As before for non-deterministic machines we define the used space as the minimum space of all accepting computations if such a computation exits and as the maximum space used by any rejecting computations otherwise. Again these definitions pass over to oracle machines.

We will use these concepts of time and space to actually define complexity classes, but before doing so we have to introduce some basic concepts of complexity theory.

2.2.2 Basic Concepts

First of all we have to clarify that computational complexity theory deals with the *asymptotic worst case algorithmic complexity of problems*. A few more words about this are appropriate. By algorithmic complexity we mean the (time/space) resources used by an (Turing machine) algorithm solving the problem. Further we consider the complexity of the hardest problem instances as the complexity of the problem, i.e. we consider some kind of worst case complexity. Finally we only analyse the asymptotic behaviour of problems, i.e. how fast the resource requirements increase with the input size.

First we want to formalise the concept of a complexity theoretic problem:

Definition 28. *In complexity theory a problem specification, or problem for short, consists of two parts. A criterion that defines an infinite set of instances and a question on these instances.*

Usually the question is such that it can be either answered by yes or no. In this case we say that the problem is a decision problem. An example for a decision problem is the REACHABILITY problem in graph-theory. Here we have as instance a graph[8] $G = (V, E)$ and two nodes $v_1, v_2 \in V$ and the question is if there is a path from v_1 to v_2.

To classify a problem's complexity we use the idea of reductions to order problems w.r.t. their complexity and identify problems with the same complexity. The idea behind this is that if a problem A can be easily reduced to another a problem B we can simple apply this reduction and then us an algorithm for the second problem to solve any instance of the original problem A. Hence one says that A is as most as hard as B and write $A \leq B$. In this work, to reduce decision problems, we use the concept of so called many-one reductions.

Definition 29 (Reduction)**.** *A decision problem A is many-one reducible to a decision problem B if there is a reduction function R such that for each problem instance x of A we have that x is a yes instance of A iff $R(x)$ is a yes instance of B. We then call R a many-one reduction from A to B.*

As we will only consider many-one reductions in the following, we omit the "many-one" and just talk about reductions. Next we consider the computational power of such reductions.

Definition 30 (log-Reduction)**.** *A decision problem A is* log*-reducible to a decision problem B if there is a reduction function R computable by a deterministic Turing machine in space $O(\log |x|)$ for each problem instance x of A.*

Definition 31 (P-Reduction)**.** *A decision problem A is P-reducible to a decision problem B if there is a polynomial $p(.)$ and a reduction function R computable by a deterministic Turing machine in time $O(p(|x|))$ for each problem instance x of A.*

It is well known that each log-Reduction is also a P-Reduction, but it is commonly believed that the converse does not hold (although it is not known yet). Further both reducibility concepts, log-reducibility and P-reducibility, are transitive, i.e. if there are log-/P-reductions from A to B and from B to C then there is also a log-/P-reduction from A to B.

Let us briefly point out things that can be done and things that can not be done within log-space. First obviously we can not use copies of (large parts of) the input for our calculations. But we can store a fixed number of integer counters as long as the values are polynomial bounded in the input size (using binary encodings). Moreover one can also implement a fixed number of cursors to parts of the input, for instance by using integer counters. So when giving log-space

[8] For the formal definitions see Section 2.3.

algorithms what we basically do is using a fixed number of cursors and and a fixed number of counters to handle arbitrary instance.

As different reductions might yield different hardness relations we have to clarify which kind of reduction should be used in which case. We will come back to this point in Section 2.2.4.

2.2.3 Complexity Classes

Here we briefly review the complexity classes used in this work and their relations. In the following we say that a machine decides a problem in polynomial time (resp. space) if there exists an polynomial $p(x)$ such that the machine decides to problem in time (resp. space) $O(p(|x|))$. Moreover we will use the notion of exponential and logarithmic time / space in a similar manner.

We start with the complexity classes based on deterministic machines and different time bounds for the computation.

Definition 32. *The class P (polynomial time) is the class of problems that can be decided by a deterministic Turing machine in polynomial time.*

Definition 33. *The class EXPTIME (exponential time) is the class of problems that can be decided by a deterministic Turing machine in exponential time.*

We consider problems in the class P to be (computationally) tractable. Towards a more fine-grained complexity scale we (additionally) use space bounds.

Definition 34. *The class L (deterministic logarithmic space) is the class of problems that can be decided by a deterministic Turing machine in logarithmic space (and polynomial time).*

The class L is the weakest complexity class in our considerations. Next we consider another class based on space bounds, but which in contrast L characterises intractable problems.

Definition 35. *The class PSPACE (deterministic polynomial space) is the class of problems that can be decided by a deterministic Turing machine in polynomial space (and exponential time).*

While the power and cost of the introduced deterministic complexity classes are quite clear, it turned out that they do not match the exact complexity of many important combinatorial problems (under usual complexity theoretic assumptions).

But most of these problems can be easily solved by non-deterministic algorithms and hence the complexity of these problems is captured by non-deterministic complexity classes, with NP and coNP being the most prominent.

Definition 36. *The class NP (non-deterministic polynomial time) is the class of problems that can be decided by a non-deterministic Turing machine in polynomial time.*

It is strongly believed, but not shown yet, that non-deterministic time complexity is not closed under complementing. This is caused by the asymmetry between “yes” and “no” instances in the non-deterministic acceptance criterion. Hence one might be interested in the complementary classes, in particular in the class coNP:

Definition 37. *The class coNP is the class of problems, where the negated question*[9] *can be decided by a non-deterministic Turing machine in polynomial time.*

Next we define the class of problems that are, in some sense, combinations of NP and coNP problems. We start with a class of problems that are conjunctions of a NP and a coNP problem.

Definition 38. *A decision problem L is in the class D^P iff L can be characterised as $L_1 \cap L_2$ for decision problems $L_1 \in$ NP and $L_2 \in$ coNP.*

Next we consider complexity classes defined via oracle machines (see Definition 27) and using the above complexity classes. To this end let $\mathcal{C}$ denote some complexity class. By a $\mathcal{C}$-oracle machine we mean a oracle Turing machine which can access an oracle that decides a given (sub)-problem in $\mathcal{C}$ within one step. We denote the class of problems decidable in polynomial time when using such a $\mathcal{C}$-oracle machine, as $\mathrm{P}^{\mathcal{C}}$ if the underlying Turing machine is deterministic and $\mathrm{NP}^{\mathcal{C}}$ (resp. $\mathrm{coNP}^{\mathcal{C}}$) if the underlying Turing machine is non-deterministic. Moreover we consider deterministic oracle machines where the number of allowed oracle calls is bounded by a function $f(|x|)$, and denote the corresponding complexity classes as $\mathrm{P}^{\mathcal{C}[f(|x|)]}$.

We now turn to concrete complexity classes.

Definition 39. *Σ_2^P denotes the class NP^{NP}, i.e. the problems which can be decided by a non-deterministic polynomial time algorithm that has access to an NP-oracle.*

As Σ_2^P is a non-deterministic class we can also define the complementary class Π_2^P.

Definition 40. *Π_2^P denotes the class $coNP^{NP}$, i.e. the problems where the complement can be decided by a non-deterministic polynomial time algorithm that has access to an NP-oracle.*

Let us briefly mention that the classes Σ_2^P, Π_2^P do not stand-alone, but are part of the so called polynomial hierarchy [97, chapter 17]. The idea behind this hierarchy is, starting from $\Sigma_1^P = \mathrm{NP}$ and $\Pi_1^P = \mathrm{coNP}$, recursively defining complexity classes Σ_i^P, Π_i^P as $\Sigma_i^P = \mathrm{NP}^{\Sigma_{i-1}^P}$ and $\Pi_i^P = \mathrm{coNP}^{\Sigma_{i-1}^P}$. We will consider NP and coNP as the first level and Σ_2^P, Π_2^P as the second level of the polynomial hierarchy.

Now one can consider the analogon of the class D^P on the second level of the polynomial hierarchy:

Definition 41. *A problem L is in the class D_2^P iff L can be characterised as $L_1 \cap L_2$ for $L_1 \in \Sigma_2^P$ and $L_2 \in \Pi_2^P$.*

Next we consider a complexity class where the number of oracle calls is bounded.

Definition 42. *Θ_2^P denotes the class $P^{NP[\log(|x|)]}$, i.e. the problems which can be decided by a deterministic polynomial time algorithm that is allowed to make a logarithmic number (w.r.t. input size) of NP-oracle calls.*

[9] The answer to the negated question is YES iff the answer to the original question is NO and vice versa.

An alternative characterisation for a problem to be Θ_2^P is that it can be solved by a deterministic algorithm which is allowed to make $O(n)$ non-adaptive calls to the NP-oracle ($\Theta_2^P = \mathrm{P}_{\|}^{\mathrm{NP}}$). In other words the exact formulation of a query does not depend on the answers of the previous queries, and thus the answers to all queries can be computed in parallel (see, e.g., [67]).

Finally we give an overview of relations between the complexity classes used in this paper:

$$\mathrm{L} \subseteq \mathrm{P} \subseteq \begin{array}{c}\mathrm{NP}\\ \mathrm{coNP}\end{array} \subseteq \mathrm{D}^P \subseteq \Theta_2^P \subseteq \begin{array}{c}\Sigma_2^P\\ \Pi_2^P\end{array} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$$

When choosing a kind of reduction one usually wants that the complexity classes under ones considerations are closed under these reductions. That is if one can reduce a problem A to a problem in the complexity class $\mathcal{C}$ then also the problem A should be in the complexity class $\mathcal{C}$. We have that all the complexity classes introduced above are closed under $\log$-reductions and all except L are closed under P-reductions.

2.2.4 Complete Problems

In complexity analysis we are not that dependent on the actual definition of a complexity class, but on “representative” problems for the class which are suitable for (nice) reductions. The concept making a problem “representative” for a complexity class is the concept of completeness. Next we give the definition for a problem being complete for a complexity class.

Definition 43. *We say that a problem B is hard under τ-reductions (where τ stands for an arbitrary type of reduction) for a complexity class $\mathcal{C}$, if each problem $A \in \mathcal{C}$ can be τ-reduced to B. If further $B \in \mathcal{C}$ we say that B is $\mathcal{C}$-complete under τ-reductions.*

We mention that any problem in the class P and in particular those in the class L would be complete for P under P-reductions, because the complexity class does not add any computational power to the reductions. That is one can solve the actual instance of a problem in P within the reduction and then reduce it to a trivial accepting (resp. rejecting) instance. Hence in the following we will use $\log$-reductions when talking about P-completeness and P-reductions when we consider complexity classes beyond P. However, probably most of the P-reductions we present in this work are also $\log$-reductions.

To show that a problem A is $\mathcal{C}$-hard, by the transitivity of reductions, it suffices to reduce a $\mathcal{C}$-hard problem to A, instead of all problems in $\mathcal{C}$. Hence for the complexity analysis in this work we are interested in complete problems for the complexity classes under our considerations. To this end we introduce the “canonical” complete problems for the complexity classes used in this work.

P-complete problems: Here we present two P-complete problems, i.e. HORNSAT and the circuit value problem (CVP); a nice compendium of P-complete problems can be found in [76].

Towards our first P-complete problem we introduce the concept of definite HORN-clauses and definite HORN-formulas. A definite HORN-clause c is the disjunction over literals from a countable domain U such that c contains exactly one positive literal. A definite HORN-formula is the conjunction over definite HORN-clauses.

For example consider the definite HORN-formula $\varphi = x \wedge (\neg x \vee \neg y \vee z) \wedge (\neg y \vee \neg z \vee x)$. A more convincing way to denote definite HORN-formulas is as set of clauses and moreover denoting clauses as (logical equivalent) rules. Thus, our example formula φ can be denoted as $\varphi = \{\rightarrow x, x \wedge y \rightarrow z, y \wedge z \rightarrow x\}$. It is well known that a definite HORN-formula has a unique minimal model which can be computed in polynomial time.

Definition 44. *The* definite Horn satisfiability *problem (HORNSAT) is: Given a definite HORN formula φ and an atom x. Deciding whether x is in the minimal model of φ.*

Theorem 2 ([83])**.** *Definite HORN SAT is P-complete.*

Another source for P-complete problems are boolean circuits (see, e.g., [73, 87]) . Let us start with a formal definition of these circuits.

Definition 45. *A* (boolean) circuit *β is a sequence $(\beta_i)_{1 \leq i \leq m}$ where each β_i is either a variable $x \in X$ or*

$$\beta_i = \begin{cases} \wedge(j,k) & \textit{with } j,k < i \\ \vee(j,k) & \textit{with } j,k < i \\ \neg(j) & \textit{with } j < i \end{cases}$$

We call the β_i the gates of the circuit. A circuit is monotone if it does not contain a $\neg(j)$ gate.

An assignment a for β is a function $a : X \mapsto \{true, false\}$. The value $v(\beta_i, a)$ of a gate β_i w.r.t. an assignment a is recursively defined as follows:

$$v(\beta_i, a) = \begin{cases} a(\beta_i) & \textit{if } \beta_i \textit{ is a variable} \\ a(\beta_j) \wedge a(\beta_k) & \textit{if } \beta_i = \wedge(j,k) \\ a(\beta_j) \vee a(\beta_k) & \textit{if } \beta_i = \vee(j,k) \\ \neg(a(\beta_j)) & \textit{if } \neg(j) \end{cases}$$

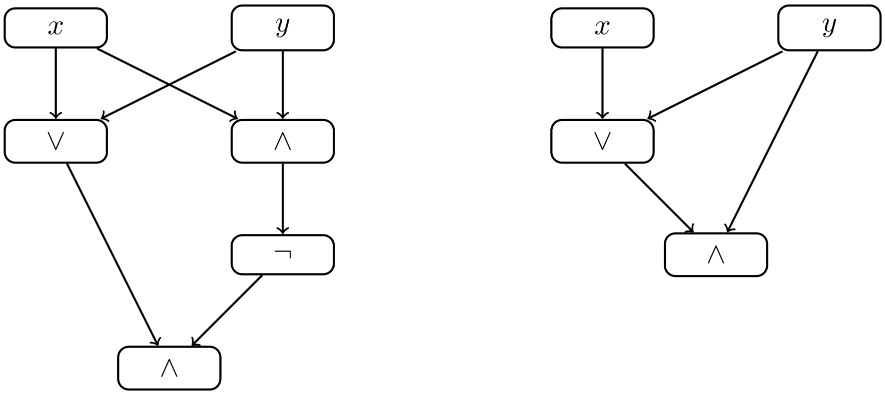
The value $v(\beta, a)$ of the circuit β w.r.t. an assignment a is defined as $v(\beta, a) = v(\beta_m, a)$.

Definition 46. *The* circuit value problem *(CVP) is: Given a circuit β and an assignment $a(.)$ for β. Deciding whether $v(\beta, a) = true$ or not.*

Definition 47. *The* monotone circuit value problem *(MCVP) is: Given a monotone circuit β and an assignment $a(.)$ for β. Deciding whether $v(\beta, a) = true$ or not.*

Theorem 3. *[73, 87] CVP and MCVP are P-complete (under* log*-reductions).*

Complete problems for the first level of the polynomial hierarchy: One of the most prominent NP-complete problems is deciding whether a propositional formula is satisfiable. Here we use the version where the formulas are in conjunctive normal form (CNF). We denote such a formula in CNF φ as a collection C of so called clauses, where a clause is a set of literals build from atoms in a countable domain U. A clause is interpreted as a disjunction over its literals while the set C is interpreted as conjunction over the clauses. For a variable y, we use $\bar{y}$ to



Boolean Circuit

$(x, y, \vee(\beta_1, \beta_2), \wedge(\beta_1, \beta_2), \neg(\beta_4), \wedge(\beta_3, \beta_5))$

Monotone Boolean Circuit

$(x, y, \vee(\beta_1, \beta_2), \wedge(\beta_2, \beta_3))$

Figure 2.5: Illustration of boolean circuits.

represent its negation. We say that φ is satisfiable if there exists an interpretation $I \subseteq U$ such that for each $c \in C$,

$$\big(I \cup \{\bar{x} \mid x \in U \setminus I\}\big) \cap c \neq \emptyset. \tag{2.1}$$

We call such an $I \subseteq U$ satisfying Equation 2.1 a model of φ. The corresponding decision problem is the following:

Definition 48. *The* satisfiability problem *(SAT) is: Given a formula in CNF φ. Deciding whether φ is satisfiable.*

Sometimes we also need restricted versions of SAT, where we only consider special kind of propositional formulas but still have the full complexity.

Definition 49. *The* monotone satisfiability problem *(MSAT) is: Given a formula in CNF φ where each clause either contains only positive or only negative literals. Deciding whether φ is satisfiable.*

We can also consider the dual problems:

Definition 50. *The* unsatisfiability problem *(UNSAT) is: Given a formula in CNF φ. Deciding whether φ is unsatisfiable.*

Definition 51. *The* monotone unsatisfiability problem *(MUNSAT) is: Given a formula in CNF φ where each clause either contains only positive or only negative literals.. Deciding whether φ is unsatisfiable.*

The complexity of these problems is given by the following theorem:

Theorem 4. *SAT is NP-complete. UNSAT is coNP-complete. MSAT is NP-complete. MUNSAT is coNP-complete.*

Towards an D^P-hard problem we consider the SAT-UNSAT problem.

Definition 52. *The* combinded satisfiability - unsatisfiability problem *(SAT-UNSAT) is: Given two formulas in CNF φ, ψ. Deciding whether φ is satisfiable and ψ is unsatisfiable.*

Theorem 5. *SAT-UNSAT is D^P-complete.*

Complete problems for the second level of the polynomial hierarchy: Towards complete problems for the second level of the polynomial hierarchy we require particular classes of quantified Boolean formulas (QBFs) which we introduce next. A $QBF^2_\exists$ formula is of the form $\exists Y \forall Z C$, while a $QBF^2_\forall$ formula is of the form $\forall Y \exists Z C$, where Y and Z are sets of propositional atoms from a countable domain U, and C is a collection of clauses over literals built from atoms $Y \cup Z$. We say that a QBF $\exists Y \forall Z C$ is true iff, there exists $I_Y \subseteq Y$ such that for each $I_Z \subseteq Z$ and $c \in C$,

$$\big(I_Y \cup I_Z \cup \{\bar{x} \mid x \in (Y \cup Z) \setminus (I_Y \cup I_Z)\}\big) \cap c \neq \emptyset. \tag{2.2}$$

Analogous we say that a QBF $\forall Y \exists Z C$ is true iff, for each $I_Y \subseteq Y$ there exists an $I_Z \subseteq Z$, such that for each $c \in C$,

$$\big(I_Y \cup I_Z \cup \{\bar{x} \mid x \in (Y \cup Z) \setminus (I_Y \cup I_Z)\}\big) \cap c \neq \emptyset. \tag{2.3}$$

Example 2. *Consider the $QBF^2_\forall$ formula*

$$\Phi = \forall y_1 y_2 \exists z_3 z_4 \{\{y_1, y_2, z_3\}, \{\bar{y}_2, \bar{z}_3, \bar{z}_4)\}, \{\bar{y}_1, \bar{y}_2, z_4\}\}$$

which we will use as a running example. It can be easily checked that Φ is true.

Note that we can also encode the SAT and the UNSAT problem as a QBF problem. To this end let Y be the set of atoms occurring in a CNF-formula φ. Then we can write $\exists Y \varphi(Y)$ to encode the satisfiability problem and $\forall Y \neg\varphi(Y)$ to encode the unsatisfiability problem.

Definition 53. *The problem $QSAT^2_\exists$ is: Given a $QBF^2_\exists$ formula φ. Deciding whether φ is true.*

Definition 54. *The problem $QSAT^2_\forall$ is: Given a $QBF^2_\forall$ formula φ. Deciding whether φ is true.*

Theorem 6. *$QSAT^2_\exists$ is Σ^P_2-complete and $QSAT^2_\forall$ is Π^P_2-complete.*

Finally we consider complete problems based on subset-minimal models of a propositional formula.

Definition 55. *The problem Minimal Model Satisfiability (MINSAT) is: Given a propositional formula φ in CNF and an atom x. Deciding whether x is contained in some $\subseteq$-minimal model of φ.*

Theorem 7. *[66] MINSAT is Σ^P_2-complete.*

In contrast we have that deciding whether an atom x is in each $\subseteq$-minimal model is just coNP-complete as one does not have to consider $\subseteq$-minimality when looking for a counter model.

2.2.5 Parameterized Complexity

Classical complexity theory deals with the complexity of problems w.r.t. the size of the instance. However often the complexity of a problem do not mainly depend on the size of an instance but on some (structural) properties of the instance. That is we can solve huge instances efficiently as long some property is satisfied or the obstacles in the structure are bounded independent of the size. The field of parameterized complexity theory deals with this observation. Here we just briefly introduce the concepts we actually need in this work; for comprehensive introductions to the field the interested reader is referred to the books of Flum and Grohe [69] and Niedermeier [93].

In parameterized complexity theory one considers parameterized problems, i.e. in addition to the ordinary problem description such a problem has designated parameter which is instantiated by each problem instance.

Definition 56. *A parameterized decision problem contains of a specification of problem instances, an integer parameter and a decision question on these instances. The parameter may occur in the description of the instance, in the question, or in both.*

An example for a parameterized problem is given a graph G and an integer parameter k deciding whether G has a clique of size k.

Definition 57. *A parameterized (decision) problem is fixed-parameter tractable (or in FPT) if it can be determined in time $f(k) \cdot |x|^{O(1)}$ for a computable function f.*

Now given that a problem is in FPT and just consider those instances where the parameter is bounded by some constant then we can decide this instances with a polynomial time algorithm. Only the constants in the polynomial time bound are affected by the parameter, but not the order of the polynomial.

There is also a weaker form of tractability w.r.t. a parameter allowing the order of the polynomial to depend on the parameter, which is actually not used in this work but might give further insights on the value of FPT results.

Definition 58. *A parameterized (decision) problem is in the class XP if it can be determined in time $f(k) \cdot |x|^{g(k)}$ for computable functions f, g.*

A problem in XP can be solved in polynomial time if we bound the parameter, but distinguishing it from FPT the order of the polynomial may highly depend on the bound of the parameter.

Let us briefly present the relations between the classes FPT, XP and P:

$$\text{P} \subseteq \text{FPT} \subseteq \text{XP}$$

When considering in principle unparameterized problems and talking about FPT we have to mention the used parameter explicitly. Thus we say a problem P is fixed-parameter tractable w.r.t. the parameter k iff the corresponding parameterized problem (P, k) is fixed-parameter tractable.

2.3 Graph Parameters

In this section we consider graph parameters measuring structural properties of graphs and have been used to obtain FPT-results for several graph problems. Clearly, as argumentation frameworks can be interpreted as directed graphs, these parameters also apply to AFs.

Firstly we introduce a formal definition of directed graphs and useful notation from graph theory. Secondly we introduce three parameters, i.e. tree-width, clique-width and cycle-rank, for directed graphs and discuss their properties and relations between them. Then we give a definition of monadic second order logic (MSO) which we finally use to present quite strong meta-theorems for classifying problems to be fixed-parameter tractable w.r.t. tree-width or clique-width.

2.3.1 Directed Graphs

Here we briefly present the basic concepts concerning directed graphs, for an comprehensive overview of the field see [4]. First of all we need a formal definition.

Definition 59. *A* directed graph *G is a pair (V, E) where V is the set of* vertices *and $E \subseteq V \times V$ is the edge relation of the graph.*

It is easy to see that the definition of abstract argumentation frameworks and directed graphs are isomorphic, i.e. arguments correspond to vertices and attacks to edges. Hence, as mentioned before, one can interpret AFs as directed graphs and use notation and methods from graph theory.

Next we introduce some useful notation from graph theory, which we may also apply to AFs.

Definition 60. *Let $G = (V, E), H = (V', E')$ denote directed graphs.*

- *We write $G \subseteq H$ if $V \subseteq V'$ and $E \subseteq E'$. If $G \subseteq H$ we say that G is a* subgraph *of H.*
- *We write $G \subset H$ if $G \subseteq H$ and either $V \subset V'$ or $E \subset E'$*
- *G is an* induced subgraph *of H if $G \subseteq H$ and $E = E' \cap V \times V$.*
- *For $S \subseteq V$, $G_{|S} = (S, E \cap S \times S)$ is the subgraph of G induced by S.*
- *$G \cup H = (V \cup V', E \cup E')$.*
- *We say G and H are* disjoint *if $V \cap V' = \emptyset$.*
- *If G and H are disjoint we may write $G \dot{\cup} H$ instead of $G \cup H$.*

An important concept in graph theory are paths between vertices and the concept of reachability.

Definition 61. *Given a directed graph $G = (V, E)$ and vertices $v_1, v_n \in V$ A* path *from v_1 to v_n is a finite sequence of vertices $\{v_i\}_{1 \leq i \leq n}$ such that $(v_i, v_{i+1}) \in E$ for $1 \leq i < n$ and $v_i \neq v_j$ for $1 \leq i < j \leq n$. We say that a vertex a is* reachable *from a vertex b if there exists a path from b to a. The* length *of a path $\{v_i\}_{1 \leq i \leq n}$ is $n - 1$.*

Definition 62. *Given a directed graph $G = (V, E)$ and vertices $v_1, v_n \in V$. A* cycle *is a finite sequence of vertices $\{v_i\}_{1 \leq i \leq n}$ such that $(v_i, v_{i+1}) \in E$ for $1 \leq i < n$ as well as $(v_n, v_1) \in E$ and $v_i \neq v_j$ for $1 \leq i \leq j < n$. The* length *of a cycle $\{v_i\}_{1 \leq i \leq n}$ is n.*

For directed graphs there are different concepts of (path) connectedness, one can choose whether to take the orientation of the edges into account or not.

Definition 63. *Given a directed graph $G = (V, E)$ and vertices $S \subseteq V$. We say that*

- *S is* weakly connected *if for each $a, b \in S$ there exists either a path from a to b and or from b to a.*
- *S is* strongly connected *if for each $a, b \in S$ there exists both a path from a to b and one from b to a.*
- *G is* weakly connected *if V is weakly connected.*
- *G is* strongly connected *if V is strongly connected.*
- *S is a* strongly connected component (SCC) *of G if S is $\subseteq$-maximal strongly connected.*

It is well known that each graph can be partitioned into a unique set of disjoint SCCs. Next we introduce some special graphs classes.

Definition 64. *A directed graph $G = (V, E)$ is a* tree *if G is weakly connected, G has no directed cycles and the underlying undirected graph (V, E'), with $E' = \{\{a, b\} \mid (a, b) \in E\}$, has no cycle.*[10]

In other words a tree is a weakly connected graph free of undirected cycles.

Definition 65. *A directed graph $G = (V, E)$ is a* (directed) acyclic graph (DAG) *if it has no directed cycles.*

A DAG is a graph which is free of directed cycles, but if we ignore the orientation of the edges it may contains cycles. Hence we have that each tree is a DAG, but not vice versa (see Figure 2.6).

Definition 66. *A directed graph $G = (V, E)$ is a* bipartite graph *if there are sets L, R such that $V = L \cup R$ and $E \subseteq (L \times R) \cup (R \times L)$. We may denote bipartite graphs also as (L, R, E) where $E \subseteq (L \times R) \cup (R \times L)$.*

Note that, in contrast to undirected graphs, there are directed acyclic graphs which are not bipartite, see Figure 2.8. However a (directed) tree is clearly bipartite.

Definition 67. *A directed graph $G = (V, E)$ is called* symmetric *if whenever $(a, b) \in E$ also $(b, a) \in E$.*

[10] That is G is the orientation of an undirected tree.

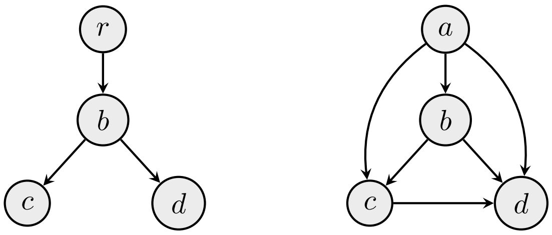


Figure 2.6: Illustration of two acyclic graphs: On the left hand side a tree; and on the right hand side a acyclic graph which is not a tree

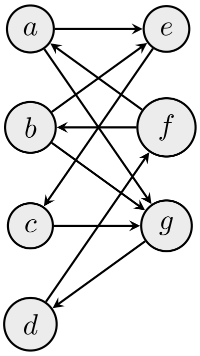


Figure 2.7: A bipartite graph, with vertex-partition $L = \{a, b, c, d\}$ and $R = \{e, f, g\}$.

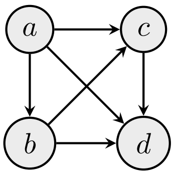


Figure 2.8: An acyclic graph, which is not bipartite

We mention that the above definition of symmetric graphs allows for loops on the graph, i.e. edges $(v, v) \in E$ for some $v \in V$.

We may also consider graphs where the vertices are labeled by numbers. In the following $[k]$ denotes the set of all positive integers less or equal to k, i.e. the set $\{1, \ldots, k\}$.

Definition 68. *A labeling of a graph $G = (V, E)$ is a function $\mathcal{L} : V \to [k]$. We call $(G, \mathcal{L})$ a k-graph.*

We consider an arbitrary graph (without labeling) as a k-graph with all vertices labeled by 1.

2.3.2 Structural Graph Parameters

In this section we introduce parameters that measure the structure of graphs, namely the parameters tree-width, (directed) clique-width and cycle-rank.

Tree-Width

Maybe the most popular parameter for graph problems is tree-width [19, 101] (see also [93]), which is original stated for undirected graphs but can be immediately applied to directed graphs. The intuition behind tree-width is that one want to measures how tree-like a graph is. Towards such a measure we define tree-decompositions of graphs.

Definition 69. *Let $G = (V, E)$ be a directed graph. A* tree-decomposition *of G is a pair $(\mathcal{T}, \mathcal{X})$ where $\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T})$ is a tree and $\mathcal{X} = (X_t)_{t \in V_\mathcal{T}}$ (we call X_t the* bag *of t) such that:*

1. *$\bigcup_{t \in V_\mathcal{T}} X_t = V$, i.e. $\mathcal{X}$ is a cover of V,*
2. *for each $v \in V$ the subgraph of $\mathcal{T}$ induced by $\{t \mid v \in X_t\}$ is connected,*
3. *for each edge $(v_i, v_j) \in E$ there exists an X_t with $\{v_i, v_j\} \subseteq X_t$.*

The width *of a decomposition $(\mathcal{T}, \mathcal{X})$ is given by* $\max\{|X_t| : t \in V_\mathcal{T}\} - 1$.

Each graph has a tree-decomposition, one can simple choose a tree consisting of only one vertex and put the whole graph into the corresponding bag. The pitfall of this would be the high width of the decomposition. To get nice (tree-like) computational properties we seek for tree-decompositions of low width. This brings us to the definition of the tree-width of a graph.

Definition 70. *The* tree-width *of a graph G is the minimum width over all tree decompositions of G.*

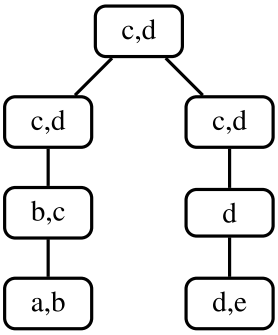


Figure 2.9: A tree-decomposition for the AF in example 1.

Clique-width

The parameter clique-width builds up on k-graphs and operations on it. So we start with defining so called initial k-graphs.

Definition 71. *A k-graph is called* initial k-graph*, if it contains exactly one vertex. We denote the initial k-graph consisting of a vertex v labeled by i as $i(v)$.*

Now k-graphs can be constructed from initial k-graphs by repeatedly using the following graph operations:

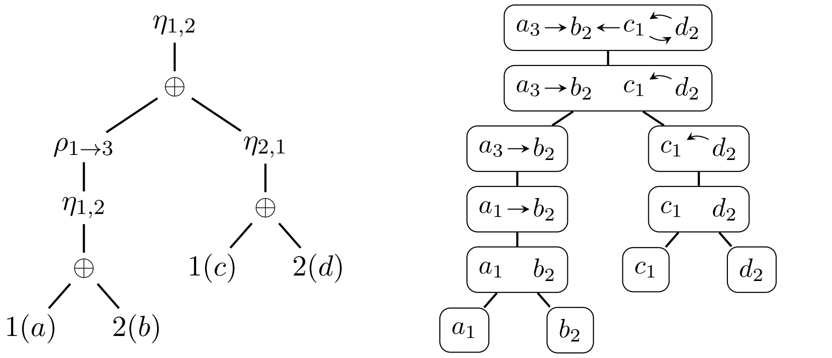
- *Disjoint union* (denoted by $\oplus$);
- *Relabeling*: changing all labels i to j (denoted by $\rho_{i \to j}$);
- *Edge insertion*: connecting all vertices labeled by i with all vertices labeled by j (denoted by $\eta_{i,j}$); already existing edges are not doubled.[11]

A construction of a k-graph G using the above operations can be represented by an algebraic term:

Definition 72. *A* cwd-expression *is a term composed of constants $i(v)$, the unary operations $\rho_{i \to j}$, $\eta_{i,j}$, and the binary operation $\oplus$ (i, j being integer, and v a vertex). A k-*expression *is a* cwd-*expression in which at most k different labels occur. We denote the set of all k-expressions by CW_k.*

We say that a *cwd-expression* defines a graph G when interpreting the algebraic operations with the above described graph operations results in the graph G. Hence, each k-expression corresponds to a unique k-graph, but a k-graph may correspond to several k-expressions as well as there exist k-graphs which have no k-expression at all. Nevertheless each k-graph has a cwd-expression that defines it, just thinking of giving each vertex a unique label and then applying an edge insertion for each edge of the graph.

[11] Some authors postulate that $i \neq j$ for the edge insertion $\eta_{i,j}$ to prohibit loops, but as AFs may have self-attacking arguments we do not.



(a) Parse tree of σ

(b) The corresponding graph construction

Figure 2.10: The parse tree and the corresponding graph construction process for the example cwd-expression $\sigma = \eta_{1,2}(\rho_{1\to3}(\eta_{1,2}(1(a) \oplus 2(b))) \oplus \eta_{2,1}(1(c) \oplus 2(d)))$. The subscript of a vertex denotes the current label of the vertex, e.g. b_2 denotes the vertex b with label 2.

Definition 73. *The* clique-width *of a graph G,* $\mathrm{cwd}(G)$*, is the smallest integer k such that G can be defined by a k-expression.*

Comparing tree-width and clique-width

The graph parameters tree-width and clique-width are closely related. Thus we discuss properties of these parameters simultaneously, in particular we concentrate on properties which are relevant for obtaining fixed-parameter tractability results.

We have that clique-width generalises tree-width in the sense that each graph class having bounded tree-width also having bounded clique-width, but there are graph classes of bounded clique-width, for instance complete graphs, which have arbitrary high tree-width [30, 35]. In particular only sparse graphs have low tree-width while there are both sparse and dense graphs with low clique-width.

Both computing the tree-width and clique-width of a graph is in general NP-hard [2, 68]. But as we are interested in FPT results we are more curious about the case where the width is bounded by a fixed k. In that case for tree-width we can check whether the graph has tree-width $\leq k$ in polynomial time and in the positive case we also get a tree-decomposition of width k [18]. For clique-width the best known algorithms work within an additive approximation error in terms of k. That is there is a function f such that one can find an $f(k)$-expression for an graph of clique-width k [82, 96]. Hence, given a graph of low tree-width, using tree-width would be of advantage as we probably get lower values for the tree-width than for the actual k-expression. For practical purposes we also can benefit from the fact that there are several well studied heuristics for computing tree-decompositions [40], while we are not aware of such heuristics for clique-width. On the other hand there are graphs of inherent high tree-width where clique-width would be the better choice.

Generalizations of tree-with

One pitfall of the parameter tree-width is that is original stated for undirected graphs and thus does not take the orientation of edges into account. There are several promising approaches around, that in some sense generalize tree-with to directed graphs: directed tree-width [81], directed path-width [5], DAG-width [16] and Kelly-width [78]. As we do not explicitly require the definitions of these graph parameters in this work, we omit the formal definitions here, and refer the interested reader to the above mentioned papers.

Instead we introduce the closely related parameter cycle-rank [63], which we will use later on also to obtain results for the above mentioned parameters.

Definition 74. *Let $G = (V, E)$ be a directed graph. The* cycle rank *of G, $cr(G)$, is defined as follows: An acyclic graph has $cr(G) = 0$. If G is strongly connected then $cr(G) = 1 + min_{v \in V_G} cr(G \setminus v)$. Otherwise, $cr(G)$ is the maximum cycle rank of $F_{|S}$ among all strongly connected components S of G.*

Example 3. *Consider our running example for AFs, $F = (A, R)$, with $A = \{a, b, c, d, e\}$ and $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$. We have the following SCCs $\{a\}$, $\{b\}$, $\{c, d\}$ and $\{e\}$. The induced subgraphs of the first two components are acyclic while the others are not. But when removing one argument also the remaining SCCs become acyclic and thus $cr(F) = 1$.*

2.3.3 Monadic Second Order Logic (MSO)

Towards a nice tool for deciding whether a problem is fixed-parameter tractable w.r.t. treewidth or clique-width we briefly recall the Monadic Second Order Logic (MSO). Informally MSO is an extension of first order logic that allows for quantification over sets.

Next we give a formal definition of Monadic Second Order Logic basically following [93, Section 10.6], the minor differences occur as we distinguish MSO (or MSO_1) and MSO_2. First of all we have an infinite amount of *individual variables* $x, y, z, \dots$ and *set variables* $X, Y, Z, \dots$ and some relation symbols. As we are only interested in graphs, respective argumentation frameworks, we only consider the relation symbols V and E (resp. A and R) where the first is a unary relation interpreted as the vertices and the second is a binary relation interpreted as the edges.

As usual MSO-formulas are defined in a recursive way starting from atomic formulas. The *atomic MSO-formulas* are of the form $x = y$, $V(x)$, $E(x, y)$ and $x \in X$ for individual variables x, y and set variable X. General MSO-formulas are then built by applying the following rules:

- if $\varphi \in MSO$ then $\neg\varphi \in MSO$;
- if $\varphi, \psi \in MSO$ then $\varphi \vee \psi \in MSO$, $\varphi \wedge \psi \in MSO$ and $\varphi \rightarrow \psi \in MSO$;
- if $\varphi \in MSO$ then $\exists x\varphi \in MSO$, $\forall x\varphi \in MSO$, $\exists X\varphi \in MSO$ and $\forall X\varphi \in MSO$.

Next let us consider the semantics of MSO-formulas. Given a graph $G = (V, E)$, an assignment α for an MSO-formula is a function mapping each individual variable to an element of V and any set variable to a subset of V. We inductively define when an assignment α satisfies a MSO-formula φ, and denote this as $(G, \alpha) \models \varphi$.

- $(G, \alpha) \models x = y$ iff $\alpha(x) = \alpha(y)$;
- $(G, \alpha) \models V(x)$ iff $\alpha(x) \in V$;
- $(G, \alpha) \models E(x, y)$ iff $(\alpha(x), \alpha(y)) \in E$;
- $(G, \alpha) \models x \in X$ iff $\alpha(x) \in \alpha(X)$;
- $(G, \alpha) \models \neg\varphi$ iff $(G, \alpha) \not\models \varphi$;
- $(G, \alpha) \models \varphi \vee \psi$ iff $(G, \alpha) \models \varphi$ or $(G, \alpha) \models \psi$;
- $(G, \alpha) \models \varphi \wedge \psi$ iff $(G, \alpha) \models \varphi$ and $(G, \alpha) \models \psi$;
- $(G, \alpha) \models \varphi \rightarrow \psi$ iff $(G, \alpha) \models \varphi$ implies $(G, \alpha) \models \psi$;
- $(G, \alpha) \models \exists x\varphi$ iff there exists an $v \in V$ such that $(G, \alpha_{x \to v}) \models \varphi$;[12]
- $(G, \alpha) \models \forall x\varphi$ iff for each $v \in V$ it holds that $(G, \alpha_{x \to v}) \models \varphi$;
- $(G, \alpha) \models \exists X\varphi$ iff there exists an $A \subseteq V$ such that $(G, \alpha_{X \to A}) \models \varphi$;[13]
- $(G, \alpha) \models \forall X\varphi$ iff for each $A \subseteq V$ it holds that $(G, \alpha_{X \to A}) \models \varphi$.

It is easy to see that only the values of α for free variables are relevant for satisfying a MSO-formula. We usually write $\varphi(x_1, \ldots, x_i, X_1, \ldots X_j)$ to denote that the free variables of φ are $x_1, \ldots, x_i, X_1, \ldots X_j$. For $v_k \in V$ and $A_k \subseteq V$ we define that $G \models \varphi(v_1, \ldots, v_i, A_1, \ldots A_j)$ iff for each assignment α with $\alpha(x_k) = v_k, 1 \leq k \leq i$ and $\alpha(X_k) = A_k, 1 \leq k \leq j$ holds that $(G, \alpha) \models \varphi$.

A natural way to extend MSO is to allow quantification over binary relations, i.e. quantification over subsets of the edges of the graph, which leads us to the definition of MSO$_2$. To this end we introduce *edge set variables* $X^E, Y^E, Z^E, \ldots$ and extend the syntax and semantics of MSO in the straightforward way (we omit details here). The avoid misunderstandings in the following we will use MSO$_1$ to denote MSO without edge set variables and MSO$_2$ to denote MSO enriched with edge set variables.

2.3.4 Meta-Theorems

In this section we present meta-theorems which allow for a easy classification of problems which are fixed-parameter tractable w.r.t. tree-width or clique-width. Further we present results that allow to propagate hardness results for graphs of bounded cycle-rank to graph classes where one of the parameters directed tree-width, directed path-width, DAG-width or Kelly-width is bounded.

The first meta-theorem by Courcelle [32, 33] basically says that each graph property that can be stated in MSO$_2$ can be easily decided on graphs of bounded tree-width. In this work we us the formulation from the book of Niedermeier [93].

[12] $\alpha_{x \to v}$ defined as: $\alpha_{x \to v}(x) = v$ and $\alpha_{x \to v}(y) = \alpha(y)$ for all $y \neq x$ as well as $\alpha_{x \to v}(Y) = \alpha(Y)$.

[13] $\alpha_{X \to A}$ defined as: $\alpha_{X \to A}(X) = A$ and $\alpha_{X \to A}(Y) = \alpha(Y)$ for all $Y \neq X$ as well as $\alpha_{X \to A}(y) = \alpha(y)$.

Theorem 8 (Courcelle's Theorem)**.** *[32, 33] Given an MSO_2 formula $\varphi(x_1, \dots, x_i, X_1, \dots X_j, X_1^E, \dots X_l^E)$ and an integer k. There is a linear time algorithm, given a graph $G = (V, E)$, $v_k \in V$, $A_k \subseteq V, B_k \subseteq E$ and a tree-decomposition for G of width at most k deciding whether $G \models \varphi(v_1, \dots, v_i, A_1, \dots A_j, B_1, \dots B_l)$.*

A similar results for clique-width and MSO_1 is by Courcelle, Makowsky, and Rotics [36].

Theorem 9. *[36] Given an MSO_1 formula $\varphi(x_1, \dots, x_i, X_1, \dots X_j)$ and an integer k. There is a linear time algorithm, given a graph $G = (V, E)$, $v_k \in V$, $A_k \subseteq V$ and a k-expression for G deciding whether $G \models \varphi(v_1, \dots, v_i, A_1, \dots A_j)$.*

Notice that both theorems require that a certain representation of the AF is given, i.e. a tree-decomposition of with at most k or a k-expression, and thus those theorems leave one issue open for obtaining fixed-parameter tractability. Fortunately, due to results in [20, 96], we have that $f(k)$-expressions[14] and tree-decompositions of with k can be computed in linear time if k is bounded by a constant.

Now comparing tree-width and clique-width in the context of fixed-parameter tractability in one sentence: We have that tree-width applies to a broader range of problems but to a strictly smaller class of graphs than clique-width.

These meta-theorems are stated in terms of decision problems, which is perfectly fine for our purposes, but they where also extended to work for counting problems [3, 37] and enumerating solutions [70] (see also [98]).

The above meta-theorems provide positive results for tree-width and clique-width, we are now interested in meta-results for negative results. That is one can propagate hardness for graphs of bounded cycle rank to graphs of bounded directed tree-width, directed path-width, DAG-width or Kelly-width.

Theorem 10. *[16, 77, 78] If a problem is $\mathcal{C}$-hard for graphs of bounded cycle-rank then it is also $\mathcal{C}$-hard for*

- *graphs of bounded directed tree-width;*
- *graphs of bounded directed path-width;*
- *graphs of bounded DAG-width; and*
- *graphs of bounded Kelly-width.*

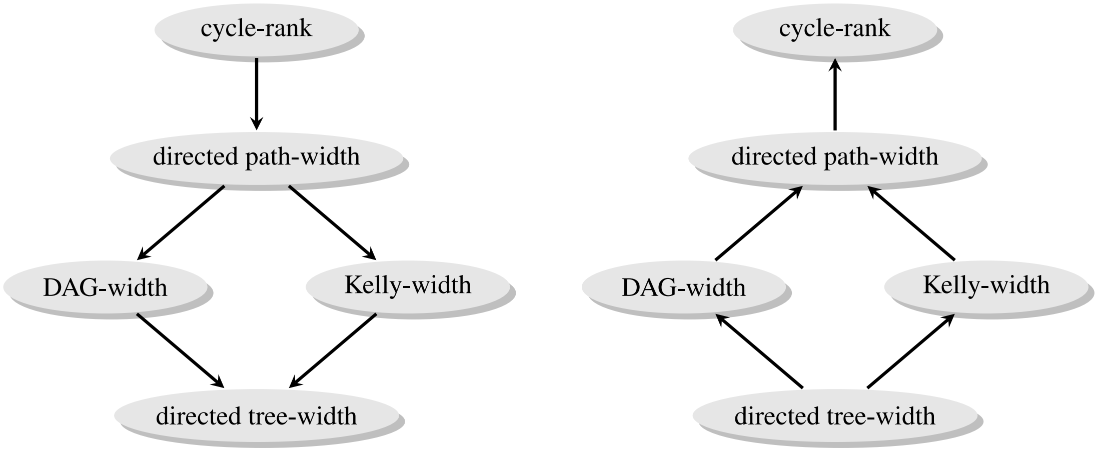
Moreover the relations depicted in Figure 2.11 hold.

Interpreting Theorem 10 one can see that it might be a good idea to state FPT-results for directed tree-width and state hardness results for cycle-rank[15].

Now given these meta-theorems as tools we are well prepared for studying fixed-parameter tractability w.r.t. graph parameters for the reasoning problems in abstract argumentation (see Section 4.2).

[14] The $f(k)$ denoting that there is a approximation error in the number of colors used in the cwd-expression provided by the algorithm.

[15] The thoughtful reader may already have a clue about which kind of results we are going to present.



(a) Propagating Hardness

(b) Propagating FPT

Figure 2.11: Relation between directed graph measures (An arrow means that if a problem is hard (resp. FPT) w.r.t. first parameter then is also hard (resp. FPT) w.r.t. second parameter)

CHAPTER 3

Complexity Analysis

This chapter is dedicated to a careful complexity analysis of reasoning tasks in abstract argumentation frameworks w.r.t. different semantics. Our motivation for doing complexity analysis in this setting is manifold.

Firstly, following what one might call the *computational point of view*, we want to estimate the (worst case) computational costs of solving problem instances. So given a complexity classification of a problem we are aware of the best worst case behaviour an algorithm can have, which might prevent us wasting time with trying to improve algorithms in that direction. Obviously from this point of view we gain for low complexity to get fast worst case algorithms.

Secondly, from the *Knowledge Representation point of view*, we are interested in the complexity of reasoning problems to explore the expressiveness of the corresponding formalisms. For instance if an formalism has too low complexity there would be a broad range of interesting problems having inherent higher complexity that can not be expressed within the formalism. Hence from this perspective a sufficient high complexity is an essential ingredient for a useful formalism, in our case for a reasonable argumentation semantics.

Finally, we consider what we call the *Practitioners point of view*. An successful approach for implementing reasoning tasks for a new formalisms is what we will call the *reduction approach*. That is instead of designing and implementing complex algorithms from scratch, one reduces the new reasoning tasks to related formalisms where sophisticated solvers already exist. Here the complexity of the actual problem and target formalism are crucial as follows: Given that an actual problem has higher complexity than the designated target formalism we know that there is no efficient encoding of our problem and we might should consider a different target formalism (or using worst case exponential time translations). On the other hand if the target formalism is of higher complexity we may end up with unnecessarily high computational costs. Then it might be a good idea to encode the problem within a restriction of the target formalism, providing lower complexity.

There are several approaches for implementations of (abstract) argumentation reasoners following the reduction approach. Most prominent there are the approaches to reduce the computation of extensions or specific argumentation reasoning task to: SAT-solving [17] and resp.

to a Quantified Boolean Formula (QBF) [64]; Answerset-Programming (ASP) [65]; or a Constraint Satisfaction Problem (CSP) [1]. Moreover, in Chapter 5 we will exploit the possibilities to reduce different argumentation semantics to each other, and indeed make heavily use of the complexity results presented in this chapter to obtain some negative results.

This chapter is organised as follows:

- In Section 3.1 we summarise and discuss existing complexity results for the problems under our consideration.
- In Section 3.2 we study *P-completeness* for argumentation reasoning problems. Firstly, we show several problems concerning grounded semantics to be P-hard and thus complete for P. Secondly, we extend these results to resolution-based grounded semantics and show that verifying a resolution-based grounded extension is hard for P. Finally, we show that several problems which are known to be solvable in polynomial time, can be actually solved in L and are thus (under typical complexity-theoretic) assumptions not P-complete.
- In Section 3.3 we study the *complexity of semi-stable and stage semantics*. That is we first complete the complexity analysis of semi-stable semantics started by Dunne and Caminada [50], presenting matching lower bounds for credulous and skeptical reasoning. Moreover we give the exact complexity classification of credulous and skeptical reasoning w.r.t. stage semantics, as well as for verifying a stage extension.
- In Section 3.4 we consider the *complexity of ideal reasoning*. That is we give generic complexity results, i.e. upper and lower complexity bounds for ideal reasoning using the complexity of the base-semantics as parameter. Moreover we give exact complexity characterisations for all base-semantics under our considerations.
- In Sections 3.5 we summarise the results obtained in this chapter and together with the results from the literature we draw the complexity landscape of abstract argumentation. Finally we discuss two problems that we have to leave open.

Parts of this chapter have been previously published: Section 3.2 builds on a P-hardness result presented in [55, 56]; Section 3.3 presents results published in [54–56]; and the results underlying Section 3.4 have been published in [59].

3.1 State-of-the Art

Here we summarise existing complexity results for abstract argumentation. For a good starting point into the computational complexity of abstract argumentation the interested reader is referred to [51]. Table 3.3 summarises the complexity results obtained from the literature. For Dungs semantics the “in P” and “trivial“ results are well-known and follow immediately by properties of the corresponding semantics already shown in Dung [42]. The case of naive semantics has been explicitly studied in [31].

The complexity results for stable, admissible and preferred semantics follow from results by Dimopoulos and Torres [41] on logic programs, except the Π_2^P completeness of $Skept_{prf}$

which is due to Dunne and Bench-Capon [48] and the Θ_2^P-membership of $Ideal_{prf}$ [46]. The complexity of complete semantics has been studied by Coste-Marquis et al. [31]. The results for semi-stable semantics are due to Dunne and Caminada [50], where they additionally to the results listed in Table 3.3 show that both $Cred_{sem}, Skept_{sem}$ are Θ_2^P-hard. Finally the complexity of resolution based grounded semantics has been studied by Baroni et al. [12].

Concerning the problem $Ideal_{prf}$, Dunne [46] has shown coNP-hardness, that a NP-hardness proof would suffice for showing Θ_2^P-completeness, and moreover that $Ideal_{prf}$ is Θ_2^P-complete under so called randomized reductions.

σ	$Cred_\sigma$	$Skept_\sigma$	$Ideal_\sigma$	Ver_σ	$Exists_\sigma$	$Exists_\sigma^{\neg\emptyset}$
cf	in P	trivial	?	in P	trivial	in P
$naive$	in P	in P	?	in P	trivial	in P
grd	in P	in P	?	in P	trivial	in P
stb	NP-c	coNP-c	?	in P	NP-c	NP-c
adm	NP-c	trivial	?	in P	trivial	NP-c
com	NP-c	in P	?	in P	trivial	NP-c
$resGr$	NP-c	coNP-c	?	in P	trivial	in P
prf	NP-c	Π_2^P-c	in Θ_2^P	coNP-c	trivial	NP-c
sem	in Σ_2^P	in Π_2^P	?	coNP-c	trivial	NP-c
stg	?	?	?	?	?	?

Table 3.1: State-of-the art complexity landscape for abstract argumentation ($\mathcal{C}$-c denotes completeness for class $\mathcal{C}$).

If an AF has no stable extensions, according to our definition of skeptical acceptance, all arguments are skeptically accepted. This may be unwanted and hence one might consider a variation of the skeptical acceptance problem, let us call it $Skept'_{stb}$, asking whether an argument is contained in all extensions and there exists at least one extension. Due to Dunne and Wooldridge the problem $Skept'_{stb}$ is complete for the class D^P [51].

3.2 Tractable Problems

In this section we study the exact complexity of the argumentation problems within the class P. That is we either show that a problems is P-complete under log-reductions or we show that it can be solved within the class L.

We start with proving a lower bound for grounded semantics which, to the best of our knowledge, have not been established yet.

Proposition 6. *The problems $Cred_{grd} = Skept_{grd} = Skept_{com}$ as well as Ver_{grd} are P-hard (under L-reductions, i.e. reductions using logarithmic space).*

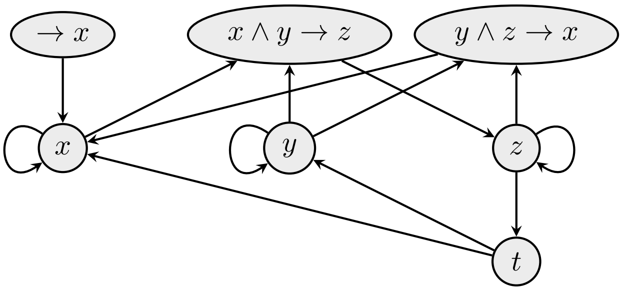


Figure 3.1: Argumentation framework $F_{\varphi,z}$ for $\varphi = \{\to x, x \wedge y \to z, y \wedge z \to x\}$.

Proof. We use a reduction from the P-hard problem HORNSAT. To this end let $\varphi = \{r_l : b_{l,1} \wedge \cdots \wedge b_{l,i_l} \to h_l \mid 1 \leq l \leq n\}$ be a definite Horn theory over atoms X. We construct the AF $F_{\varphi,z} = (A, R)$ as follows:

$$\begin{aligned} A &= \varphi \cup X \cup \{t\} \\ R &= \{(x, x), (t, x) \mid x \in X \setminus \{z\}\} \cup \{(z, t)\} \cup \\ &\quad \{(r_l, h_l), (b_{l,j}, r_l) \mid r_l \in \varphi, 1 \leq j \leq i_l)\} \end{aligned}$$

where t is a fresh argument. See Figure 3.1 for an example. Clearly the AF $F_{\varphi,z}$ can be constructed using only logarithmic space in the size of φ.

In the following we show that z is in the minimal model of φ iff t is in the grounded extension of $F_{\varphi,z}$ iff $grd(F_{\varphi,z}) = \{\varphi \cup \{t\}\}$.

First we attend that t is in the grounded extension E of $F_{\varphi,z}$ iff $E = \{\varphi \cup \{t\}\}$. Obviously the if-direction holds. Thus let us assume $t \in E$, then each $x \in X \setminus \{z\}$ is attacked by t and as z attacks t we have that E also attacks z. Thus each $r \in \varphi$ is defended by E. Hence $E = \{\varphi \cup \{t\}\}$.

It remains to show that z is in the minimal model of φ iff t is in the grounded extension E of $F_{\varphi,z}$. We recall the definition of the characteristic function $\mathcal{F}_F$ of an AF F, defined as $\mathcal{F}_F(S) = \{x \in A_F \mid x \text{ is defended by } S\}$, and that the grounded extension of F is the least fix-point of $\mathcal{F}_F$. To show the only-if part, let us assume that z is in the minimal model of φ. Thus there exists a finite sequence of rules $(r_{l_i})_{1 \leq i \leq k}$, such that (i) for each rule r_{l_i} and each atom $b_{l_i,s}$ there exists a rule $r_{l_j}, j < i$ with $h_{l_j} = b_{l_i,s}$ and (ii) $h_{l_k} = z$. Clearly r_{l_1} has empty body and thus the corresponding argument has no attackers in $F_{\varphi,z}$, i.e. $r_{l_1} \in E$. We now claim that for each i, $1 \leq i \leq k$, $r_{l_i} \in E$ holds as well and prove this by induction. To this end, we assume the claim holds for all $m < i$, i.e. $r_{l_m} \in E$, and thus $E \rightarrowtail h_{l_m}$ for $m < i$ holds. Using (i) we get that for each argument $a \in A$ with $a \rightarrowtail r_{l_i}$, it holds that $E \rightarrowtail a$. Hence $r_{l_i} \in E$. Now in particular $r_{l_k} \in E$ and by (ii) we have that $E \rightarrowtail z$. As z is the only argument attacking t we also have that $t \in E$.

To show the if-part, let us assume that t is contained in the grounded extensions E of $F_{\varphi,z}$. Then by construction $E \rightarrowtail z$ and thus there exists an integer k, such that $\mathcal{F}_F^k(\emptyset) \rightarrowtail z$ and for each $m < k : \mathcal{F}_F^m(\emptyset) \not\rightarrowtail z$. We claim that for $1 \leq m \leq k$ and $x \in X$ it holds that if $\mathcal{F}_F^m(\emptyset) \rightarrowtail x$ then x is in the minimal model of φ. The proof is by induction on m. As induction base consider $\mathcal{F}_F(\emptyset)$. By construction $\mathcal{F}_F(\emptyset)$ is the set of arguments that correspond to rules in φ having empty body. The arguments attacked by $\mathcal{F}_F(\emptyset)$ are the head atoms of these rules, which are clearly in the minimal model. For the induction step assume that $\mathcal{F}_F^{m-1}(\emptyset)$ only attacks arguments corresponding to atoms in the minimal model. As $\mathcal{F}_F^{m-1}(\emptyset) \not\rightarrowtail z$ we have $t \notin \mathcal{F}_F^{m-1}(\emptyset)$. Let $x \in X$ be an argument such that $\mathcal{F}_F^m(\emptyset) \rightarrowtail x$, but $\mathcal{F}_F^{m-1}(\emptyset) \not\rightarrowtail z$. Then there exists an $r_i \in \varphi$ such that $h_i = x$ and $r_i \in \mathcal{F}_F^m(\emptyset)$. By construction of $F_{\varphi,z}$ we have that the argument r_i is defended by $\mathcal{F}_F^{m-1}(\emptyset)$ iff each atom in the body of r_i is attacked by $\mathcal{F}_F^{m-1}(\emptyset)$. Hence, by assumption each atom in the body of r_i is contained in the minimal model of φ. But then the head h_i of r_i is in the minimal model of φ. Hence, as $\mathcal{F}_F^k(\emptyset) \rightarrowtail z$, we get that z is in the minimal model of φ. □

The above hardness result can be extended to resolution-based grounded semantics, by "cleaning" the Horn theory from redundant rules.

Proposition 7. *The problem Ver_{resGr} is P-complete.*

Proof. The membership part was shown in [12]. Recall the reduction from the proof of Proposition 6, which shows P-hardness for Ver_{grd}. Now we restrict ourselves to definite Horn theories such that head and body are disjoint. Clearly such rules do not effect the models and we can identify such rules in logarithmic space. For such Horn theories we have that Ver_{grd} is P-hard on AFs without symmetric attacks. On such frameworks grounded and resolution-based grounded semantics coincide and therefore Ver_{resGr} is P-hard. □

Finally we show that several problems can be decided within L, and are thus most likely not P-complete.

Theorem 11. *The following Problems can be decided within L:*

1. $Cred_{cf}$, $Cred_{naive}$, $Skept_{naive}$
2. Ver_{cf}, Ver_{naive}
3. Ver_{stb}, Ver_{adm}, Ver_{com},
4. ${Exists^{\neg\emptyset}}_{cf}$, ${Exists^{\neg\emptyset}}_{naive}$, ${Exists^{\neg\emptyset}}_{stg}$
5. ${Exists^{\neg\emptyset}}_{grd}$, ${Exists^{\neg\emptyset}}_{com}$

and are thus not P-complete (unless L = P).

Proof. In the following proofs we denote the AF we are interested in as $F = (A, R)$.

$Cred_{cf} = Cred_{naive}$: An argument a that is self-attacking can not be in a conflict-free set. But if $a \in A$ is not self-attacking then the set $\{a\}$ is a conflict-free set containing a. Thus we can decide $Cred_{cf} = Cred_{naive}$ by testing whether a is self-attacking which is certainly in L.

$Skept_{naive}$: We have that an argument $a \in A$ is skeptically accepted w.r.t. naive semantics iff it is credulously accepted and none of its neighbours $b \in \{a\}^{\ominus} \cup \{a\}^{\oplus}$ is credulously accepted.

So we can use a cursor to iterate over the neighbourhood and check whether these arguments are credulously accepted, which we have already shown to be in L.

Ver_{cf}: To check that a set $S \subset A$ is conflict-free one can use two cursors iterating over all pairs (a, b) for $a, b \in S$ and testing whether they are in conflict.

Ver_{naive}: To check that a conflict-free set $S \subset A$ is $\subseteq$-maximal one can use a cursor iterating over all arguments in $A \setminus S$ and test whether they are in conflict with at least one argument in S (using another cursor iterating over S).

Ver_{stb}: To check that a conflict-free set $S \subset A$ is stable, one can use a cursor iterating over all arguments in $A \setminus S$ and test whether they are attacked by at least one argument in S (using another cursor iterating over S).

Ver_{adm}: As mentioned above we can check conflict-freeness in L. It remains to check that each argument in S is defended. To this end we use a cursor that iterates over all arguments $a \in S$ and then tests each of these arguments as follows. We use a cursor to iterate over arguments $b \in \{a\}^{\ominus}$ (the attackers of a) and test whether they are attacked by S (using another cursor).

Ver_{com}: To check that an admissible set is also a complete extension, one iterates over all arguments $a \in A \setminus S$ proving that there is at least one unattacked attacker of a. That is one iterates over all attackers $b \in \{a\}^{\ominus}$ and tests whether b is attacked by S.

${Exists^{\neg\emptyset}}_{cf} = {Exists^{\neg\emptyset}}_{naive} = {Exists^{\neg\emptyset}}_{stg}$: We have that there exists a nonempty conflict-free set iff there exists an argument which is not self-attacking.

${Exists^{\neg\emptyset}}_{grd} = {Exists^{\neg\emptyset}}_{com}$: The grounded extension is non-empty iff there exists an argument which is not attacked. □

We have classified the complexity of almost all of the problems in the class P. We only have to leave the exact complexity (w.r.t. log-reductions) of the problem ${Exists^{\neg\emptyset}}_{resGr}$ open.

3.3 Complexity of Semi-Stable and Stage Semantics

In this section we complement existing complexity results for credulous and skeptical acceptance. That is we give exact complexity characterisations for semi-stable and stage semantics. The complexity of semi-stable semantics has been studied in [50]. There the authors show the problems $Cred_{sem}, Skept_{sem}$ to be Θ_2^P-hard and that $Cred_{sem} \in \Sigma_2^P$ resp. $Skept_{sem} \in \Pi_2^P$, but they leave the exact complexity as an open problem. For the stage semantic the author is not aware of any work addressing computational complexity [1].

Towards our results for credulous and skeptical reasoning we first address the problems of verifying a semi-stable resp. stage extension. The former was studied in [50] and shown to be coNP-complete. We provide the corresponding result for stage semantics.

In what follows we consider a countable set U of propositional atoms (we will use atoms and arguments interchangeably). Moreover, we have the following pairwise disjoint sets of arguments $\bar{U} = \{\bar{u} \mid u \in U\}$, $U' = \{u' \mid u \in U\}$, $\bar{U}' = \{\bar{u}' \mid u \in U\}$. For any set $V \subseteq U$,

[1] While [25] introduces an algorithm for stage semantics there is no complexity analysis, neither of the reasoning problems nor of the algorithm.

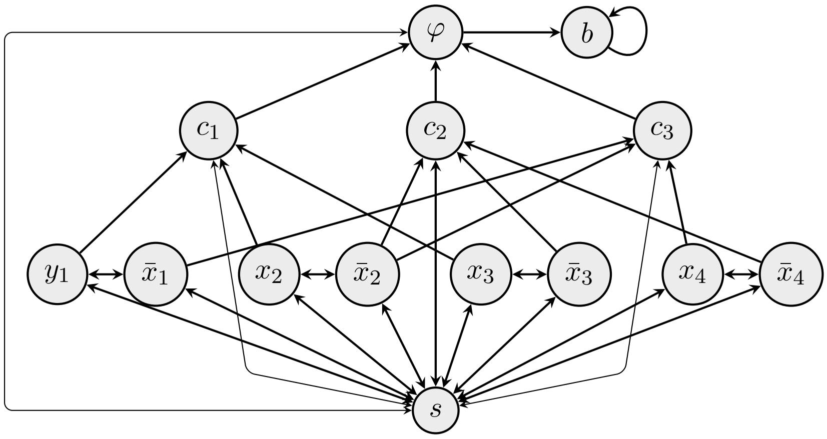


Figure 3.2: AF $F_{\{c_1,c_2,c_3\}}$ with $c_1 = \{x_1, x_2, x_3\}$, $c_2 = \{\bar{x}_2, \bar{x}_3, \bar{x}_4\}$, $c_3 = \{\bar{x}_1, x_2, x_4\}$.

we use $\bar{V}$, V', $\bar{V}'$, also as renaming schemes in the usual way (for instance, V' denotes the set $\{v' \mid v \in V\}$). Finally, we use further new arguments $\varphi, \bar{\varphi}, b, s, \ldots$ and $\{c_1, c_2, \ldots\}$.

Proposition 8. *Ver_{stg} is coNP-complete.*

Proof. Let us first consider the membership part: By definition, S is a stage extension of F iff (i) $S \in cf(F)$ and (ii) $\forall\, T \subseteq A, T \in cf(F)$ only if $S_R^+ \not\subset T_R^+$. Given S, we can decide $S \in cf(F)$ in polynomial time. For the complement of (ii), we guess a set T and then we verify (again, in polynomial time), whether $S_R^+ \subset T_R^+$ and $T \in cf(F)$. This yields membership in NP for the complement of (ii), thus, given set S, (ii) is in coNP, and thus the entire problem is in coNP.

We show coNP-hardness by reducing the NP-hard problem 3-SAT to the complementary problem of Ver_{stg}. We assume that a 3-CNF formula is given as a set C of clauses, where each clause is a set over atoms and negated atoms (denoted by $\bar{x}$). For such a CNF φ over variables X, define the AF $F_\varphi = (A, R)$ with

$$\begin{aligned} A &= X \cup \bar{X} \cup C \cup \{s, \varphi, b\} \\ R &= \{(x, \bar{x}), (\bar{x}, x) \mid x \in X\} \cup \{(l, c) \mid l \in c, c \in C\} \cup \\ &\quad\; \{(c, \varphi) \mid c \in C\} \cup \{(s, y), (y, s) \mid y \in A \setminus \{s, b\}\} \cup \{(\varphi, b), (b, b)\} \end{aligned}$$

where $\bar{X} = \{\bar{x} \mid x \in X\}$ and s, φ, b are fresh arguments. See Figure 3.2 for an illustrating example. We show that φ is satisfiable iff $\{s\}$ is not a stage extension of F_φ. First let us assume φ is satisfiable and let M be a model of φ. Then the set $E = \{\varphi\} \cup M \cup \overline{X \setminus M}$ is a stable extension of F_φ, i.e. $E_R^+ = A$, and since $\{s\}_R^+ = A \setminus \{b\}$, $\{s\}$ is not a stage extension of F_φ. Now let us assume that $\{s\}$ is a stage extension. By the same argumentation as above, i.e. using $\{s\}_R^+ \subset A$, we get that F_φ has no stable extension. But as we have seen before each model of φ corresponds to a stable extension of F_φ. Thus we can conclude that φ is unsatisfiable. □

Towards our main results in this section we introduce the following reduction from $QBF_{\forall}^2$ formulas to AFs.

Reduction 1. *Given a $QBF_{\forall}^2$ formula $\Phi = \forall Y \exists Z C$, we define $F_\Phi = (A, R)$, where*

$$\begin{aligned} A &= \{\varphi, \bar{\varphi}, b\} \cup C \cup Y \cup \bar{Y} \cup Y' \cup \bar{Y}' \cup Z \cup \bar{Z} \\ R &= \{(c, \varphi) \mid c \in C\} \cup \{(\varphi, \bar{\varphi}), (\bar{\varphi}, \varphi), (\varphi, b), (b, b)\} \cup \\ & \quad \{(x, \bar{x}), (\bar{x}, x) \mid x \in Y \cup Z\} \cup \\ & \quad \{(y, y'), (\bar{y}, \bar{y}'), (y', y'), (\bar{y}', \bar{y}') \mid y \in Y\} \cup \\ & \quad \{(l, c) \mid l \in C, c \in C\}. \end{aligned}$$

Figure 3.3 illustrates the corresponding AF F_Φ for Φ from Example 2. We first present several technical lemmata concerning Reduction 1.

Lemma 5. *For every stage (resp. semi-stable) extension S of an AF $F_\Phi = (A, R)$, the following propositions hold: (i) $b \notin S$, as well as $y' \notin S$ and $\bar{y}' \notin S$ for each $y \in Y$ and (ii) $x \notin S \Leftrightarrow \bar{x} \in S$ for each $x \in \{\varphi\} \cup Y \cup Z$.*

Proof. Let $\Phi = \forall Y \exists Z C$ and $F_\Phi = (A, R)$ be the corresponding AF.

ad (i) Clear, since all these arguments are self-attacking.

ad (ii) Obviously, for each $x \in \{\varphi\} \cup Y \cup Z$, $\{x, \bar{x}\} \subseteq S$ cannot hold, since S has to be conflict-free in F_Φ. It remains to show $\{x, \bar{x}\} \cap S \neq \emptyset$. Towards a contradiction, let us assume there exists such an x, such that $\{x, \bar{x}\} \cap S = \emptyset$ holds for a stage (resp. semi-stable) extension S of F_Φ.

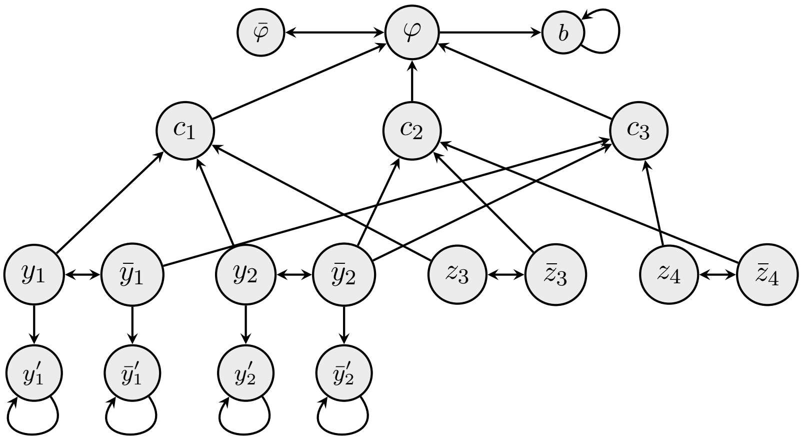


Figure 3.3: Illustration of the AF F_Φ, for the valid $QBF_{\forall}^2$ Φ, taken from Example 2, that is $\Phi = \forall y_1 y_2 \exists z_3 z_4 \{\{y_1, y_2, z_3\}, \{\bar{y}_2, \bar{z}_3, \bar{z}_4)\}, \{\bar{y}_1, \bar{y}_2, z_4\}\}$.

Let us first assume $x = \varphi$. Then the set $T = S \cup \{\bar{\varphi}\}$ is conflict-free and we have $S <_R^+ T$. The argument $\bar{\varphi}$ defends itself and therefore T is admissible if S is. This already shows that S then cannot be a stage or semi-stable extension.

Let us thus assume that $x \in Y \cup Z$ and let $T = (S \setminus \{c \in C \mid (\bar{x}, c) \in R\}) \cup \{\bar{x}\}$. One can check that T is conflict-free and that if S is admissible then T is admissible. Moreover, we again have $S <_R^+ T$. In fact, for the removed arguments $c \in C$, we have $c \in T_R^+$. Moreover, the only argument attacked by such c is φ, but $\varphi \in T_R^+$, since we can already assume $\{\varphi, \bar{\varphi}\} \cap S \neq \emptyset$. This shows that S cannot be a stage (resp. semi-stable) extension. □

Lemma 6. *Let $Y^* = Y \cup \bar{Y} \cup Y' \cup \bar{Y}'$ and S, T be conflict-free sets in $F_\Phi = (A, R)$. Then $S \cap Y^* \subseteq T \cap Y^*$ iff $(S \cap Y^*)_R^+ \subseteq (T \cap Y^*)_R^+$ and further $S \cap Y^* = T \cap Y^*$ iff $(S \cap Y^*)_R^+ = (T \cap Y^*)_R^+$.*

Proof. First, assume $S \cap Y^* \subseteq T \cap Y^*$. By the monotonicity of $(.)_R^+$ we get $(S \cap Y^*)_R^+ \subseteq (T \cap Y^*)_R^+$. So, assume now $(S \cap Y^*)_R^+ \subseteq (T \cap Y^*)_R^+$ and let $l \in S \cap Y^*$. By Lemma 5(i), l is either of form y or $\bar{y}$. If $l \in S \cap Y^*$, then $l, \bar{l}, l' \in (S \cap Y^*)_R^+$. Using our assumption we get $l, \bar{l}, l' \in (T \cap Y^*)_R^+$. But then, $l \in T \cap Y^*$ follows from $l' \in (T \cap Y^*)_R^+$. This shows $S \cap Y^* \subseteq T \cap Y^*$ iff $(S \cap Y^*)_R^+ \subseteq (T \cap Y^*)_R^+$. By symmetry, $S \cap Y^* = T \cap Y^*$ iff $(S \cap Y^*)_R^+ = (T \cap Y^*)_R^+$ follows. □

Lemma 7. *Let Φ be a $QBF_\forall^2$ formula. If Φ is true, then φ is contained in every stage and in every semi-stable extension of F_Φ.*

Proof. Suppose $\Phi = \forall Y \exists Z C$ is true and let, towards a contradiction, S be a stage or a semi-stable extension of $F_\Phi = (A, R)$ with $\varphi \notin S$. By Lemma 5 (ii), we know that for each $y \in Y$, either y or $\bar{y}$ is in S. Let $I_Y = Y \cap S$. Since Φ is true we know there exists an $I_Z \subseteq Z$, such that (2.3) holds, for each $c \in C$. Consider now the set $T = I_Y \cup I_Z \cup \{\bar{x} \mid x \in (Y \cup Z) \setminus (I_Y \cup I_Z)\} \cup \{\varphi\}$. We show that T is admissible in F_Φ and that $S <_R^+ T$ holds. This will contradict our assumption in both cases, i.e. that S is a stage or a semi-stable extension of F_Φ. It is easily verified that T is conflict-free in F_Φ. Next we show that each $a \in T$ is defended by T in F_Φ. This is quite obvious for each $a \in T$ except φ, since all those arguments defend themselves. To have φ defended by T in F_Φ, each argument $c \in C$ has to be attacked by an element from T. But this is the case since (2.3) holds and by the construction of F_Φ, i.e. by the definition of attacks $\{(l, c) \mid l \in c, c \in C\}$, each such attacker c is attacked by an argument $x \in T$. It remains to show $S <_R^+ T$. By Lemma 6, $(S \cap Y^*)_R^+ = (T \cap Y^*)_R^+$, for $Y^* = Y \cup \bar{Y} \cup Y' \cup \bar{Y}'$. Moreover, by Lemma 5 (ii) either z or $\bar{z}$ in S, for each $z \in Z$; the same holds for T, by definition. We observe that $S_R^+ \cap (Z \cup \bar{Z}) = T_R^+ \cap (Z \cup \bar{Z}) = Z \cup \bar{Z}$. Moreover, we already have argued that each $c \in C$ is attacked by some argument in T. Let $A^- = A \setminus \{\varphi, \bar{\varphi}, b\}$. So far, we thus have shown that $S_R^+ \cap A^- \subseteq T_R^+ \cap A^- = Y \cup \bar{Y} \cup I_Y' \cup (\bar{Y}' \setminus \bar{I}_Y') \cup Z \cup \bar{Z} \cup C$. We finally observe that $S_R^+ \cap \{\varphi, \bar{\varphi}, b\} = \{\varphi, \bar{\varphi}\} \subset \{\varphi, \bar{\varphi}, b\} = T_R^+ \cap \{\varphi, \bar{\varphi}, b\}$, since $\varphi \notin S$ by assumption and $\varphi \in T$ by definition. This shows $S <_R^+ T$ as desired. □

We are now prepared to give our first main result.

Theorem 12. *$Skept_{sem}$ is Π_2^P-hard.*

Proof. We use our reduction from $QBF_{\forall}^{2}$ formulas to AFs and show that, for each such QBF Φ, it holds that φ is contained in all semi-stable extensions of F_Φ iff Φ is true. Since F_Φ can be constructed from Φ in polynomial time, the claim then follows.

Let $\Phi = \forall Y \exists Z C$ and $F_\Phi = (A, R)$ be the corresponding AF. The if direction is captured by Lemma 7. We prove the only-if direction by showing that if Φ is false, then there exists a semi-stable extension S of F_Φ such that $\varphi \notin S$.

In case Φ is false, there exists an $I_Y \subseteq Y$, such that for each $I_Z \subseteq Z$, there exists a $c \in C$, such that

$$\big(I_Y \cup I_Z \cup \{\bar{x} \mid x \in (Y \cup Z) \setminus (I_Y \cup I_Z)\}\big) \cap c = \emptyset. \tag{3.1}$$

Consider now a maximal (wrt. $\leq_R^+$) admissible (in F_Φ) set S, such that $I_Y \subseteq S$ (note that such a set exists, since I_Y itself is admissible in F_Φ). Using Lemma 6, one can show that S then has to be a semi-stable extension of F_Φ. To wit, let T be an admissible (in F_Φ) set such that $I_Y \not\subseteq T$. By Lemma 6 it holds that $(S \cap Y^*)_R^+ \not\subseteq (T \cap Y^*)_R^+$ and therefore $S_R^+ \not\subseteq T_R^+$. Putting this together with the maximality of S in the set $\{T \mid T \text{ is admissible in } F_\Phi \text{ and } I_Y \subseteq T\}$ we get that there is no admissible (in F_Φ) set T, such that $S_R^+ \subset T_R^+$. Hence, S is a semi-stable extension of F_Φ.

It remains to show $\varphi \notin S$. We prove this by contradiction and assume $\varphi \in S$. As S is admissible in F_Φ, S defends φ and therefore it attacks all $c \in C$. As all attacks against arguments in C come from $Y \cup \bar{Y} \cup Z \cup \bar{Z}$, the set $U = \big(I_Y \cup (S \cap (Z \cup \bar{Z})) \cup \{\bar{y} \mid y \in Y \setminus I_Y\}\big)$ attacks all $c \in C$. By Lemma 5(ii), for each $z \in Z$, either z or $\bar{z}$ is contained in S. We get an equivalent characterization for U by $U = \big(I_Y \cup I_Z \cup \{\bar{x} \mid x \in (Y \cup Z) \setminus (I_Y \cap I_Z)\}\big)$ with $I_Z = S \cap Z$. Thus, for all $c \in C$,

$$\big(I_Y \cup I_Z \cup \{\bar{x} \mid x \in (Y \cup Z) \setminus (I_Y \cup I_Z)\}\big) \cap c \neq \emptyset,$$

which contradicts assumption (3.1). □

We now turn our attention to the stage semantics.

Theorem 13. *$Skept_{stg}$ is Π_2^P-hard.*

Proof. We again use our reduction from $QBF_{\forall}^{2}$ formulas to AFs and show that, for each such QBF Φ, it holds that φ is contained in all stage extensions of F_Φ iff Φ is true. Thus, let $\Phi = \forall Y \exists Z C$ and $F_\Phi = (A, R)$ be the corresponding AF. The if direction is already captured by Lemma 7. We prove the only-if direction by showing that, if Φ is false, then there exists a stage extension S of F_Φ such that $\varphi \notin S$.

If Φ is false, there is an $I_Y \subseteq Y$, such that for each $I_Z \subseteq Z$, we have a $c \in C$ with

$$\big(I_Y \cup I_Z \cup \{\bar{x} \mid x \in (Y \cup Z) \setminus (I_Y \cup I_Z)\}\big) \cap c = \emptyset. \tag{3.2}$$

Consider the collection $W = \{S \mid I_Y \subseteq S, S \text{ is conflict-free in } F_\Phi\}$ of conflict-free sets in F_Φ. Using Lemma 6, we can show that for every conflict-free (in F_Φ) set T, $S \leq_R^+ T$ implies $I_Y \subseteq T$. For verifying $\leq_R^+$-maximality of a set $S \in W$ we thus can restrict ourselves to sets $T \in W$.

It remains to show that there is a stage extension S in W with $\varphi \notin S$. We prove that (i) for every set $S \in W$ with $\varphi \in S$, there exists a $c \in C$, such that $c \notin S_R^+$; and (ii) existence of a set

$S \in W$ such that $C \subseteq S^+_R$. Note that (i)+(ii) imply existence of a stage extension S of F_Φ with $\varphi \notin S$.

We prove (i) by contradiction and assume that $C \subseteq S^+_R$. As S is conflict-free in F_Φ and $\varphi \in S$, we get $C \cap S = \emptyset$. Since $C \subseteq S^+_R$, S attacks all $c \in C$. As all attacks against C come from $Y \cup \bar{Y} \cup Z \cup \bar{Z}$, the set $U = \big(I_Y \cup (S \cap (Z \cup Z')) \cup \{\bar{y} \mid y \in Y \setminus I_Y\}\big)$ attacks all $c \in C$. By Lemma 5(ii), for each $z \in Z$, either z or $\bar{z}$ is contained in S and so we get $U = \big(I_Y \cup I_Z \cup \{\bar{x} \mid x \in (Y \cup Z) \setminus (I_Y \cup I_Z)\}\big)$ with $I_Z = S \cap Z$. Thus, for each $c \in C$, $U \cap c \neq \emptyset$, which contradicts assumption (3.2).

To show (ii) we just construct such a set $S = U \cup V$ using $U = I_Y \cup \{\bar{y} \in Y \setminus I_Y\} \cup Z$ and $V = \{c \in C \mid \nexists u \in U \text{ with } (u, c) \in R\}$. It is easy to verify that S is conflict-free in F_Φ. It remains to show $c \in S^+_R$, for all $c \in C$. Note that for each $c \in C$ we have that either c is attacked by U or contained in V. In both cases, $c \in S^+_R$ is clear. □

Given the results for skeptical reasoning we easily obtain the complexity of credulously acceptance.

Theorem 14. *$Cred_{sem}$ is Σ^P_2-hard.*

Proof. In the proof of Theorem 12, we have shown that a $QBF^2_\forall$ formula Φ is true iff φ is contained in each semi-stable extension of F_Φ. According to Lemma 5(ii), this holds iff $\bar{\varphi}$ is not contained in any semi-stable extension of F_Φ. Thus, the complementary problem of $Cred_{sem}$ is also Π^P_2-hard. Σ^P_2-hardness of $Cred_{sem}$ follows immediately. □

The following result is proven analogously to Theorem 14.

Theorem 15. *$Cred_{stg}$ is Σ^P_2-hard.*

Our hardness results can be extended to AFs without self-attacking arguments. To this end, we adapt our reduction by replacing all self-attacking arguments in the framework F_Φ by cycles of odd length (for instance, of length 3). Figure 3.4 illustrates such a framework F^m_Φ for our example QBF. In case of semi-stable extensions, we use the fact that the only admissible set of an (unattacked) odd-length cycle is the empty set. Indeed, S is a semi-stable extension of F_Φ iff S is a semi-stable extension of F^m_Φ.

The same construction can be used to obtain hardness for stage semantics, although the argumentation is slightly different: As stage extensions only require conflict-freeness and not admissibility, the arguments of the introduced cycles may now be part of stage extensions. However, to repair the correctness proofs for the modified reduction, we use the observation that for each cycle of length 3 at most one argument can be in a stage extension S (see also Figure 3.3) and at least one argument in the cycle is not attacked by S. Thus each such cycle contributes in three different but incomparable ways to stage extensions. More formally, let A^m be the set of arguments in F^m_Φ, $X = \{b\} \cup Y' \cup \bar{Y}'$ and denote by x^- be the (unique) attacker of an argument $x \in X$ in the original framework $F_\Phi = (A, R)$. Then, we get that (i) if S is a stage extension of F_Φ, then each $S' \subseteq A^m$, such that $S' \cap A = S$ and for each $x \in X$,

$$card(S' \cap \{x_1, x_2, x_3\}) = \begin{cases} 1 & \text{if } x^- \notin S \\ 0 & \text{otherwise} \end{cases}$$

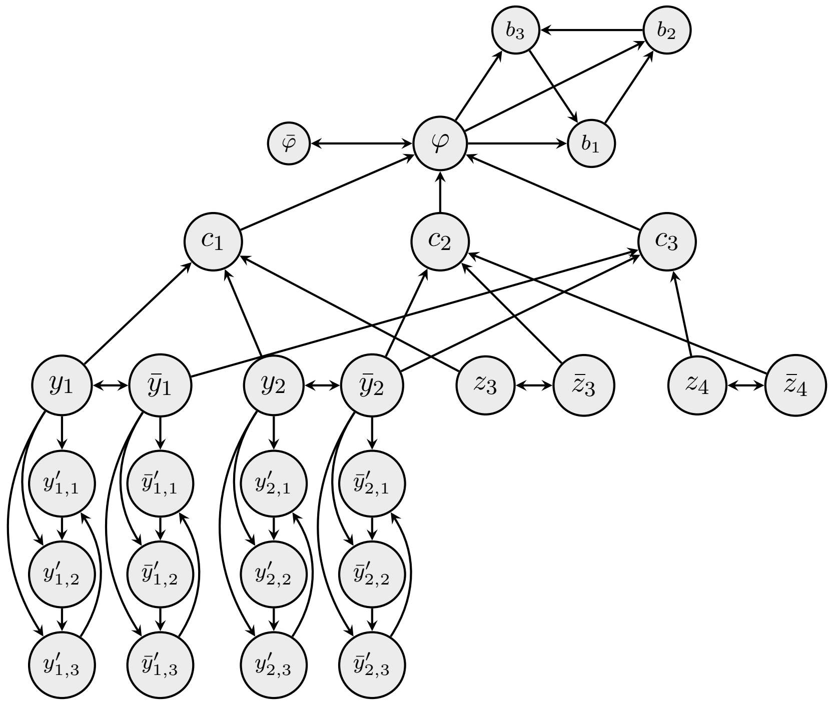


Figure 3.4: The modified framework F^m_Φ for Φ from Example 2.

is a stage extension of F^m_Φ; and (ii) if S is a stage extension of F^m_Φ, then $S \cap A$ is a stage extension of F_Φ. This correspondence between extensions suffices to show that our hardness results carry over to self-attack free AFs.

We summarize our results in terms of completeness results. The matching upper bounds for semi-stable semantics have been reported in [50]; for the stage semantics we give them in the proof of the following theorem.

Theorem 16. *$Cred_{stg}$ and $Cred_{sem}$ are Σ^P_2- complete; $Skept_{stg}$ and $Skept_{sem}$ are Π^P_2- complete. For all problems, hardness holds even for AFs without self-attacking arguments.*

Proof. Hardness is by Theorems 12– 15 and by the observations above.

For the matching upper bounds, we first recall that verifying a stage or semi-stable extension is in coNP (see Proposition 8 and [50]).

We now can give algorithms for $Cred_{stg}$ and $Cred_{sem}$ as follows. Given an AF $F = (A, R)$ and an argument $a \in A$. We guess a set $S \subseteq A$ with $a \in S$ and then use an NP-oracle (we recall that oracle calls are closed under complement), to check whether S is a stage (resp. semi-stable) extension of F. Obviously this algorithm correctly decides the considered problems. Hence, these problems are in Σ^P_2.

For $Skept_{stg}$ and $Skept_{sem}$ we argue as follows: Given an AF $F = (A, R)$, to decide if an argument $a \in A$ is contained in each stage (resp. semi-stable) extension of F, we have to prove that every set S with $a \notin S$ is *not* a stage (resp. semi-stable) extension of F. Thus, for the complementary problem, we can guess a set S with $a \notin S$ and check whether S is a stage (resp. semi-stable) extension of F. Again, this check can be done with a single call to an NP-oracle, and thus the complementary problems of $Skept_{stg}$ and $Skept_{sem}$ are in Σ_2^P. Π_2^P-membership of $Skept_{stg}$ and $Skept_{sem}$ follows immediately. □

3.4 The Complexity of Ideal Reasoning

In this section we, study the complexity of ideal reasoning for the semantics under our considerations and moreover provide complexity results which go beyond specific semantics. That is, we present two kind of complexity results for ideal reasoning. First we present generic complexity bounds for ideal reasoning problems, i.e. complexity bounds which depend on the complexity of other reasoning problems for the base semantics. Then we use these results to draw the complexity landscape for the concrete base-semantics under our considerations. In the entire section, we will assume the base semantics to be at least cf-preserving and thus guaranteeing a unique ideal extensions (cf. Proposition 2).

For a given AF $F = (A, R)$, the computational problems we are interested in here are the following:

1. Ideal acceptance $Ideal_\sigma(F, x)$: $(x \in A)$. Is there *any* $S \in \sigma^{idl}(F)$ for which $x \in S$?
2. Verifying ideal sets $Ver_\sigma^{idl}(F, S)$: $(S \subseteq A)$. Is it the case that $S \in \sigma^{idl}(F)$?
3. Non-emptiness $Exists_\sigma^{\neg\emptyset^{idl}}(F)$: Is there any $S \in \sigma^{idl}(F)$ for which $S \neq \emptyset$?
4. Verifying the ideal extension $Ver_\sigma^{ie}(F, S)$ $(S \subseteq A)$: Is it the case that $S = E_\sigma^{ie}(F)$?

For cf-preserving semantics, ideal acceptance could be equivalently defined by checking whether $x \in E_\sigma^{ie}(F)$ (since the ideal extension is the maximal ideal set). Note that we do not consider variants of skeptical acceptance here. Since ideal extensions are unique for our base-semantics, there is no need to distinguish between credulous and skeptical acceptance in terms of the ideal extensions. Moreover skeptical acceptance in terms of ideal sets would be a trivial problem, since the empty set is always an ideal set and thus no argument could be accepted. Finally, let us mention that asking whether there exists a non-empty ideal set is clearly equivalent to asking whether the ideal extension is non-empty.

3.4.1 Algorithms for Parameterised Ideal Reasoning

We first consider algorithms for constructing the ideal extensions. As the decision problems can be easily answered as soon as the ideal extension is available this gives us immediate upper bounds for the complexity of the decision problems. Here we present two algorithms for computing the ideal extension w.r.t. to a given base semantics σ.

The first is a generalisation of an algorithm presented in [46] and relies on the credulous acceptance problem for σ. We will show that such an algorithm can be used for any base semantics σ, as long as each extension $S \in \sigma(F)$ is also a preferred or naive extension. Our second algorithm is closer to the original definition of ideal sets and thus makes use of skeptical acceptance in σ. In contrast to the first algorithm this algorithm is applicable to any cf-preserving base semantics σ. Having these two algorithms at hand clearly is also of practical value. In fact, whenever both algorithms are applicable, then one can now select in view of the computational complexity of credulous acceptance and skeptical acceptance for the base semantics.

Before actually giving the algorithms we address a sub-problem, i.e. computing the maximal admissible subset of a conflict-free set. To this end we define a variation of the characteristic function (cf. Definition 20).

Definition 75. *Let $F = (A, R)$ be an AF. We define the* restricted characteristic function $\hat{\mathcal{F}}_F$ *as $\hat{\mathcal{F}}_F(E) = \mathcal{F}_F(E) \cap E$ for each $E \subseteq A$. In case no ambiguity arises we omit the subscript and just write $\hat{\mathcal{F}}$.*

Using the above definition we obtain a nice characterisation of the maximal admissible subset of a conflict-free set (cf. Lemma 2).

Lemma 8. *Let $F = (A, R)$ be an AF and $C \subseteq A$ a conflict-free set. For the $\subseteq$-maximal set $S \subseteq C$ admissible in F, it holds that $A = \hat{\mathcal{F}}^{|C|}(C)$.*

Proof. Obviously $\hat{\mathcal{F}}$ is a monotone operator and the series $(\ \hat{\mathcal{F}}^i(C)\ )_{i \geq 0}$ is non-increasing. Further, the empty set is a lower bound and thus a fixed-point is reached after at most $|C|$ steps. We claim that this fixed point is also the desired $\subseteq$-maximal admissible set S, i.e. $A = \hat{\mathcal{F}}^{|C|}(C)$.

By definition, the fixed point $\hat{\mathcal{F}}^{|C|}(C)$ defends all its arguments and is conflict-free (since $\hat{\mathcal{F}}^{|C|}(C) \subseteq C$ and by assumption that C is conflict-free). Hence it holds that $\hat{\mathcal{F}}^{|C|}(C)$ is an admissible set. To complete the proof we consider maximality and show that $A \subseteq \hat{\mathcal{F}}^i(C)$ for $i \geq 0$. As A is admissible we have that $\hat{\mathcal{F}}(S) = S$ and further for each set B with $S \subseteq B$ that $S \subseteq \hat{\mathcal{F}}(B)$. Moreover $S \subseteq C$ and thus for $i \geq 0$ it holds that $S \subseteq \hat{\mathcal{F}}^i(C)$. Hence we have that $S = \hat{\mathcal{F}}^{|C|}(C)$. □

We mention that Lemma 8 is basically a restatement of algorithms used for reasoning in bipartite AFs (see [44]) making them more convenient for our purposes.

Proposition 9. *Let $F = (A, R)$ be an AF and $S \subseteq A$ a conflict-free set. The $\subseteq$-maximal set $A \subseteq S$ admissible in F can be computed in polynomial time.*

Proof. We can compute the set A by applying the characterisation in Lemma 8. Further applying the function $\hat{\mathcal{F}}$ is clearly in polynomial time and as we apply it just linearly often the whole procedure is in polynomial time. □

We are now prepared to present the characterisation of ideal sets (resp. the ideal extension) which underlies our first algorithm.

Proposition 10. *If σ is prf-preserving, then for each AF $F = (A, R)$ with $|\sigma(F)| \geq 1$ the following relations hold:*

C1. $S \in \sigma^{idl}(F) \ \Leftrightarrow\ S \in adm(F)\ \&\ \forall y \in S^{\ominus} \neg Cred_{\sigma}(F, y)$

C2. $x \in E^{ie}_{\sigma}(F) \;\Leftrightarrow\; \forall y \in \{x\}^{\ominus} \left(\; \neg Cred_{\sigma}(F, y) \;\&\; \{y\}^{\ominus} \cap E^{ie}_{\sigma}(F) \neq \emptyset \;\right)$

Proof. For (C1), suppose first that $S \in \sigma^{idl}(F)$. Certainly $S \in adm(F)$ (by definition), so consider any $y \in S^{\ominus}$. If, in contradiction to the claim, we have $Cred_{\sigma}(F, y)$ then there is some set $T \in \sigma(F)$ for which $y \in T$. We must, however, have $S \subseteq T$ (since S is an ideal set w.r.t. σ), leading to $S \cup \{y\} \subseteq T$ which contradicts $T \in prf(F)$ (since T would fail to be conflict-free). On the other hand, suppose S is such that $S \in adm(F)$ and every $y \in S^{\ominus}$ satisfies $\neg Cred_{\sigma}(F, y)$. We show that this leads to $S \in \sigma^{idl}(F)$. If this failed to be the case we can find $T \in \sigma(F)$ for which $S \setminus T \neq \emptyset$. Consider the set $S \cup T$: this must be conflict-free since $S \in adm(F)$, $T \in prf(F)$ so that $S \cup T \notin cf(F)$ would imply the presence of arguments $s \in S$, $t \in T$ with $(s, t) \in R$ or $(t, s) \in R$. From the premises the latter is ruled out since it would lead to the contradiction $\neg Cred_{\sigma}(F, t)$; the former possibility, however, is also ruled out since from $T \in prf(F)$, $(s, t) \in R$ yields $u \in T$ with $(u, s) \in R$ and, again, we obtain a contradiction. It follows that $S \cup T \in cf(F)$ and also $S \cup T \in adm(F)$. The set T, however, is also in $prf(F)$ so that $S \cup T = T$, i.e. $S \subseteq T$ as required.

For (C2), if $x \in E^{ie}_{\sigma}(F)$ then no $y \in \{x\}^{\ominus}$ can have $Cred_{\sigma}(F, y)$ (from C1) and, trivially from the fact $E^{ie}_{\sigma}(F) \in adm(F)$ we obtain, for each $y \in \{x\}^{\ominus}$, that $\{y\}^{\ominus} \cap E^{ie}_{\sigma}(F) \neq \emptyset$. For the converse direction consider any $x \in A$ for which each $y \in \{x\}^{\ominus}$ has $\neg Cred_{\sigma}(F, y)$ and $\{y\}^{\ominus} \cap E^{ie}_{\sigma}(F) \neq \emptyset$. The set $T = \{x\} \cup E^{ie}_{\sigma}(F)$ is admissible (by similar arguments to those used in C1), and furthermore, any $y \in T^{\ominus}$ is such that $\neg Cred_{\sigma}(F, y)$ (that this is the case for $\{x\}$ holds via the premise, and that it holds for $E^{ie}_{\sigma}(F)$ is immediate from C1). We deduce that $T = \{x\} \cup E^{ie}_{\sigma}(F) \in \sigma^{idl}(F)$ and, since $E^{ie}_{\sigma}(F)$ is $\subseteq$-maximal, we obtain $x \in E^{ie}_{\sigma}(F)$ as required. □

In the case of standard ideal semantics the characterisation of Proposition 10 has previously been shown in [43, 46]. In addition, we now see that it holds also for the case when *semi-stable* semantics are employed as base semantics for ideal reasoning. If we consider stable-consistent AFs then the characterisation of Proposition 10 also applies when considering stable as base-semantics.

However, the above characterisation does not apply to semantics which are not based on admissibility, for instance if stage semantics is considered. Thus our next step is to give a similar characterisation for semantics which are $naive$-preserving. The only subtle difference is due to the fact that naive semantics do not take the orientation of attacks into account. Thus, we have to add $S^{\oplus}$ (resp. $\{x\}^{\oplus}$) to the conditions from Proposition 10. The proof then proceeds quite similar to the one of Proposition 10.

Proposition 11. *If σ is naive-preserving, then for each AF (A, R) with $|\sigma(F)| \geq 1$ the following relations hold:*

C1'. $S \in \sigma^{idl}(F) \;\Leftrightarrow\; S \in adm(F) \;\&\; \forall y \in S^{\ominus} \cup S^{\oplus} \neg Cred_{\sigma}(F, y)$

C2'. $x \in E^{ie}_{\sigma}(F) \;\Leftrightarrow\; \forall\, y \in \{x\}^{\ominus} \cup \{x\}^{\oplus} \;\left(\; \neg Cred_{\sigma}(F, y) \;\;\&\; \{y\}^{\ominus} \cap E^{ie}_{\sigma}(F) \neq \emptyset \;\right)$

Proof. For (C1'), suppose first that $S \in \sigma^{idl}(F)$. Certainly $S \in adm(F)$ (by definition), so consider any $y \in S^{\ominus} \cup S^{\oplus}$. If, in contradiction to the claim, we have $Cred_{\sigma}(F, y)$ then there is some set $T \in \sigma(F)$ for which $y \in T$. We must, however, have $S \subseteq T$ (since S is an ideal set w.r.t. σ), leading to $S \cup \{y\} \subseteq T$ which contradicts $T \in naive((A,R))$ (since T would fail to be conflict-free). On the other hand, suppose S is such that $S \in adm(F)$ and every $y \in S^{\ominus}$ satisfies $\neg Cred_\sigma(F,y)$. We show that this leads to $S \in \sigma^{idl}(F)$. If this failed to be the case we can find $T \in \sigma(F)$ and $x \in S$ such that $x \notin T$. By assumption we have that $\{x\}^{\ominus} \cup \{x\}^{\oplus} \cap T = \emptyset$ and thus that the set $T \cup \{x\}$ is a conflict-free set. As T is defined as $\subseteq$-maximal conflict-free set this leads to the desired contradiction.

For (C2'), if $x \in E^{ie}_\sigma(F)$ then no $y \in \{x\}^{\ominus} \cup \{x\}^{\oplus}$ can be credulously accepted. From the fact $E^{ie}_\sigma(F) \in adm(F)$ we obtain, for each $y \in \{x\}^{\ominus}$, that $\{y\}^{\ominus} \cap E^{ie}_\sigma(F) \neq \emptyset$. Further as by assumption $x \in E^{ie}_\sigma(F)$ we have that $\forall y \in \{x\}^{\oplus}$, $\{y\}^{\ominus} \cap E^{ie}_\sigma(F) \neq \emptyset$. For the converse direction consider any $x \in A$ for which each $y \in \{x\}^{\ominus} \cup \{x\}^{\oplus}$ has $\neg Cred_\sigma(F,y)$ and $\{y\}^{\ominus} \cap E^{ie}_\sigma(F) \neq \emptyset$. The set $T = \{x\} \cup E^{ie}_\sigma(F)$ is conflict free and defends x, i.e. the set is admissible. Furthermore, any $y \in T^{\ominus} \cup T^{\oplus}$ is such that $\neg Cred_\sigma(F,y)$. We deduce that $T = \{x\} \cup E^{ie}_\sigma(F) \in \sigma^{idl}(F)$ and, since $E^{ie}_\sigma(F)$ is *maximal*, obtain $x \in E^{ie}_\sigma(F)$ as required. □

We mention that combining the proofs of Proposition 10 and Proposition 11, one can show that (C1') and (C2') also hold for semantics which are neither purely prf-preserving nor purely $naive$-preserving but satisfying $\sigma(F) \subseteq prf(F) \cup naive((A,R))$. That is, it is sufficient that each σ-extension is at least a preferred or naive extensions.

The characterisations (C2) and (C2') from above results suggest how to compute the ideal extension w.r.t. a semantics σ, in case we have given a function that decides credulous acceptance for σ. Algorithm 1 describes this idea.

Algorithm 1. *Input: AF $F = (A,R)$, function $Cred_\sigma$ deciding credulous acceptance.*

1. *Determine the following subset of A:*

$$A_{PSA} = \{\ x \in Cred_\sigma(F) \mid \{x\}^{\ominus} \cup \{x\}^{\oplus} \cap Cred_\sigma(F) = \emptyset\}$$

with $Cred_\sigma(F)$ being the set of credulously accepted arguments w.r.t. σ.

2. *Return $E = (\hat{\mathcal{F}}_F)^n(A_{PSA})$ where $n = |A_{PSA}|$.*

Notice that even the formulation of the algorithm is quite different to that in [45]. Our algorithm, when instantiated with preferred semantics, essentially does the same, we just avoid the reduction to bipartite AFs [44] and directly apply the corresponding techniques. The correctness of this algorithm (for appropriate base semantics as outlined above) is an consequence of Propositions 10 and 11.

Theorem 17. *For any semantics σ which is prf-preserving or $naive$-preserving and AF F with $|\sigma(F)| \geq 1$, Algorithm 1 constructs the ideal extension E^{ie}_σ.*

Proof. By Proposition 10 and Proposition 11 we have that $E^{ie}_\sigma \subseteq A_{PSA}$. Next we show that A_{PSA} is conflict free. Towards a contradiction let us assume that $a, b \in A_{PSA}$ and $(a,b) \in R$. By definition of A_{PSA}, $a, b \in Cred_\sigma(F)$ and as $a \in \{b\}^{\ominus}$ also $a \notin Cred_\sigma(F)$, a contradiction. Thus, by Lemma 8 the maximal admissible subset of A_{PSA} is given by $\hat{\mathcal{F}_F}^n(A_{PSA})$ □

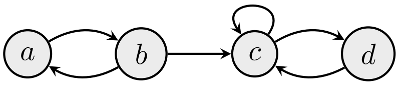


Figure 3.5: Ideal semantics w.r.t. $resGr$.

Although Algorithm 1 is already applicable to a wide range of base semantics, it does not work when considering resolution-based grounded as base semantics. For instance, consider the AF $F = (\{a,b,c,d\}, \{(a,b),(b,a), (b,c),(c,c),(c,d),(d,c)\})$ illustrated in Figure 3.5. In this we have $resGr(F) = \{\{a\},\{b,d\}\}$ so that the $E^{ie}_{resGr} = \emptyset$. However, the conditions from above propositions apply for the set $\{d\}$, i.e. $\{d\}$ is admissible and none of its neighbours is credulously accepted w.r.t. $resGr$. In fact, $\{d\}$ is the standard ideal extension, i.e. using preferred as base semantics.

Hence Proposition 10 and Proposition 11 fail to hold for base semantics which are resolution based. Thus, Algorithm 1 is not fully satisfying, as it cannot capture all base semantics under our considerations. This motivates Algorithm 2 which follows the definition of ideal semantics more closely: it first computes the set of all *skeptically accepted* arguments and then iteratively computes the maximal admissible subset.

Algorithm 2. *Input: AF $F = (A,R)$, function $Skept_\sigma$ deciding the skeptical acceptance.*

- *Compute the set $A_{sa} = \{\, x \ : \ Skept_\sigma(F,x)\}$*
- *Return $E = \hat{\mathcal{F}_F}^n(A_{sa})$ where $n = |A_{sa}|$.*

We next show that this algorithm is correct for every reasonable base semantics, i.e. for base semantics that are cf-preserving. The following lemma captures the correctness of the fixed-point iteration in the algorithm.

Theorem 18. *For any semantics σ which is cf-preserving and AF F with $|\sigma(F)| \geq 1$, Algorithm 2 constructs $E^{ie}_\sigma(F)$.*

Proof. Since σ is cf-preserving, the set A_{sa} is also conflict-free. Thus by Lemma 8, $E^{ie}_\sigma(F) = \hat{\mathcal{F}_F}^{|A_{sa}|}(A_{sa})$. □

Theorem 18 is more general than Theorem 17 in the sense that whenever a base semantics σ satisfies the conditions to apply Algorithm 1 one can also apply Algorithm 2. However, Algorithm 1 becomes valuable if deciding credulous acceptance is easier than deciding skeptical acceptance (a situation which holds for standard ideal semantics, as already mentioned earlier).

3.4.2 Instantiations

Here we consider concrete base semantics for ideal reasoning, briefly recall relevant results from Section 2.1.2 and present results which will be helpful in the following complexity analysis.

First let us mention that for semantics that always include the empty-set as an extension the ideal extension is clearly the empty-set, and reasoning problems become trivial. Hence, we do not consider adm and cf as base-semantics here. Moreover, let us recall that for base semantics $\sigma \in \{grd, com\}$, we have that the ideal extension coincides with the grounded extension (see Proposition 5). Thus the complexity results for grounded semantics carry over.

Hence, we are interested in the computational complexity of ideal semantics w.r.t. base semantics, $resGr$, prf, sem, $naive$, stg, and stb. Since all these semantics σ are cf-preserving, we know that σ^{ie} is a unique status semantics (cf. Proposition 2). Moreover, for semantics $\sigma \in \{prf, sem, resGr\}$, σ has the reinstatement property, and thus by Proposition 3, we know that for these base semantics, the ideal extension is a complete set for any AF. In general, this does not hold for $\sigma \in \{naive, stage\}$.

Let us next have a closer look on the ideal extension w.r.t. naive semantics. We can give the following characterisation of $naive^{ie}$.

Proposition 12. *For any AF $F = (A, R)$,*

$$E^{ie}_{naive}(F) \quad = \quad \max\{A : A \in adm(F), A \subseteq A_{sa}\}$$

with $A_{sa} = \{x : (x, x) \notin R, \{x\}^{\ominus} \cup \{x\}^{\oplus} \subseteq \{y : (y, y) \in R\}$.

Proof. It suffices to show that the set A_{sa} is the set of skeptically accepted arguments w.r.t. naive extensions. It is easy see to that for all $x \in A$ with $(x, x) \notin R$, $Cred_{naive}(F, x)$ holds, as the set $\{x\}$ is clearly conflict-free (cf. proof of Theorem 11). It follows that only those arguments neither attacked by nor attacking such arguments can belong to every set in $naive(F)$. □

For AFs without self-attacking arguments this characterisation simplifies as follows.

Corollary 1. *For any AFs (A, R) without self-attacking arguments $E^{ie}_{naive}(F)$ matches the set of skeptically accepted arguments w.r.t. naive semantics; i.e. $E^{ie}_{naive}(F) \;=\; \{\, x : \{x\}^{\ominus} \cup \{x\}^{\oplus} = \emptyset\}$.*

Proof. Follows from Proposition 12 when $\{y : (y, y) \in R\} = \emptyset$ is assumed. □

So we have that for naive semantics, the usage of self-attacking arguments gives us additional expressive power, in particular for ideal reasoning. This is in contrast to admissibility based semantics where self-attacking arguments can always be replaced by odd length cycles without changing the extensions of the framework.

3.4.3 Generic Complexity Results

We first exploit the fact that the decision problems under our considerations are easy, as soon we have computed the ideal extension. That is we can use the algorithms given in Section 3.4.1 to get upper bounds for the complexity of these problems.

Theorem 19. *For any semantics σ which is prf-preserving or $naive$-preserving, the problems $Ideal_{\sigma}$, Ver^{idl}_{σ}, $Exists^{\neg\emptyset}{}^{idl}_{\sigma}$, and Ver^{ie}_{σ} can be decided in $P^{\mathcal{C}}_{||}$, where $Cred_{\sigma}, Exists_{\sigma} \in \mathcal{C}$.*

Proof. An algorithm would first use the oracle to test whether there exists an extension for the base-semantics. If not then, by definition, the ideal extension is empty, otherwise, by Theorem 17, we can use Algorithm 1 to compute E^{ie}_{σ}. Moreover the set A_{PSA} in Algorithm 1 is directly constructed in $\mathrm{FP}^{\mathcal{C}}_{||}$ and further the maximal admissible set can be found in polynomial time (see Proposition 9). Hence, Algorithm 1 is a $\mathrm{P}^{\mathcal{C}}_{||}$ algorithm constructing E^{ie}_{σ}. Given E^{ie}_{σ} one can clearly decide all the mentioned problems in polynomial time. □

Theorem 20. *For any semantics σ which is cf-preserving, the problems $Ideal_{\sigma}$, Ver^{idl}_{σ}, $Exists^{\neg\emptyset}{}^{idl}_{\sigma}$, and Ver^{ie}_{σ} can be decided in $P^{\mathcal{C}}_{||}$, where $Skept_{\sigma}, Exists_{\sigma} \in \mathcal{C}$.*

Proof. An algorithm would first use the oracle to test whether there exists an extensions for the base-semantics. If not then, by definition, the ideal extension is empty, otherwise, by Theorem 18, we can use Algorithm 2 to compute E^{ie}_{σ}. The set A_{sa} can be directly constructed in $\mathrm{FP}^{\mathcal{C}}_{||}$. Further the maximal admissible set can be found in polynomial time (see Proposition 9). Hence Algorithm 2 is a $\mathrm{P}^{\mathcal{C}}_{||}$ algorithm. Again given having constructed E^{ie}_{σ} one can decide all the mentioned problems in polynomial time. □

Note that for all the semantics under our considerations, except stable semantics, the problem $Exists_{\sigma}$ is trivial and therefore the condition $Exists_{\sigma} \in \mathcal{C}$ in the above theorems can be omitted.

We have that upper bounds for constructing the ideal extension immediately lead to upper bounds for decision problems, but in general this is not the most efficient way. The following theorem provides more sophisticated upper bounds for the complexity of the decision problems we are interested in. To this end, we make use of the complexity of the verification problem for the base semantics.

Theorem 21. *Let σ be a cf-preserving semantics with $|\sigma(F)| \geq 1$ for each AF F, and let $\mathcal{V}$ be the complexity of the problem Ver_{σ}. Then the following holds:*

i. $Ideal_{\sigma} \in coNP^{\mathcal{V}}$

ii. $Ver^{idl}_{\sigma} \in coNP^{\mathcal{V}}$

iii. $Exists^{\neg\emptyset}{}^{idl}_{\sigma} \in coNP^{\mathcal{V}}$

iv. $Ver^{ie}_{\sigma} \in NP^{\mathcal{V}} \wedge coNP^{\mathcal{V}}$

Proof. Given a framework (A, R), a base semantics σ, an argument $x \in A$ and a set $S \subseteq A$.

We prove (i) by providing an $\mathrm{NP}^{\mathcal{V}}$ algorithm for disproving x to be in the ideal extension.

1. Guess extensions $E_1, \dots, E_n$ (for $n = |A|$)
2. Verify extensions using the $\mathcal{V}$-oracle
3. compute $S_{sa} := \bigcap_{i=1}^{n} E_i$
4. compute the $\subseteq$-maximal admissible set $A \subseteq S_{sa}$.
5. accept iff $x \notin A$

Note that since σ is cf-preserving, Lemma 8 shows that A in Step 4 can be computed in polynomial time.

A few words about the correctness of the above algorithm: First we have that the extensions $E_1, \ldots, E_n$ are not necessarily different (there may not exist n different extensions). However, for at least one guess the set S_{sa} coincides with the set of skeptical accepted arguments, i.e. $\bigcap_{i=1}^{n} E_i = \bigcap_{E \in \sigma((A,R))} E$. This is by the fact the number of guessed extensions is equal to the number of arguments in framework. Moreover, for every guess it holds that $\bigcap_{E \in \sigma((A,R))} E \subseteq S_{sa}$. If $x \in E_{\sigma}^{ie}((A,R))$, i.e. there is an admissible set $S' \subseteq \bigcap_{E \in \sigma((A,R))} E$ with $x \in S'$, then for every such guess, we have $S' \subseteq S_{sa}$ and thus $x \in A$. Hence the algorithm does not accept such an instance. Let us now consider the case that $x \notin E_{\sigma}^{ie}((A,R))$, i.e. for S' being the maximal admissible set of $\bigcap_{E \in \sigma((A,R))} E$ we have $x \notin S'$. Then x is accepted by the computation where $S_{sa} = \bigcap_{E \in \sigma((A,R))} E$. This thus solves the complementary problem to Ideal_{σ} as desired.

We get (ii) by a simple adaptation of the above algorithm: instead of testing $x \notin A$ one tests whether $S \not\subseteq A$, which disproves the set S to be an ideal set.

For (iii) we use another adaptation of the above algorithm, now one tests for $A = \emptyset$, which proves that $E_{\sigma}^{ie} = \emptyset$.

(iv): To verify that $E_{\sigma}^{ie} = S$ one can first use the coNP$^{\mathcal{V}}$-algorithm to verify that $S \in \mathcal{E}_{\sigma}^{idl}((A,R))$. Then one can use the above NP$^{\mathcal{V}}$-algorithm to disprove the ideal acceptance for all arguments $R \setminus S$. As $E_{\sigma}^{ie} = S$ iff both of these algorithms accept S we have that $\mathit{Ver}_{\sigma}^{ie} \in$ NP$^{\mathcal{V}} \wedge$ coNP$^{\mathcal{V}}$. □

Again the above theorem only applies to semantics σ satisfying $|\sigma(F)| \geq 1$ for arbitrary AFs. However, we can generalise the complexity bounds taking the complexity of deciding whether there exists an extension of the base-semantics into account.

Corollary 2. *Let σ be a cf-preserving semantics, and let $\mathcal{V}$ be the complexity of the problem Ver_{σ}. Then the following holds:*

i. $\mathit{Ideal}_{\sigma} \in NP^{\mathcal{V}} \wedge coNP^{\mathcal{V}}$

ii. $\mathit{Ver}_{\sigma}^{idl} \in NP^{\mathcal{V}} \wedge coNP^{\mathcal{V}}$

iii. $\mathit{Exists}^{\neg\emptyset}{}_{\sigma}^{idl} \in NP^{\mathcal{V}} \wedge coNP^{\mathcal{V}}$

iv. $\mathit{Ver}_{\sigma}^{ie} \in NP^{\mathcal{V}} \wedge coNP^{\mathcal{V}}$

Proof. We have that $\mathit{Exists}_{\sigma} \in$ NP$^{\mathcal{V}}$ as Exists_{σ} can be decided by guessing an extension and verifying it. Combining this additional check with the results of Theorem 21 yields the above complexity bounds. □

In the following we give generic lower bounds, i.e. generic hardness results, for the problems $\mathit{Ver}_{\sigma}^{idl}$ and Ideal_{σ} depending on the complexity of the problem Cred_{σ}. To this end, we introduce the mutual attack property of a semantics:

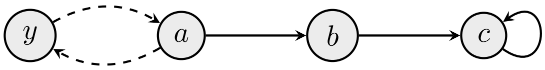


Figure 3.6: Example illustrating that sem and stg do not have the mutual attack property.

Definition 76. *A pair* $((A,R),x)$ *of an AF* (A,R) *and an argument* $x \in A$ *satisfies the* mutual attack property *w.r.t. a semantics* σ *iff*

$$\mathit{Cred}_\sigma((A,R),x) \Rightarrow \mathit{Cred}_\sigma((A \cup \{y\}, R \cup \{(x,y),(y,x)\}),x)$$

(where $y \notin A$*).*[2] *Further we say that a semantics* σ *satisfies the* mutual attack property *iff each* $((A,R),x)$ *has the* mutual attack property *w.r.t.* σ*.*

Proposition 13. *Sematics* adm*,* $comp$*,* prf*,* stb*,* $resGr$*, and* $naive$ *all have the mutual attack property.*

Proof. Consider an AF $F = (A,R)$ and $E \in adm(F)$ and an argument $x \in E$. If we extend the AF F with a new argument y and attacks $(x,y),(y,x)$, denoted by $F' = (A \cup \{y\}, R \cup \{(x,y),(y,x)\})$, then y attacks E and also E attacks y. Thus E is still an admissible set, i.e. $E \in adm(F')$. Next as $x \in E$ we have that adding y would cause a conflict and thus that $E \in \sigma(F) \Rightarrow E \in \sigma(F')$ for $\sigma \in \{comp, pr\}$. Further if $E \in stb(F)$ then clearly $E \in stb(F')$ as E attacks the new argument y. For the naive semantics, we have that $\mathit{Cred}(F,x)$ holds iff $(x,x) \notin R$. Thus adding the argument y does not change anything, as well.

Finally let us consider $resGr$ semantics. For each full resolution we either keep the attack (x,y) or (x,y). First it is easy to see that if the argument x is not credulously accepted in (A,R), the additional argument does not help and also it is not credulously accepted in $(A \cup \{y\}, R \cup \{(x,y),(y,x)\})$. Let us assume there exists $E \in resGr((A,R))$, and further let β be such that $E = E_{gr}((A, R \setminus \beta))$. First if we keep (x,y) we have that $E = E_{gr}((A \cup \{y\}, R \setminus \beta \cup \{(x,y)\}))$ and thus that E is still a candidate for a $resGr$-extension. What remains to show is that E is still $\subseteq$-minimal. But as $y \notin E$ and $y \in E_{gr}((A \cup \{y\}, R \setminus \beta' \cup \{(y,z)\}))$ for arbitrary β' this is also satisfied. □

To see that sem and stg semantics do not have the mutual attack property let us once again consider the AF (A,R) of the form $(\{a,b,c\}, \{(a,b),(b,c),(c,c)\})$ (see also Figure 3.6): $\mathit{Cred}_\sigma((A,R),a)$ holds for both sem and stg. But when adding an argument y together with a mutual attack to a, then a is no longer credulously accepted, neither for sem nor for stg semantics.

The following theorem exploits the mutual attack property to provide generic hardness results. It follows from a more general observation which we give below in Proposition 14.

[2] In terms of the work by Cayrol et al. [29], the pair $((A,R),x)$ satisfies the mutual attack property iff adding $(\{y\}, \{(x,y),(y,x)\})$ to the AF (A,R) is *partial monotone* for x.

Theorem 22. *If a semantics σ satisfies the mutual attack property, is prf-preserving (resp. $naive$-preserving), for each AF F it holds that $|\sigma(F)| \geq 1$, and $Cred_\sigma$ is $\mathcal{C}$–complete for some complexity class $\mathcal{C}$ which is closed under $\cup$ then:*

a. Ver^{idl}_{σ} *is co$\mathcal{C}$–complete.*

b. $Ideal_\sigma$ *is co$\mathcal{C}$–hard.*

As mentioned above, some semantics σ in general do not satisfy the mutual attack property (e.g. sem-semantics) and thus the above theorem does not directly apply to them. But there may exist a infinite class of pairs $((A, R), a)$ satisfying the mutual attack property w.r.t. σ. Such a class of pairs is in particular interesting if credulous reasoning for these pairs has the same complexity as for arbitrary instances. We will exploit this fact — which is proven in the forthcoming result — in the next subsection.

Proposition 14. *If a semantics σ satisfies the mutual attack property, is prf-preserving (resp. $naive$-preserving), for each AF F it holds that $|\sigma(F)| \geq 1$, $Cred_\sigma \in \mathcal{C}$ for some complexity class $\mathcal{C}$ which is closed under $\cup$, and there exist $\mathcal{C}$–hard instances of the $Cred_\sigma$ problem satisfying the mutual attack property then:*

a. Ver^{idl}_{σ} *is co$\mathcal{C}$–complete.*

b. $Ideal_\sigma$ *is co$\mathcal{C}$–hard.*

Proof. We show (a). The proof for part (b) uses an identical reduction. The complementary problem $\neg\, Ver^{idl}_{\sigma}$ is in $\mathcal{C}$: using Proposition 10 (resp. Proposition 11) $S \notin \sigma^{idl}$ if it is either not admissible or some $y \in S^{\ominus}$ (resp. $y \in S^{\ominus} \cup S^{\oplus}$) is credulously accepted wrt σ. From $Cred_\sigma \in \mathcal{C}$ and $\mathcal{C}$ closed under $\cup$ the latter condition can be tested in $\mathcal{C}$. For the lower bound given an instance (A, R, x) of $Cred_\sigma$ form the instance $(A \cup \{y\}, R \cup \{(x, y), (y, x)\}, \{y\})$ of Ver^{idl}_{σ}. This is accepted if and only if (A, R, x) fails to be accepted as an instance of $Cred_\sigma$. □

3.4.4 Exact Complexity for Instantiations

We finally provide exact bounds for computational problems of ideal reasoning w.r.t. to base semantics semi-stable, stage, resolution-based grounded, naive and stable, partly exploiting the generic results from the previous subsection.

We start with our results for semi-stable and stage semantics.

Theorem 23.

i. Ver^{idl}_{sem}, Ver^{idl}_{stg} *are* Π^P_2*–complete.*

ii. $Ideal_{sem}$, $Ideal_{stg}$ *are* Π^P_2*–complete.*

iii. $Exists^{\neg\emptyset}{}^{idl}_{sem}$, $Exists^{\neg\emptyset}{}^{idl}_{stg}$ *are* Π^P_2*–complete.*

iv. Ver^{ie}_{sem}, Ver^{ie}_{stg} *are* D^P_2*–complete.*

Proof. The membership part for i–iv follows directly by Theorem 21 and the fact that Ver_{sem}, Ver_{stg} are in coNP.

To prove the desired hardness results we use Reduction 1 from the Π_2^P–hard problem $QBF_2^\forall$ to the problems Skept_{sem}, Skept_{stg}, coCred_{sem} and coCred_{stg}. That is for a given $QBF_\forall^2$ Φ we construct the AF F_Φ as defined in Reduction 1. For illustration, we depict an example in Figure 3.7 (for the moment ignore the dotted addition with argument y). As shown in Section 3.3 the argument φ is skeptically accepted (for stg and sem) iff $\bar{\varphi}$ is not credulously accepted (for stg and sem) iff Φ is valid. One can see that the pairs $(F_\Phi, \bar{\varphi})$ are Σ_2^P–hard instances for Skept_{sem} and Skept_{stg} and moreover satisfy the mutual attack property (for the argument Ψ). Thus by Proposition 14 we immediately get the desired Π_2^P lower bounds for Ver_{sem}^{idl}, Ver_{stg}^{idl}, Ideal_{sem}, and Ideal_{stg}.

To show (iii), we consider a restricted class of $QBFs$, namely those $QBFs$ Φ where $\bigwedge_{c \in C} c$ has a model M, with $M \cap Z = \emptyset$ and a model M', with $Z \subseteq M'$. The $QBF_2^\forall$ problem remains Σ_2^P-hard for those formulas. To show this we give a reduction from an arbitrary $QBF_2^\forall$ to a restricted one. Given a QBF formula $\Phi = \forall Y \exists Z \bigwedge_{c \in C} c$, one can build the the restricted QBF $\Phi' = \forall Y \cup \{u\} \exists Z \bigwedge_{c \in C'} c$, with $C' = \{c \cup \{u\} : c \in C\}$. On can see that Φ is valid iff Φ' is valid. Further $\{u\}$ is a partial model of $\bigwedge_{c \in C'} c$ and thus one can find both a model M, with $M \cap Z = \emptyset$ and a model M', with $Z \subseteq M'$.

We extend F_Φ by a "mutual attack" argument y, i.e. we ebuild $F'_\Phi = (A_{F_\Phi} \cup \{y\}, R_{F_\Phi} \cup \{(y, \bar{\varphi}), (\bar{\varphi}, y)\})$ (see Figure 3.7 again). By Propositions 10 and 11 we have that y is ideal accepted iff Ψ is not credulously accepted iff Φ is skeptically accepted. In the next step we identify the arguments of $G_\Phi = (A, R)$, which are possibly ideal accepted. Obviously the self-attacking arguments $Y' \cup \bar{Y}' \cup \{b\}$ are not ideal accepted. Further in Section 3.3 we showed that for every argument $x \in Y \cup \bar{Y}$ there exists a semi-stable (resp. stage) extensions E such that $\bar{x} \in E$ and thus $x \notin E$. Hence none of the arguments $Y \cup \bar{Y}$ is skeptically accepted. Moreover it was shown that each model of Φ corresponds to a semi-stable (resp. stage) extension and thus by the existence of the models M, M' we conclude that none of the arguments $z \in Z \cup \bar{Z}$ is skeptically accepted. The arguments $c \in C$, φ are not defended by ideal accepted arguments and thus they are not ideal accepted. Further as the CNF is satisfiable we have that φ is always in at least one semi-stable extension and thus $\bar{\varphi}$ is not ideal accepted. Thus the only argument that can be ideal accepted is y. Hence for G'_Φ, it holds that $E_{sem}^{ie} \neq \emptyset$ iff y is ideal accepted iff Φ is a valid QBF.

To show (iv), we reduce an instance $(\neg\Phi_1, \Phi_2)$ of the restricted $QBF_2^\exists$-$QBF_2^\forall$ problem to an AF (A, R) and a set $S \subseteq A$ such that $S = E_\sigma^{ie}((A, R))$ iff $(\neg\Phi_1, \Phi_2)$ is a yes-instance of the restricted $QBF_2^\exists$-$QBF_2^\forall$ problem. By the observations in (iii) we get that for the AF $G'_{\Phi_1} \dot\cup G'_{\Phi_2}$ it holds that $\{y_2\}$ is the ideal extension iff $(\neg\Phi_1, \Phi_2)$ is a positive instance of the $QBF_2^\exists$-$QBF_2^\forall$ problem. □

We continue with ideal semantics w.r.t. resolution-based grounded semantics, resGr. Since resGr does neither preserve prf nor naive we cannot make direct use of Proposition 14. However, the main ideas of the proofs are similar as for the previous result.

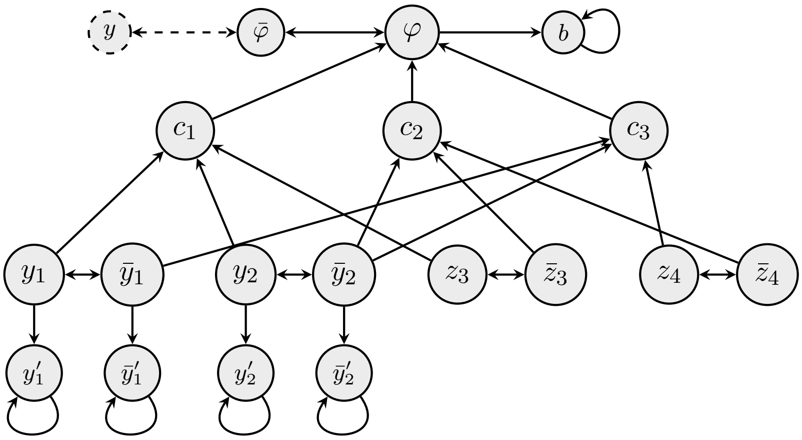


Figure 3.7: An example for the AFs G_Φ, G'_Φ used in the proof of Theorem 23, using the QBF $\Phi = \forall Y \exists Z \{\{y_1, y_2, z_3\}, \{\bar{y}_2, \bar{z}_3, \bar{z}_4\}, \{\bar{y}_1, y_2, z_4\}\}$.

Theorem 24.

i. $\mathit{Ver}^{idl}_{resGr}$ *is coNP–complete.*

ii. Ideal_{resGr} *is coNP–complete.*

iii. $\mathit{Exists}^{\neg\emptyset}{}^{idl}_{resGr}$ *is coNP–complete.*

iv. $\mathit{Ver}^{ie}_{resGr}$ *is D*^{P}*–complete.*

Proof. The membership part for i–iv follows by Theorem 21 and the fact that Ver_{resGr} can be decided in polynomial time [9] (we recall that coNP$^{\mathrm{P}}$ = coNP). (i) To prove the coNP hardness we reduce the UNSAT-problem to $\mathit{Ver}^{idl}_{resGr}$. To do so we construct the AF $\mathcal{G}_\varphi$ as follows: given a CNF formula $\varphi(X) = \bigwedge_{c \in C} c$ with each clause $c \in C$ a disjunction of literals from Z,

$$\begin{aligned} A &= \{\varphi, \bar{\varphi}\} \cup C \cup X \cup \bar{X} \\ R &= \{(c, \varphi) \mid c \in C\} \cup \{(x, \bar{x}), (\bar{x}, x) \mid x \in X\} \cup \{(\varphi, \bar{\varphi}), (\bar{\varphi}, \varphi)\} \cup \\ &\quad \{(x, c) \mid x \text{ occurs in } c\} \cup \{(\bar{x}, c) \mid \neg x \text{ occurs in } c\} \end{aligned}$$

See Figure 3.8 for an illustration of the construction on a concrete example formula φ. We claim that $\{\bar{\varphi}\}$ is an ideal extension w.r.t. $resGr$ semantics iff φ is unsatisfiable. Thus first let us assume that φ is satisfiable and thus there exists a truth assignment τ such that $\tau(\varphi) = true$. Now consider the full resolution β and the resolved AF (A, R_β) (where $R_\beta = R \setminus \beta$) with $(z_i, \bar{z}_i) \in R_\beta \Leftrightarrow \tau(z_i) = true$ and $(\Phi, \Psi) \in R_\beta$. The grounded extension of this resolution is given by the set $\{x \in X : \tau(x) = true\} \cup \{\bar{x} \in X : \tau(x) = false\} \cup \{\varphi\}$. Hence $\bar{\varphi}$ is not skeptically accepted and thus not ideal accepted.

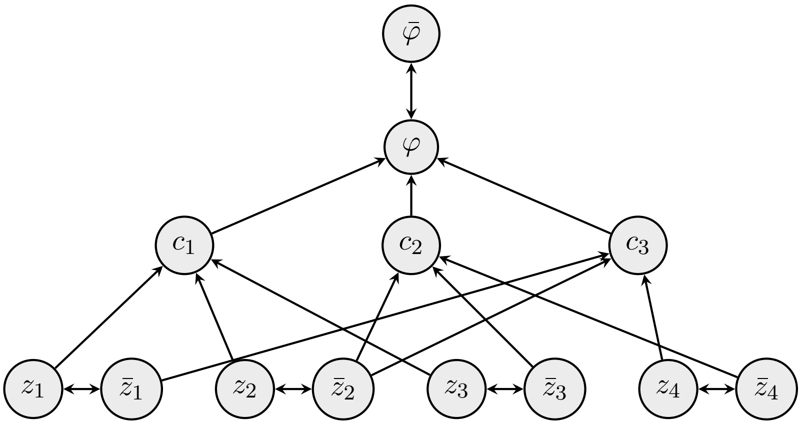


Figure 3.8: An example for the AF $\mathcal{G}_\Phi$ used in the proof of Theorem 24, using the formula $\varphi = C_1 \wedge C_2 \wedge C_3$ with $C_1 = \{z_1, z_2, z_3\}$, $C_2 = \{\bar{z}_2, \bar{z}_3, \bar{z}_4\}$, $C_3 = \{\bar{z}_1, z_2, z_4\}$.

Now let us assume that φ is unsatisfiable. As the set $\{\bar{\varphi}\}$ defends itself it suffices to show that $\{\bar{\varphi}\}$ is skeptical accepted, i.e. for every full resolution β the argument φ is in the grounded extension of the resolved AF (A, R_β). First if $(\bar{\varphi}, \varphi) \in R_\beta$ then $\bar{\varphi}$ is unattacked and thus clearly in the grounded extension. Now let us consider a full resolution β such that $(\varphi, \bar{\varphi}) \in R_\beta$ and towards a contradiction $\bar{\varphi} \notin grd(A, R_\beta)$. As β is a full resolution we have for each $x \in X$ that either $(x, \bar{x}) \in R_\beta$ or $(\bar{x}, x) \in R_\beta$ holds but not both of them. Thus each argument $x, \bar{x}$ is either in the grounded extension or attacked by the grounded extension. The arguments c_i are only attacked from arguments $x, \bar{x}$ and therefore each argument c_i is either part of the grounded extension or attacked by the grounded extension. Further if there is an c_i such that $c_i \in grd(A, R_\beta)$ then $grd(A, R_\beta)$ attacks φ and thus $\bar{\varphi} \in grd(A, R_\beta)$, a contradiction. If there is no such c_i we have that $grd(A, R_\beta)$ defends φ and thus $\varphi \in grd(A, R_\beta)$. But then the truth assignment τ defined by $\tau(x) = true \Leftrightarrow x \in grd(A, R_\beta)$ satisfies φ, which is in contradiction to the unsatisfiability of φ.

(ii) Immediate by the fact that $\bar{\varphi}$ is ideal accepted iff $\{\bar{\varphi}\}$ is an ideal set.

(iii) We have that for each argument x (resp. $\bar{x}$) there is a full resolution β such that $x \notin grd(A, R_\beta)$ (resp. $\bar{x} \notin grd(A, R_\beta)$) and thus none of them is skeptical accepted. Hence none of the arguments c_i as well as the argument φ can be in an ideal set. So we have that there is a non-empty ideal set iff $\bar{\varphi}$ is ideal accepted which is coNP–hard.

(iv) By the observations in (iii) we get that for the AF $\mathcal{G}_\varphi \,\dot{\cup}\, \mathcal{G}_\psi$ it holds that $\{\bar{\varphi}\}$ is the ideal extension E^{ie}_{resGr} iff the pair (φ, ψ) is a positive instance of the D^P-complete SAT-UNSAT problem. □

We continue with the naive semantics, where all problems remain tractable. We note that hardness for P only holds in case of frameworks where self-loops are allowed. If this is not the case, the ideal extension coincides with the set of skeptical accepted arguments (we recall that $Skept_{naive}$ is in L, see Theorem 11).

Theorem 25.

i. Ver^{idl}_{naive} *is in L.*

ii. $Ideal_{naive}$ *is P–complete.*

iii. $Exists^{\neg\emptyset}{}^{idl}_{naive}$ *is P–complete.*

iv. Ver^{ie}_{naive} *is P–complete.*

Proof. By Theorem 20 and the fact that $Ver_{naive} \in$ L we can compute the ideal extension in polynomial time. Hence the mentioned reasoning tasks can be clearly decided in P. Using the characterization from Proposition 12 we get an better upper bound for the problem Ver^{idl}_{naive}. To verify that a set $S \subseteq A$ is an ideal set, one has to check if $S \in adm(F)$ and further if for each argument $x \in S^{\ominus} \cup S^{\oplus}$ it holds that $(x, x) \in R$. Clearly both can be done in logarithmic space.

To show P-hardness we use a reduction from the P-hard problem to decide, given a propositional definite Horn theory T and an atom x, whether x is true in the minimal model of T. Let, for a definite Horn theory $T = \{r_l : b_{l,1} \wedge \cdots \wedge b_{l,m_l} \to h_l \mid 1 \leq l \leq n\}$ over atoms X and an atom $z \in X$, $(A_{T,z}, R_{T,z})$ be an AF with

$$\begin{aligned}
A_{T,z} &= X_1 \cup T_1 \cdots \cup X_{|T|} \cup T_{|T|} \cup X_{|T|+1} \cup X \cup \{t\} \\
R_{T,z} &= \{(x_i, x_i) \mid x_i \in X_1 \cup \cdots \cup X_{|T|+1}\} \cup \\
&\quad \{(x_i, r_{l,i}) \mid x_i \in X_i, r_{l,i} \in T_i, x \in \{b_{l,1}, \ldots, b_{l,m_l}\}\} \cup \\
&\quad \{(r_{l,i}, x_{i+1}) \mid x \in X_{i+1}, r_{l,i} \in T_i, x = h_l\} \cup \\
&\quad \{(x_{|T|+1}, x) \mid x_{|T|+1} \in X_{|T|+1}\} \cup \\
&\quad \{(t, r_{l,i}) \mid r_{l,i} \in T_1 \cup \cdots \cup T_{|T|}\} \cup \{(t,t), (z,t)\}
\end{aligned} \tag{3.3}$$

where t is a fresh argument (see Figure 3.9 for an example of the reduction).

Clearly the AF $(A_{T,z}, R_{T,z})$ can be constructed using only logarithmic space in the size of T. One can show that z is in the minimal model of T iff z is in the ideal extension of $(A_{T,z}, R_{T,z})$ iff the ideal extension of $(A_{T,z}, R_{T,z})$ contains at least one argument.

We start with proving the first equivalence. First we mention that for the set of skeptical accepted arguments A_{sa} it holds that $A_{sa} = T_1 \cup \cdots \cup T_{|T|} \cup X$. This is by the fact that these arguments are conflict-free and all the other arguments attack themselves. Notice that the argument z is in the minimal model of T iff there exists a sequence of Horn rules $r_{k_1}, \ldots, r_{k_n}$ such that $h_{k_n} = z$ and for $1 \leq i \leq n$ it holds that the rule body $\{b_{k_i,1}, \cdots, b_{k_i,m_{k_i}}\} \subseteq \{h_{k_1}, \ldots, h_{k_{i-1}}\}$. Given such a proof for z we can define the set $E = \{z\} \cup \{r_{k_i,j} \mid 1 \leq i \leq j \leq n\}$ By the construction of $(A_{T,z}, R_{T,z})$, in particular by the attacks defined in (3.3), we have that the set E is an admissible set and thus part of the ideal extension. Hence z is ideal accepted.

To show the "only if" part let E be the ideal extension and $z \in E$. To construct a proof for z we start with the following sequence $R_1 = \{r_l : r_{l,1} \in E\}, R_2 = \{r_l : r_{l,2} \in E\}, \ldots, R_n = \{r_l : r_{l,n} \in E \wedge h_l = z\}$. By the structure of $(A_{T,z}, R_{T,z})$ we have that the set $\{r_{l,i+1} \in E\}$ is defended by the set $\{r_{l,i} \in E\}$ (except against argument t which is attacked by z). But by construction we have that for every rule $(b_1 \wedge \cdots \wedge b_m \to h) \in R_{i+1}$ there exists rules

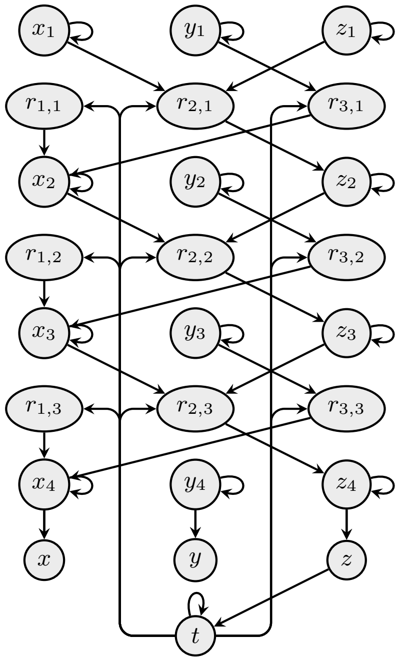


Figure 3.9: An illustrative example of the AF $(A_{T,z}, R_{T,z})$, as defined in the proof of Theorem 25, for the definite Horn theory $T = \{\rightarrow x, x \wedge y \rightarrow z, y \wedge z \rightarrow x\}$.

$r_{k_1}, \dots r_{k_m} \in R_i$ such that $h_{k_i} = b_i$ for$1 \leq i \leq m$. Thus we get a proof for z by using an arbitrary ordering $\prec$ over the horn rules and concatenating the sequences $R_1^{\prec}, \dots R_n^{\prec}$ ($R_i^{\prec}$ denotes the sequence corresponding to the set R_i ordered by $\prec$). Hence z is in the minimal model iff z is ideal accepted.

Next, as the argument t attacks all arguments $y \in A_{sa}$ and is only attacked by the argument z we have that the ideal extension is non-empty iff z is ideal accepted. Hence $\mathit{Exists}^{\neg\emptyset}{}^{idl}_{naive}$ is P–hard.

To show P-hardness of $\mathit{Ver}^{ie}_{naive}$, we consider the AF $(A_{T,z} \cup \{u\}, R_{T,z})$ where u is a fresh argument. Now we have that $\{u\}$ is the ideal extension iff the ideal extension of $(A_{T,z}, R_{T,z})$ is empty. Thus we have an reduction from co$\mathit{Exists}^{\neg\emptyset}{}^{idl}_{naive}$ to $\mathit{Ver}^{ie}_{naive}$ and thus $\mathit{Ver}^{ie}_{naive}$ is P-hard. □

We conclude our results with the stable semantics, where we in addition have to check whether there exists an extension of the base semantics – which adds another source of complexity.

Theorem 26.

i. Ver^{idl}_{stb} is D^P–complete.

ii. Ideal_{stb} is D^P–complete.

iii. $\mathit{Exists}^{\neg\emptyset}{}^{idl}_{stb}$ *is* D^P*–complete.*

iv. Ver^{ie}_{stb} *is* D^P*–complete.*

Proof. The memberships in D^P follows by Corollary 2 and the fact that $\mathit{Ver}_{stb} \in$ L.

For the hardness part we use the following two constructions: Given a propositional formula $\varphi = \bigwedge_{c \in C} c$ over variables X we define the AFs $F^1_\varphi = (A_1, R_1)$, $F^2_\varphi = (A_2, R_2)$ with:

$$\begin{aligned}
A_1 &= \{\varphi, \bar{\varphi}\} \cup C \cup X \cup \bar{X} \\
R_1 &= \{(c, \varphi) \mid c \in C\} \cup \{(x, \bar{x}), (\bar{x}, x) \mid x \in X\} \cup \{(\varphi, \bar{\varphi}), (\bar{\varphi}, \varphi)\} \cup \\
&\quad \{(l, c) \mid l \in c, c \in C\}
\end{aligned}$$

$$\begin{aligned}
A_2 &= C \cup X \cup \bar{X} \\
R_2 &= \{(x, \bar{x}), (\bar{x}, x) \mid x \in X\} \cup \{(c, c) \mid c \in C\} \\
&\quad \{(l, c) \mid l \in c, c \in C\}
\end{aligned}$$

Notice that the first one is a simplification of the construction presented in [46], showing hardness results for standard ideal semantics. It is easy to verify that this variation does not effect the proofs there and thus we can use that φ is unsatisfiable iff $\{\bar{\varphi}\}$ is an ideal set w.r.t. prf in F^1_φ iff $\{\bar{\varphi}\}$ is the ideal extension w.r.t. prf in F^1_φ. Now as the AF does not contain any odd length cycle it is coherent [42], and thus the results turn over to stable semantics.

The idea behind the construction of F^2_φ is that it is stable-consistent iff φ is satisfiable. We prove this as follows: First, let M be a model of φ then $E = M \cup \overline{X \setminus M}$ is a conflict free set in $F^2_\varphi = (A_2, R_2)$ and by construction of E each argument in $X \cup \bar{X}$ is either contained in E or attacked by E. Moreover as M is a model, for each clause $c \in C$ E contains at least one literal which is in the clause, hence each $c \in C$ is attacked. Thus E is a stable extension and $F^2_\varphi = (A_2, R_2)$ is stable-consistent. Now let us assume $F^2_\varphi = (A_2, R_2)$ is stable-consistent

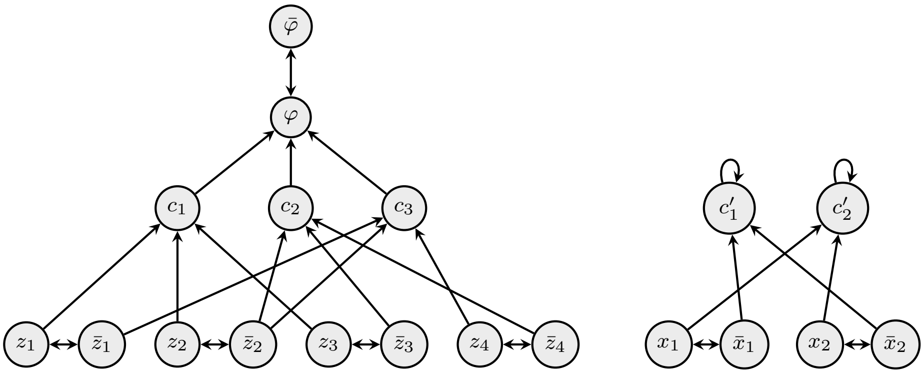


Figure 3.10: An illustration of the AF $F^1_{\{c_1,c_2,c_3\}} \dot{\cup} F^2_{\{c'_1,c'_2\}}$, as defined in proof of Theorem 26, with $c_1 = \{z_1, z_2, z_3\}$, $c_2 = \{\bar{z}_2, \bar{z}_3, \bar{z}_4\}$, $c_3 = \{\bar{z}_1, z_2, z_4\}$, $c'_1 = \{\bar{x}_1, \bar{x}_2\}$, $c'_2 = \{x_1, x_2\}$.

σ	Ver^{idl}_{σ}	$Ideal_{\sigma}$	$Exists^{\neg\emptyset}{}^{idl}_{\sigma}$	Ver^{ie}_{σ}
$comp$	P-c	P-c	in L	P-c
prf	coNP-c	in Θ_2^P	in Θ_2^P	in Θ_2^P
$naive$	in L	P-c	P-c	P-c
stb	D^P-c	D^P-c	D^P-c	D^P-c
$resGr$	coNP-c	coNP-c	coNP-c	D^P-c
sem	Π_2^P-c	Π_2^P-c	Π_2^P-c	D_2^P-c
stg	Π_2^P-c	Π_2^P-c	Π_2^P-c	D_2^P-c

Table 3.2: Complexity of ideal reasoning

and E being a stable extension as the arguments $c \in C$ are self-attacking none of them can be contained in E. But then E attacks all $c \in C$ and $E \cap X$ is a model of φ.

To show D^P - hardness we reduce an instance (φ, ψ) of the SAT-UNSAT to our reasoning problems

Ver^{idl}_{stb}: We build the AF $F = F^1_{\psi} \dot{\cup} F^2_{\varphi}$ (illustrated in Figure 3.10). We have that F is stable-consistent iff F^2_{φ} is stable consistent, and that $\{\bar{\psi}\}$ is an ideal set iff it is an ideal set in F^1_{ψ} and F is stable consistent. Hence $\{\psi\}$ is an ideal set in F iff φ is satisfiable and ψ is unsatisfiable.

$Ideal_{stb}$: Again use the AF $F = F^1_{\psi} \dot{\cup} F^2_{\varphi}$ and the observation that $\bar{\psi}$ is ideal accepted iff $\{\bar{\psi}\}$ is an ideal set.

$Exists^{\neg\emptyset}{}^{idl}_{stb}$: To this end we modify F^2_{φ} such that it always has the emptyset as ideal extension $F^3_{\varphi} := F^2_{\varphi} \cup (\{u, w, b\}, \{(u, w), (w, u), (u, b), (w, b), (b, b)\} \cup \{(b, x), (b, \bar{x}) \mid x \in X\})$. One can easily show that $stb(F^3_{\varphi}) = \{E \cup \{u\}, E \cup \{w\} \mid E \in stb(F^2_{\varphi})\}$. Hence F^3_{φ} is stable consistent iff F^2_{φ} is stable-consistent iff φ is satisfiable. Moreover each set $\emptyset \neq E \in adm(F^3_{\varphi})$ has to contain either u or w (otherwise none of the $c \in C$ is defended). If F^3_{φ} is stable consistent neither u nor w is skeptically accepted and thus the ideal extension is the empty set. However if F^3_{φ} is not stable consistent the ideal extension is empty by definition.

Now we consider the AF $F = F^1_{\psi} \dot{\cup} F^3_{\varphi}$. We have that $\{\bar{\psi}\}$ is an ideal extension iff it is an ideal set in F^1_{ψ} and F is stable consistent. Thus we obtain that $\{\bar{\psi}\}$ is an ideal extension in F iff φ is satisfiable and ψ is unsatisfiable.

Ver^{ie}_{stb}: We build the AF $F = F^1_{\varphi} \dot{\cup} F^1_{\psi}$. There are just four candidates for the ideal extension, i.e. $\emptyset$, $\{\bar{\varphi}\}$, $\{\bar{\psi}\}$ and $\{\bar{\varphi}, \bar{\psi}\}$. We have $\bar{\psi}$/$\bar{\varphi}$ is ideal accepted iff ψ/ψ is unsatisfiable. Hence $\{\bar{\psi}\}$ is an ideal extensions iff φ is satisfiable and ψ is unsatisfiable. □

This concludes our complexity analysis of ideal reasoning. Our results are collected in Table 3.2. There the lower part shows new results.

3.5 Summary

We briefly summarise the obtained results, together with the results from the literature discussed in Section 3.1, and draw the complexity landscape of abstract argumentation. In this chapter we complemented existing complexity results as follows

- We classified the "tractable" semantics, i.e. $cf, naive, grd$, w.r.t. *P-completeness*. That is, computing and even verifying the grounded extensions is P-complete and moreover the P-completeness of the verification problem turns over to resolution-based grounded semantics. When analysing ideal reasoning, we obtained that ideal reasoning w.r.t. naive semantics is also P-complete. Furthermore, we have showed that most of the tractable problems can be even carried out in L.

- We *complemented the complexity analysis of semi-stable semantics* answering two open questions raised by Dunne and Caminada [50]. That is, we provided Σ_2^P (resp. Π_2^P) lower complexity bounds for credulous (resp. skeptical) reasoning improved the existing $P_{||}^{NP}$-lower bounds [50].

- We provided a *comprehensive complexity analysis of stage semantics*, most importantly showing that both skeptical and credulous acceptance are on the second level of the polynomial hierarchy.

- Finally we studied the *complexity of ideal reasoning* with different base semantics. There we provided generic complexity results that allow to obtain both upper and lower complexity bounds for ideal reasoning given the complexity of the base-semantics. Moreover we gave a comprehensive analysis of ideal reasoning w.r.t. the semantics under our considerations.

The complexity classifications of the reasoning problems introduced in Section 2.1.3 are summarised in Table 3.3 (for the argumentation semantics under our considerations).

The first conclusion we can draw from the P-completeness results is that the corresponding problems are inherent sequential and thus are not (well) amenable for parallelisation (cf. [76]). Secondly, when comparing the "tractable" semantics $cf, naive, grd$ we have that credulous and skeptical reasoning in $cf, naive$ can be done in logarithmic space. So we have that grounded semantics offer somehow higher expressiveness than $cf, naive$. However, the picture is different when considering ideal reasoning and allowing self-attacks in the AFs, then naive and grounded semantics provide the same complexity.

Next let us consider the complexity results for semi-stable and stage semantics. We have that stage, semi-stable together with preferred-semantics form the argumentation semantics with reasoning problems on the second level of the polynomial hierarchy. However, stage and semi-stable semantics are the only semantics where credulous acceptance is on the second level, while for all the other semantics under our considerations credulous acceptance is at most NP-hard.

It is somehow surprising that both stage and semi-stable semantics are of the same complexity, as stage semantics are based on the computational highly tractable concept of conflict-freeness while semi-stable semantics are based on the computational hard concept of admissible sets. This suggests that maximising the range of an extension is harder than maximising

σ	Cred_σ	Skept_σ	Ideal_σ	Ver_σ	Exists_σ	$\mathit{Exists}_\sigma^{\neg\emptyset}$
cf	**in L**	trivial	trivial	**in L**	trivial	**in L**
naive	**in L**	**in L**	**P-c**	**in L**	trivial	**in L**
grd	**P-c**	**P-c**	**P-c**	**P-c**	trivial	**in L**
stb	NP-c	coNP-c	**$\mathbf{D}^P$-c**	**in L**	NP-c	NP-c
adm	NP-c	trivial	trivial	**in L**	trivial	NP-c
com	NP-c	**P-c**	**P-c**	**in L**	trivial	NP-c
resGr	NP-c	coNP-c	**coNP-c**	**P-c**	trivial	in P
prf	NP-c	Π_2^P-c	in Θ_2^P	coNP-c	trivial	NP-c
sem	**$\mathbf{\Sigma_2^P}$-c**	**$\mathbf{\Pi_2^P}$-c**	**$\mathbf{\Pi_2^P}$-c**	coNP-c	trivial	NP-c
stg	**$\mathbf{\Sigma_2^P}$-c**	**$\mathbf{\Pi_2^P}$-c**	**$\mathbf{\Pi_2^P}$-c**	**coNP-c**	trivial	**in L**

Table 3.3: Complexity of abstract argumentation ($\mathcal{C}$-c denotes completeness for class $\mathcal{C}$). Novel results are highlighted in boldface.

the extension itself. This is further mirrored by the following two facts. Firstly, starting from conflict-free sets we have that $\subseteq$-maximality, i.e. naive semantics, is still tractable while range-maximality leads to hardness for the second level of the polynomial hierarchy. Secondly, when considering credulous reasoning $\subseteq$-maximality can be neglected and thus does not add complexity while range-maximality has to be taken into account.

Considering the overall picture of ideal reasoning we have that ideal reasoning just adds polynomial run-time to skeptical acceptance algorithms. Moreover we have that ideal reasoning is as hard as skeptical reasoning, with two notable exceptions. First, when considering preferred as base-semantics we can use the complexity gap between credulous and skeptical acceptance and obtain a Θ_2^P algorithm for ideal reasoning, which is actually easier than the Π_2^P-hard skeptical acceptance. Second, when considering naive as base-semantics we have that the complexity of skeptical acceptance is actually to low to subsume the additional effort for ideal acceptance and thus the complexity slightly increases when switching to ideal reasoning. For stable semantics we have that ideal reasoning takes care whether there exists a stable extension or not. Thus the complexity of ideal reasoning is in accordance with the complexity of the skeptical reasoning version that also takes care of this, i.e. Skept', but increases the complexity of classical skeptical reasoning.

Finally, we identify the following two complexity issues which we have to leave open:

1. Identifying the exact complexity of $\mathit{Ideal}_{\mathit{prf}}$ (under standard reductions), i.e. deciding whether an argument is in the standard ideal extension.

2. Identifying the exact complexity of $\mathit{Exists}_{\mathit{resGr}}^{\neg\emptyset}$, i.e. deciding whether there exists a non-empty resolution-based grounded semantics.

As already mentioned for $Ideal_{prf}$, Dunne [46] has shown that it is coNP-hard (cf. Theorem 22) and can be decided within Θ_2^P (cf. Theorem 19). However we are not aware of any matching bounds. The work in [46] puts quite some effort in attempting lower bounds, proving that NP-hardness would suffices to obtain Θ_2^P-completeness. Moreover a hardness proof using randomized reductions is provided. Note that this complexity picture is somehow similar to that of deciding whether applying the closed world assumption (CWA) [66, 100] to a propositional theory is consistent. We have that a propositional theory is CWA-consistent if and only if the theory has a unique minimal model, i.e. the intersection of all models being a model. Although the latter characterisation provides some similarities to $Ideal_{prf}$ we did not manage to reduce one problem to the other. So when failing classifying whether $Ideal_{prf}$ is hard for Θ_2^P, another interesting open problem would be how $Ideal_{prf}$ relates to closed world reasoning.

The problem $Exists_{resGr}^{\neg\emptyset}$ was shown to be in the class P in [12] and to the best of our knowledge P-completeness for argumentation problems was not considered in the literature before[3]. However, the author expects the problem's complexity to lie somewhere between L and P, and thus not being accessible with complexity classes we consider in this work.

[3] Hence, there is just the author of this work to be to blame for this open issue.

CHAPTER 4

Towards Tractability

The complexity results from Chapter 3 show that most of the reasoning tasks in abstract argumentation are highly intractable, but on the other side the necessity of efficient algorithms is evident. So the best we can do is handling problem instances which are of lower complexity by appropriate efficient algorithms. This chapter is dedicated to identifying structural properties of argumentation frameworks that make reasoning tasks tractable, and trying to capture as most as possible (practical) instances by such characterisations. We firstly consider graph properties that allow for polynomial time algorithms, that is an AF being acyclic, free of even length cycles, bipartite or symmetric, Secondly we apply structural graph parameters such that the complexity of reasoning merely depends on this parameters but is polynomial in the size of the AF.

This chapter is organised as follows:

- Section 4.1 studies *Tractable Fragments*, i.e. graph classes on which argumentation problems are tractable which are intractable in the general case. Four classes of AFs are considered, namely acyclic AFs, AFs which are free of even-length cycles, bipartite AFs and symmetric AFs. We review and complement existing results from the literature and extend them to all of our semantics. Finally we consider fragments that do not yield tractability but allow to solve argumentation problems, which are in general hard for the second level of the polynomial hierarchy, within NP or coNP.

- Section 4.2 provides *Fixed-Parameter Tractability* results concerning the graph parameters tree-width and clique-width. To this end we first provide monadic second order logic encodings of the argumentation semantics and then apply the meta-theorems presented in Section 2.3.4.

- In Section 4.3 we prove *Fixed-Parameter Intractability* results for several parameters generalising the parameter tree-width for directed graphs. That is we prove that argumentation problems remains hard when the parameter cycle-rank is bounded and then use a meta-theorem from Section 2.3.4 to extend these results to the parameters directed path-width, Kelly-width, DAG-width, and directed tree-width.

- Finally in Section 4.4 we provide a *Summary* of our results and discuss them together with related work.

Parts of this chapter have been previously published. Sections 4.2 & 4.3 build on [57, 61], also incorporating the fixed-parameter tractability results from [54].

4.1 Tractable Fragments

Here we revisit known tractable fragments for abstract argumentation, i.e. acyclic AFs, AFs without even cycles, bipartite AFs and symmetric AFs. We extend existing results in two ways: Firstly, we extend known tractable fragments to all of the semantics under our consideration (whenever possible). Secondly, we give a more fine-grained complexity analysis than provided by the literature, i.e. we classify problems in such fragments w.r.t. P-completeness and L-membership.

4.1.1 Acyclic Argumentation Frameworks

To obtain exact complexity classifications for reasoning with AFs in the tractable fragments, we first present a result showing that deciding whether an argument is in the grounded extension is P-hard even for very simple structured AFs. To this end we present a reduction from the monotone circuit value problem[1].

Theorem 27. *$Cred_{grd}$ is P-complete even for acyclic bipartite AFs.*

Proof. The membership follows from the general case and we prove hardness by reduction from MCVP (see Definition 47). Consider a monotone circuit $(\beta_i)_{1 \leq i \leq m}$ over variables $X = \{x_1, \dots x_n\}$ and an assignment a. We construct the AF $F_\beta = (A, R)$ as follows:

$$\begin{aligned}
A &= \{\beta_i \mid 1 \leq i \leq m\} \cup \{\bar{\beta}_i^1, \bar{\beta}_i^2 \mid \beta_i = \wedge(j,k)\} \cup \{\bar{\beta}_i \mid \beta_i = \vee(j,k) \text{ or } \beta_i = x\} \cup \{a\} \\
R &= \{(a, \bar{\beta}_i) \mid \beta_i = x \text{ and } a(x) = true\} \cup \\
&\quad \{(\beta_j, \bar{\beta}_i^1), (\beta_k, \bar{\beta}_i^2), (\bar{\beta}_i^1, \beta_i), (\bar{\beta}_i^2, \beta_i) \mid \beta_i = \wedge(j,k)\} \cup \\
&\quad \{(\beta_j, \bar{\beta}_i), (\beta_k, \bar{\beta}_i), (\bar{\beta}_i, \beta_i) \mid \beta_i = \vee(j,k)\}
\end{aligned}$$

Clearly this reduction can be done with a constant number of cursors and thus in L. Further we have that each attack goes from a lower indexed argument to an higher indexed argument and thus the resulting AF is clearly acyclic. Moreover there are no attacks between arguments in the set $\{\beta_i \mid 1 \leq i \leq m\} \cup \{a\}$ as well as in the set $\{\bar{\beta}_i^1, \bar{\beta}_i^2 \mid \beta_i = \wedge(j,k)\} \cup \{\bar{\beta}_i \mid \beta_i = \vee(j,k) \text{ or } \beta_i = x\}$, which together build a partition F. Hence F is a acyclic bipartite AF.

For the correctness of the reduction we claim that $v(\beta_i, a) = true$ iff $\beta_i \in grd(F)$. The proof is by structural induction.

Let us first mention that by construction the argument a is not attacked in F, and hence $a \in grd(F)$. Towards the induction base we consider gates β_i that are variables. For such a gate

[1] Notice that, alternatively one can also use a slightly modified version of Translation 8 in Chapter 5, that deletes the original attacks, and translate each AF faithfully to an acyclic AF.

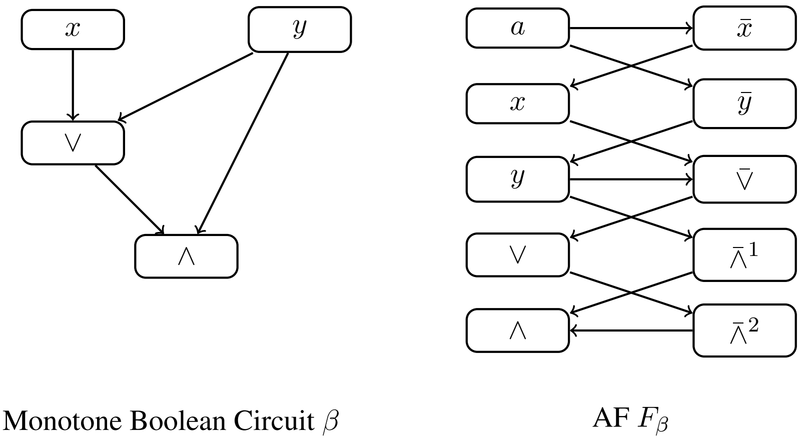


Figure 4.1: An illustrative example for the reduction in proof of Theorem 27, with the monotone circuit $\beta = (x, y, \vee(\beta_1, \beta_2), \wedge(\beta_2, \beta_3))$ and assignment $a(x) = a(y) = true$.

we have that $\bar{\beta}_i$ is the only argument attacking β_i. If $a(x) = true$ then a attacks $\bar{\beta}_i$ and β_i is defended. Thus $\beta_i \in grd(F)$. If $a(x) = false$ then $\bar{\beta}_i$ is not attacked and thus $\bar{\beta}_i \in grd(F)$. As $\bar{\beta}_i$ attacks β_i we have $\beta_i \notin grd(F)$. Hence $v(\beta_i, a) = true$ iff $\beta_i \in grd(F)$ for gates β_i that are variables. As β_1 must be a variable this gives us the induction base for $\vee, \wedge$ gates.

Now let us consider a gate $\beta_i = \wedge(j, k)$, with $j, k < i$. This argument β_i is attacked by the two arguments $\bar{\beta}_i^1$, $\bar{\beta}_i^2$, where the first is only attacked by β_j and the latter is only attacked by β_k. Thus we have that β_i is defended by $grd(F)$ iff $\beta_j \in grd(F)$ and $\beta_k \in grd(F)$. By the induction hypothesis this is equivalent to $v(\beta_j, a) = true$ and $v(\beta_k, a) = true$. By the definition of $v(.,.)$ we have that $v(\beta_i, a) = true$ iff $\beta_i \in grd(F)$.

Finally let us consider a gate $\beta_i = \vee(j, k)$, with $j, k < i$. The argument β_i is only attacked by the argument $\bar{\beta}_i$, which itself is attacked by β_j and β_k. Thus we have that β_i is defended by $grd(F)$ iff $\beta_j \in grd(F)$ or $\beta_k \in grd(F)$. By the induction hypothesis this is equivalent to $v(\beta_j, a) = true$ or $v(\beta_k, a) = true$. By the definition of $v(.,.)$ we have that $v(\beta_i, a) = true$ iff $\beta_i \in grd(F)$.

Finally we have $v(\beta, a) = v(\beta_m, a) = true$ iff $\beta_m \in grd(F)$. □

Corollary 3. *For $\sigma \in \{stb, com, resGr, prf, sem, stg\}$ the problems $Cred_\sigma$, $Skept_\sigma$, $Ideal_\sigma$ are P-complete when restricted to acyclic bipartite AFs.*

Proof. Immediate via Theorem 27 and the fact that on such AFs all the semantics σ coincide with grounded semantics. □

We are now prepared to give the complexity landscape on acyclic AFs.

Theorem 28. *For acyclic AFs the complexity results depicted in Table 4.1 holds.*

Proof. By Theorem 27, Corollary 3 and the identity $Cred_{adm} = Cred_{com}$. □

σ	grd	stb	adm	com	$resGr$	prf	sem	stg
$Cred_\sigma$	P-c	P-c	P-c	P-c	P-c	P-c	P-c	P-c
$Skept_\sigma$	P-c	P-c	trivial	P-c	P-c	P-c	P-c	P-c
$Ideal_\sigma$	P-c	P-c	trivial	P-c	P-c	P-c	P-c	P-c

Table 4.1: Complexity results for acyclic AFs ($\mathcal{C}$-c denotes completeness for class $\mathcal{C}$).

4.1.2 Even-Cycle free Argumentation Frameworks

The next tractable fragments is based on a observation by Dunne and Bench-Capon [47], that AFs without even-length cycles have a unique preferred extension, which can be computed in polynomial time. We first state this result for complete extensions.

Proposition 15. *Given an AF $F = (A, R)$ with $|com(F)| \geq 2$ then F contains an even length cycle.*

Proof. Suppose that $G, E \in com(F)$ where G is the grounded extension and E different from G. Thus we have $G \subseteq E$ and there exists $x_0 \in E \setminus G$. As $x_0 \notin G$ we have that x_0 is attacked by some y_0 such that $G \not\mapsto y_0$. Further as $x_0 \in E$ we have that $y_0 \notin E$. As E is admissible there exists an argument $x_1 \in E \setminus G$ attacking y_1. Now the same argument applies to the argument x_1 and we obtain that there exist $y_1 \notin E$ and $x_2 \in E \setminus G$. Inductively we obtain an infinite sequence $x_0, y_0, x_1, y_1, \dots, x_i, y_i, \dots$ such that $x_i \in E \setminus G$ and $(y_i, x_i) \in R$ as well as $(x_{i+1}, y_i) \in R$. As $E \setminus G$ is a finite set we certainly have that $x_i = x_j$ for some $i \neq j$. Then $x_i, y_i, x_{i+1}, y_{i+1}, \dots, y_{j-1}$ forms a cycle which is of even length. □

We have that in AFs without even-length cycles the grounded extension is the only complete extension. Now having Proposition 15 at hand we easily obtain the complexity results for admissibility based semantics.

Theorem 29. *On even-cycle free AFs the problems $Cred_\sigma$, $Skept_\sigma$, $Ideal_\sigma$ are P-complete for $\sigma \in \{grd, stb, com, resGr, prf, sem\}$. Moreover $Cred_{adm}$ is P-complete.*

Proof. The hardness follows immediately by the corresponding hardness results for acyclic AFs. It remains to prove the membership:

By Proposition 15 we have that each such AF F has only one complete extension which is also the grounded. Now as all semantics σ are com-preserving and provide at least one extension, except for stable semantics, we have that $grd, com, resGr, prf, sem$ semantics coincide and thus the lower complexity of grounded semantics carries over to the other ones. For stable semantics we have that there is just one candidate for being a stable extension, i.e. the grounded extension, which can be computed in polynomial time. We can check whether it is also stable in polynomial time and if so we have that stable and grounded semantics coincide. Otherwise there is no stable extension and we can immediately return the acceptance status of the argument.

Again we obtain the tractability for adm semantics via the identity $Cred_{adm} = Cred_{com}$. □

As stage semantics do not build on admissible sets the above observation does not lead to tractability. In fact we have that the absence of even cycles does not help when evaluating AFs w.r.t. stage semantics and all the reasoning tasks still maintain their full complexity.

Theorem 30. *The problem $Cred_{stg}$ is Σ_2^P-complete and the problems $Skept_{stg}$, $Ideal_{stg}$ are Π_2^P-complete even for AFs without even-cycles.*

Proof. The membership -part follows immediately from the general case. To show hardness we provide a reduction from the Σ_2^P-complete MINSAT problem (see Definition 55).

To this end let (φ, x_α) be an instance for the MINSAT problem, i.e. φ is a propositional formula over atoms X in CNF and $x_\alpha \in X$. We additionally assume an arbitrary order $<$ on the clauses of φ. Then we define the AF $F_{\varphi,x_\alpha} = (A, R)$ as follows:

$$\begin{array}{rcl} A & = & \{\varphi, b, q\} \cup C \cup X \cup \bar{X} \cup \{E_c \mid c \in C\} \\ R & = & \{(c, \varphi) \mid c \in C\} \cup \{(\varphi, b), (b, b), (q, x_\alpha)\} \cup \\ & & \{(\bar{x}, x) \mid x \in X\} \cup \\ & & \{(l, c) \mid l \in c, c \in C\} \cup \\ & & \{(E_c, a) \mid c \in C, a \in A \setminus (\{c, \varphi, b\} \cup \{E_{c'} : c' < c\})\} \end{array}$$

The AF F_{φ,x_α} is illustrated in Figure 4.2. We now claim that the following statements are equivalent:

1. The atom x_α is in a minimal model of φ.
2. The argument x_α is credulously accepted in F_{φ,x_α}.
3. The argument q is not skeptically accepted in F_{φ,x_α}.
4. The argument q is not ideally accepted in F_{φ,x_α}.

$(1) \Leftrightarrow (2)$ Recall that each stage extension is also a naive extensions, and hence we consider only naive extensions as candidates for stage extensions.

First let us consider naive extensions of $F_{\varphi,x_\alpha} = (A, R)$ containing an argument E_c. For simplicity we enumerate the clauses $c_1, \dots, c_m$ and the arguments $E_1, \dots, E_m$, according to the order $<$ on the clauses. Now one can easily check that these naive extensions are given by $\{\{E_i, \varphi, q\}, \{E_i, c_i, q\} \mid 1 \leq i \leq m\}$. Further we have that the arguments $E_1, \dots, E_m$ are in conflict with each other but not attacked from any other argument. Thus when concerning the $\leq_R^+$-maximality of the above naive extensions they only compete with each other but not with any other naive extension. Comparing the range of these extensions we get that stage extensions E such that for some i, $E_i \in E$ are the following $\{\{E_i, \varphi, q\} \mid 1 \leq i \leq n\} \cup \{\{E_1, c_1, q\}\}$.

Now let us consider naive sets E such that for each $1 \leq i \leq m$, $E_i \notin E$. As we already have stage extensions with $\{E_i, c_i, q\}^+ = A \setminus \{b\}$ and $\{E_i, \varphi, q\}^+ = A \setminus (\{c_i, E_1, \dots, E_{i-1}\})$. Clearly $\{E_1, \dots, E_m\} \cap E = \emptyset$ and thus the only way for E being $\leq_R^+$-maximal is that $\{b, c_1, \dots, c_m\} \subseteq E^+$. When $b \in E^+$ then we have that $\varphi \in E$ and hence for $1 \leq i \leq m$ $c_i \notin E$. That is that $\{b, c_1, \dots, c_m\} \subseteq E^+$ iff $\varphi \in E$ and $X \cap E$ is a model of φ. Hence

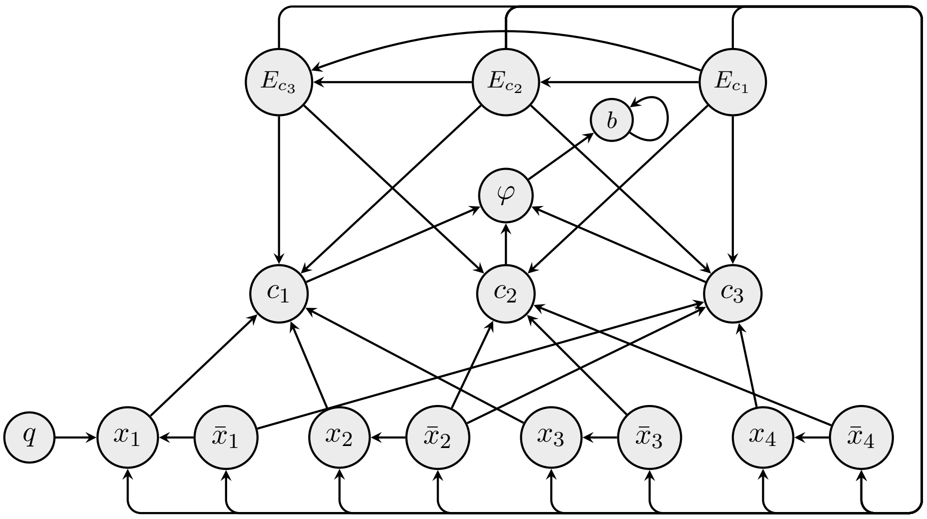


Figure 4.2: Illustration of the AF F_{Φ,x_1} as described in the proof of Theorem 30 for the CNF-formula $\varphi = \bigwedge_{c\in C} c$ with $C = \{\{y_1, y_2, z_3\}, \{\bar{y}_2, \bar{z}_3, \bar{z}_4)\}, \{\bar{y}_1, \bar{y}_2, z_4\}\}$.

there is a one-to-one correspondence between models M of φ and candidates for stage extensions $M^* := M \cup (\overline{X \setminus M}) \cup \{\varphi\} \cup \{q \mid \text{if } x_\alpha \notin M\}$. By the construction the range of each candidate is clearly incomparable with the ranges of the already determined stage extensions $\{\{E_i, \varphi, q\} \mid 1 \leq i \leq n\} \cup \{\{E_1, c_1, q\}\}$. and thus the $\leq^+_R$-maximality of such a candidate only depends on the other candidates.

It remains to show that for two models M, N, we have that $M \subseteq N$ iff $M^* \geq^+_R N^*$. For the "only if" direction consider $M \subseteq N$. We have that $(M^*)^+ = A \setminus (\bar{M} \cup \{E_1, \dots, E_m\}) \cup \{q \mid \text{if } x \notin M\}$ and $(N^*)^+ = A \setminus (\bar{N} \cup \{E_1, \dots, E_m\}) \cup \{q \mid \text{if } x \notin M\}$. As by assumption $M \subseteq N$ we finally have that $A \setminus (\bar{M} \cup \{E_1, \dots, E_m\}) \supseteq A \setminus (\bar{N} \cup \{E_1, \dots, E_m\})$.

For the "if" part let us consider $M \not\subseteq N$. Hence there is some $x \in M$ such that $x \notin N$. But then we have that $\bar{x} \notin (M^*)^+$ and $\bar{x} \in (N^*)^+$. That is that $M^* \not\geq^+_R N^*$.

(2) ⇔ (3): As x_α is the only argument which has a conflict with q we have that each naive extensions, and thus also each stage extension, either contains q or x_α. Hence if q is in all stage extensions then x_α is not credulously accepted and vice versa.

(3) ⇔ (4): As q is not attacked at all, it is ideally accepted iff it is skeptically accepted. □

We mention that the AF F_{Φ,x_α} in the above proof contains just one cycle, i.e. the self-attacking argument b. That is when starting from a acyclic AF which is easily solvable, just adding one attack can bring us back to the full complexity on the second-level of the polynomial hierarchy.

The complexity landscape of reasoning on even-cycle free AFs is given in Table 4.2.

σ	grd	stb	adm	com	$resGr$	prf	sem	stg
$Cred_\sigma$	P-c	P-c	P-c	P-c	P-c	P-c	P-c	Σ_2^P-c
$Skept_\sigma$	P-c	P-c	trivial	P-c	P-c	P-c	P-c	Π_2^P-c
$Ideal_\sigma$	P-c	P-c	trivial	P-c	P-c	P-c	P-c	Π_2^P-c

Table 4.2: Complexity results for even-cycle free AFs ($\mathcal{C}$-c denotes completeness for class $\mathcal{C}$).

4.1.3 Bipartite Argumentation Frameworks

The class of bipartite AFs was first discovered by Dunne [44], who showed that credulous and skeptical reasoning w.r.t. preferred or stable semantics can be done in P. Further he mentioned that these AFs are free of odd-length cycles and thus coherent [42]. This observation will be the key to extend these results to semi-stable and stage semantics.

Baroni et al. [12] showed that given a set of arguments deciding whether they are simultaneously accepted by one resolution-based grounded extension is NP-complete. The following proposition strengths this result as it shows that credulous reasoning is also NP-hard if we only consider a single argument for being accepted.

Proposition 16. *The problem $Cred_{resGr}$ is NP-complete even for bipartite AFs.*

Proof. The membership is immediate via the results for the general case in [12]. We prove hardness by a reduction from the Monotone SAT problem. Thus let $\varphi = \bigwedge_{c \in C} c$ be a monotone CNF over atoms X and (C_p, C_n) a partition of C in positive clauses C_p and negative clauses C_n. Then we define the following AF $F_\varphi = (A, R)$

$$\begin{array}{rcl} A & = & \{t\} \cup C \cup X \cup \bar{X} \cup \{a_c, b_c, d_c, e_c \mid c \in C_n\} \\ R & = & \{(c,t) \mid c \in C\} \cup \{(x,\bar{x}),(\bar{x},x) \mid x \in X\} \cup \\ & & \{(l,c) \mid \text{literal } l \text{ occurs in } c \in C_p \cup\} \\ & & \{(l,d_c) \mid \text{literal } l \text{ occurs in } c \in C_n\} \cup \\ & & \{(a_c,c),(a_c,b_c),(b_c,a_c),(d_c,b_c),(a_c,e_c),(e_c,d_c) \mid c \in C_n\} \end{array}$$

The reduction is illustrated in Figure 4.3. We can partition the arguments A in two independent sets, i.e. in the sets $X \cup \{a_c, d_c, t \mid c \in C_n\}$ and $\bar{X} \cup \{b_c, e_c, \mid c \in C_n\} \cup C$. Hence F_φ is a bipartite AF. We have to show that φ is satisfiable iff t is credulously accepted in F_φ.

We start with some observations on F_φ. When resolving a symmetric attack between an $x \in X$ and $\bar{x}$ we choose either x or $\bar{x}$ for being in the grounded extension of the resolved AF. Thus for two resolutions β, β' that such that $(x, \bar{x}) \in \beta$ and $(\bar{x}, x) \in \beta'$ we have that the corresponding grounded extensions of the resolved AFs are clearly not in $\subseteq$-relation. So to prove that the grounded extension of a resolved AF is also a resolution-based grounded extension of F_φ, we only have to consider resolutions which make the same choice on the arguments $X \cup \bar{X}$.

$\Rightarrow$: Given model $M \subseteq X$ satisfying φ. Let us consider the resolutions β such that $M \cup \overline{(A \setminus M)} \subseteq grd(A, R \setminus \beta)$. As M is a model we have that each argument $c \in C_p$ is attacked

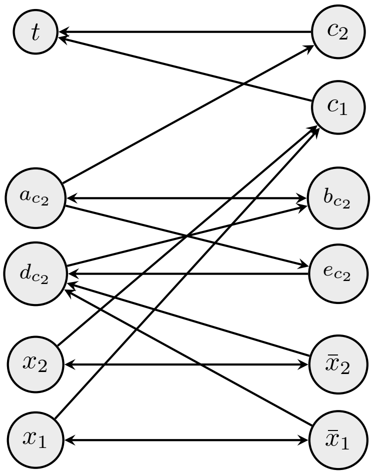


Figure 4.3: Illustration of the AF F_Φ as defined in the proof of Proposition 16, for the propositional formula $\varphi = c_1 \wedge c_2$ with $c_1 = x_1 \vee x_2$ and $c_2 = \neg x_1 \vee \neg x_2$.

by M and for each $c \in C_n$ the argument d_c is attacked by $\overline{A \setminus M}$. Now for each $c \in C_n$ we have to resolve the attacks $(a_c, b_c), (b_c, a_c)$, and dependent on the choice we either get $\{a_c\} \subseteq grd(A, R \setminus \beta)$ or $\{b_c, e_c, c\} \subset grd(A, R \setminus \beta)$. As these sets are not in $\subseteq$-relation, both give rise to different resolution-based grounded extensions of F_φ.

Now let us consider the resolution β such that for each $c \in C_n : a_c \in grd(A, R \setminus \beta)$ then the argument t is defended and hence $t \in grd(A, R \setminus \beta)$. Now as $grd(A, R \setminus \beta)$ is $\subseteq$-minimal and $t \in grd(A, R \setminus \beta)$ we have that t is credulously accepted.

$\Leftarrow$: To this end let us assume that φ is unsatisfiable. Towards a contradiction let us assume there is an $E \in resGr(F_\varphi)$ with $t \in E$. We consider the corresponding set of variables $M = E \cap X$. As M is not a model of φ we have that there exists a $c \in C$ such that M does not satisfy c. If $c \in C_p$ then by construction E does not attack c and hence t is not defended, a contradiction for E being a resolution-based grounded extension. Now let us consider the case where $c \in C_n$. Then d_c is not attacked by M and in order to build a $\subseteq$-minimal extension we have to put (a_c, b_c) in the resolution. But then none of the arguments a_c, b_c, d_c, e_c is contained in E and thus c is not attacked and t not defended, a contradiction. □

Next we give the complexity landscape for bipartite AFs.

Theorem 31. *For bipartite AFs the complexity results depicted in Table 4.3 holds.*

Proof. The P-hardness results follow immediately from Corollary 3.

The membership for preferred and stable semantics has been shown in Dunne [44], as well it is mentioned there that bipartite AFs are free of odd-length cycles and thus coherent (this dates back to [42]). Thus the results extend also to semi-stable and stage semantics, as the coincide with preferred semantics for this class of frameworks. For admissible and complete semantics we use the identities $Cred_{adm} = Cred_{com} = Cred_{prf}$. The case of resolution-based grounded

σ	grd	stb	adm	com	$resGr$	prf	sem	stg
$Cred_\sigma$	P-c	P-c	P-c	P-c	NP-c	P-c	P-c	P-c
$Skept_\sigma$	P-c	P-c	trivial	P-c	P-c	P-c	P-c	P-c
$Ideal_\sigma$	P-c	P-c	trivial	P-c	P-c	P-c	P-c	P-c

Table 4.3: Complexity results for bipartite AFs ($\mathcal{C}$-c denotes completeness for class $\mathcal{C}$).

semantics is covered by Proposition 16 and by results in [12], stating that in bipartite AFs the problem $Skept_{resGr}$ can be solved in polynomial time. □

4.1.4 Symmetric Argumentation Frameworks

The class of *Symmetric Argumentation Frameworks* was studied by Coste-Marquis et al. [31], but with the additional assumption that the attack relation is irreflexive, i.e. there are now self-attacking arguments. Here we extend the results presented there to semi-stable and stage semantics (the case of $resGr$ has been studied in [12]) and to symmetric AFs allowing self-attacks.

We first recall important facts about symmetric frameworks, starting with those that do not build on the irreflexivity of the attack relation. As each argument defends itself against all its attackers, the conflict-free sets coincide with the admissible sets and thus also naive and preferred extensions coincide [31]. Clearly this also applies to semi-stable and stage semantics, and hence semi-stable and stage extensions coincide on symmetric AFs.

Considering reasoning with an cf-preserving semantics that proposes at least one extension, we have that ideal and skeptical reasoning coincide[2]. This is because the set of skeptically accepted arguments is conflict-free and by the above observation also admissible and therefore already the ideal extension. Next we have that the grounded extension is the set of unattacked arguments [31]. Moreover symmetric AFs are coherent if the attack-relation is irreflexive [31].

We first summarise and complement complexity results for irreflexive symmetric AFs.

Theorem 32. *For irreflexive symmetric AFs the complexity results depicted in Table 4.4 hold.*

Proof. Such AFs are coherent and thus sem, stg, stb, prf, and $naive$ semantics coincide. Hence, the complexity of $naive$ semantics turns over to the other semantics. Furthermore, as ideal and skeptical reasoning coincides, also the lower complexity of skeptical reasoning turns over to ideal reasoning.

To test whether an argument is in the grounded extension we just have to check whether it is attacked or not – which is clearly in L. Now the results for admissible and complete semantics follow from the accordance of certain reasoning problems (see Proposition 5).

For $resGr$ Baroni et al. [12] showed the membership in the class P and moreover that the skeptically accepted arguments are those arguments which are not attacked at all, i.e. the

[2] In the case of stable semantics we have that $Skept'_{stb} = Ideal_{stb}$

σ	grd	stb	adm	com	$resGr$	prf	sem	stg
$Cred_\sigma$	in L	in L	in L	in L	in P	in L	in L	in L
$Skept_\sigma$	in L	in L	trivial	in L	in L	in L	in L	in L
$Ideal_\sigma$	in L	in L	trivial	in L	in L	in L	in L	in L

Table 4.4: Complexity results for irreflexive symmetric AFs.

grounded extension coincide with the arguments skeptically accepted w.r.t. $resGr$. Hence we obtain L membership for skeptical and ideal reasoning. □

We now turn to general symmetric AFs allowing self-attacking arguments. We still have that preferred and naive semantics coincides, and that the characterisation of the grounded extension holds. Moreover, the results for resolution-based grounded semantics in [12] do not make use of the irreflexiveness. Hence, it suffices to consider stable, semi-stable, and stage semantics.

We start with the case of semi-stable and stage semantics (which actually coincide on symmetric AFs).

Proposition 17. *$Cred_{sem}$, $Cred_{stg}$ are Σ_2^P-complete and $Skept_{sem}$,$Skept_{stg}$, $Ideal_{sem}$, $Ideal_{stg}$ are Π_2^P-complete even for symmetric AFs allowing self-attacks.*

Proof. The membership is immediate via the membership for the general case.

To show hardness we will reduce an arbitrary AF F to a symmetric AF F^* (with self-attacks) such that $stg(F) = stg(F^*) = sem(F^*)$[3].

Given an AF $F = (A_F, R_F)$ we define $F^* = (A^*, R^*)$ as follows

$$\begin{aligned} A^* &= A_F \cup A'_F \\ R^* &= R_F \cup \{(b,a),(a,b'),(b',a) \mid (a,b) \in R_F\} \\ &\quad \cup \{(a,b),(b,a) \mid a \in A_F, (b,b) \in R_F\} \\ &\quad \cup \{(a,a'),(a',a),(a',a') \mid a \in A_F\} \end{aligned}$$

The reduction is illustrated in Figure 4.4. We claim that $stg(F) = stg(F^*) = sem(F^*)$.

First, we mention that every stage extension of an AF F is also maximal (w.r.t. $\subseteq$) conflict-free in F. Let us now consider the case where $\emptyset \in stg(F)$. We then have that $stg(F) = \{\emptyset\}$ which is equivalent to, for each $a \in A_F$ also $(a,a) \in R_F$. Then by construction of F^* for each $a \in A^*$ also $(a,a) \in R^*$ and therefore $stg(F^*) = sem(F^*) = \{\emptyset\}$. Hence the lemma holds for such AFs, and for the remainder of the proof we can assume that $\emptyset \notin stg(F)$.

For (1)$stg(F) = stg(F^*)$(2), we again observe that a set E is conflict-free in F iff it is conflict-free in F^*. In the following we use $(E^+_{R_F})'$ as a short hand for $\{a' \in A' \mid a \in E^+_{R_F}\}$. Then we have that $(E^+_{R_F})' \subseteq E^+_{R^*}$, since for each attack $(a,b) \in R_F$, we have $(a,b') \in R^*$. Furthermore, for each maximal conflict-free set E in F (and thus in F^*), it holds that $A_F \subseteq E^+_{R^*}$.

[3] We will introduce this form of reductions as so called translations in Chapter 5.

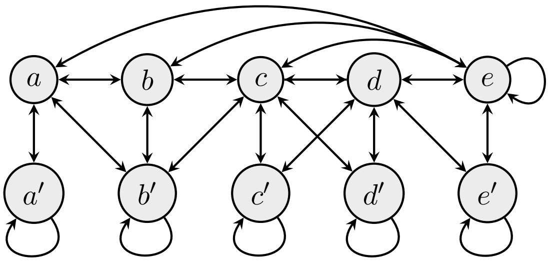


Figure 4.4: Illustration for the reduction in the proof of Theorem 17 for our running AF from Example 1.

We show this by contradiction. To this end, let us assume that $A_F \not\subseteq E^+_{R^*}$, i.e. there exists $a \in A_F$ such that $a \notin E^+_{R^*}$. As $E \neq \emptyset$ we have that all self-attacking arguments are contained in $E^+_{R^*}$, thus $(a, a) \notin R^*$. As $a \notin E^+_{R^*}$ we have that $E \not\rightarrowtail^R a$ and $a \not\rightarrowtail^R E$, but then the set $E \cup \{a\}$ is conflict-free in F and as E is maximal $a \in E$; a contradiction. Hence, for each maximal conflict-free set $E \subseteq A_F$ in F, i.e. the candidates for stage extensions, it holds that $E^+_{R^*} = A_F \cup (E^+_{R_F})'$ and thus $E^+_{R_F}$ is maximal (w.r.t. subset inclusion) iff $E^+_{R^*}$ is maximal.

For (2)$stg(F^*) = sem(F^*)$(3): For $E \in stg(F^*)$ iff $E \in sem(F^*)$, recall that on symmetric AFs stage and semi-stable semantics coincides.

As the reduction preserves the extensions clearly also the skeptical and credulous acceptance problems of F are reduced to corresponding problems in F^*. Finally to obtain the hardness result for ideal reasoning we recall that on symmetric AFs ideal and skeptical reasoning coincides. □

Next we give complexity results for stable semantics.

Proposition 18. *We have that $Cred_{stb}$ is NP-complete, $Skept_{stb}$ is coNP-complete, and $Ideal_{stb}$ is D^P-complete even for symmetric AFs allowing self-attacks.*

Proof. Again we can use the construction from the proof of Proposition 17 to build $F^* = (A^*, R^*)$ from a given F.

Given that $E^+_{R^*} = A_F \cup (E^+_{R_F})'$ (from the proof of Proposition 17) we immediately obtain that also $stb(F) = stb(F^*)$. Hence we have that the problems $Cred_{stb}$, $Skept_{stb}$, $Skept'_{stb}$ reduces to the versions restricted to symmetric AFs. Finally, we mention that, as admissible set and conflict-free sets coincide as well $Skept'_{stb}$ and $Ideal_{stb}$ coincide. □

The complexity results for symmetric AFs (allowing self-attacks) are summarised in Table 4.5. Notice that the complexities on general symmetric AFs and on irreflexive symmetric AFs only differ on stable, semi-stable and stage semantics.

4.1.5 Fragments beyond Tractability

So far we considered graph classes such that argumentation reasoning tasks are tractable, i.e. in the class P. But as the complexities of reasoning under preferred, semi-stable and stage seman-

σ	grd	stb	adm	com	$resGr$	prf	sem	stg
$Cred_\sigma$	in L	NP-c	in L	in L	in P	in L	Σ_2^P-c	Σ_2^P-c
$Skept_\sigma$	in L	coNP-c	trivial	in L	in L	in L	Π_2^P-c	Π_2^P-c
$Ideal_\sigma$	in L	D^P-c	trivial	in L	in L	in L	Π_2^P-c	Π_2^P-c

Table 4.5: Complexity results for symmetric AFs ($\mathcal{C}$-c denotes completeness for class $\mathcal{C}$).

tics are located at the second level of the polynomial hierarchy, one might be also interested in fragments reducing complexity from the second level to the first level of the polynomial hierarchy, i.e. fragments where reasoning can be done within NP or coNP.

We start with the class of coherent AFs, i.e. AFs where stable, preferred, semi-stable and stage semantics coincide.

Proposition 19. *When considering coherent AFs and semantics $\sigma \in \{prf, sem, stg\}$ we have that*

- $Cred_\sigma \in NP$
- $Skept_\sigma \in coNP$
- $Ideal_\sigma \in coNP$

Proof. We have that stable, preferred, semi-stable and stage semantics coincide. Thus we can use the NP (resp. coNP) procedures for stable semantics. □

Unfortunately it is in general Π_2^P-hard to decide whether an AF is coherent or not [48]. However one can consider subclasses that are easier to detect. One of that being AFs without odd-length cycles.

Corollary 4. *When considering AFs without odd-length cycles and semantics $\sigma \in \{prf, sem, stg\}$ we have that*

- $Cred_\sigma \in NP$
- $Skept_\sigma \in coNP$
- $Ideal_\sigma \in coNP$

Proof. Immediate by the fact that each odd-cycle free AF is coherent [42]. □

While the class of odd-cycle free AFs is a strict subset of the coherent AFs, it is easy to detect, i.e. deciding whether a given AF is free of odd-cycles is in P (see e.g. [4]).

Next we consider AFs guaranteeing the existence of a stable extension.

Proposition 20. *When considering stable-consistent AFs, we have that*

- $\mathit{Cred}_{sem} \in$ *NP*, $\mathit{Cred}_{stg} \in$ *NP*
- $\mathit{Skept}_{sem} \in$ *coNP*, $\mathit{Skept}_{stg} \in$ *coNP*
- $\mathit{Ideal}_{sem} \in$ *coNP*, $\mathit{Ideal}_{stg} \in$ *coNP*

Proof. We have that stable, semi-stable and stage semantics coincide. Thus we can use the NP (resp. coNP) procedures for stable semantics. □

We recall that deciding whether an AF is stable-consistent is NP-complete. We have that stable-consistent AFs are an NP-fragment of semi-stable and stage semantics. However one can show that reasoning under preferred semantics remains hard for stable-consistent AFs, by considering the hardness proof of Skept_{prf} [48] for QBFs $\forall Y \exists Z \varphi(Y, Z)$ such that φ is satisfiable.

Finally we have the class of AFs with cycle-rank 1, which reduces the complexity of skeptical reasoning under preferred semantics (cf. Theorems 35 & 36).

Proposition 21. $\mathit{Skept}_{prf} \in$ *coNP when restricted to AFs of cycle-rank* 1.

Proof. To prove $\mathit{Skept} \in$ coNP we provide a polynomial-time algorithm for verifying that a given set is a preferred extension. Then one can build a coNP-algorithm for Skept by deciding its complement by a standard guess and check approach. To verify whether a set E is a preferred extension of an AF $F = (A, R)$ we first compute the SCCs and build a linear order $S_1, \ldots S_m$ of the SCCs which respects the partial order given by the attacks between different components, i.e. for $i < j$ we have that $S_j \not\mapsto S_i$. Note that both the identification of SCCs and obtaining such a linear order can be done in polynomial time by depth-first search. Now one can decide the verification problem by considering each SCC separately starting with S_1 and then following the linear ordering. Therefore, we use a multi-labeling $\mathcal{M} : V_S \to 2^{\{in, def, undec\}}$ which maps vertices to sets of labels, as well as ordinary labelings $\mathcal{L} : V_S \to \{in, def, undec\}$ (see [27]). Intuitively such a labeling corresponds to an extension in the following way: an argument is labeled in if it is in the extension. An argument is labeled def if it is not in the extension and attacked by some argument in the extension. Intuitively, the label def indicates that the extension is “defended” against potential attacks from this argument. Finally, an argument is labeled $undec$ if it is neither in the extension nor attacked by an argument in the extension. Intuitively, the label $undec$ indicates that the status of this argument is in a sense “undecided” yet.

The multi-labeling will be used as a certain form of initialization of the currently considered SCC S_j (for $j > 1$ this might take results from SCCs S_i with $i < j$ into account); ordinary labelings are then obtained from $\mathcal{M}$ by taking a designated argument as a starting point and are finally compared to the candidate E.

The verification algorithm (see also Example 4 below for illustration) for a given AF $F = (A, R)$ with linearly ordered SCCs $S_1, \ldots, S_m$ and a set of arguments E is as follows and loops over j with $1 \leq j \leq m$.

1. First, initialize a multi-labeling $\mathcal{M}_j$ with $\mathcal{M}_j(a) = \{in, def, undec\}$, for all vertices a in S_j. For each attack (a, b) in F with $a \in S_i$, $b \in S_j$ and $i < j$, we set

$$\begin{array}{ll} \mathcal{M}_j(b) := \mathcal{M}_j(b) \setminus \{in, undec\} & \text{if } a \in E \\ \mathcal{M}_j(b) := \mathcal{M}_j(b) \setminus \{in\} & \text{if } a \notin E \wedge E \not\rightarrowtail a \end{array}$$

2. Identify an argument $x \in S_j$ such that $S_j \setminus \{x\}$ is acyclic.

3. Compute a labeling $\mathcal{L}_j^l$ for each label $l \in \mathcal{M}_j(x)$ as follows: $\mathcal{L}_j^l(x) = l$ and for all vertices $a \neq x$ in S_j:

$$\mathcal{L}_j^l(a) = \begin{cases} in & \text{if } in \in \mathcal{M}_j(a) \ \wedge \ \forall b \in S_j : b \rightarrowtail a \Rightarrow \mathcal{L}_j^l(b) = def \\ def & \text{if } \mathcal{M}_j(a) = \{def\} \text{ or } \exists b \in S_j : \mathcal{L}_j^l(b) = in \ \wedge \ b \rightarrowtail a \\ undec & \text{otherwise} \end{cases}$$

4. Verify the status of the selected argument x in labelings $\mathcal{L}_j^l$:

   - $\mathcal{L}_j^{in}$ is valid iff $\forall b \in S_j : b \rightarrowtail x \Rightarrow \mathcal{L}_j^l(b) = def$
   - $\mathcal{L}_j^{def}$ is valid iff $\mathcal{M}_j(x) = \{def\}$ or $\exists b \in S_j : \mathcal{L}_j^{def}(b) = in \ \wedge \ b \rightarrowtail x$
   - $\mathcal{L}_j^{undec}$ is valid anyway

   Let L_j be the set of valid labelings for S_j.

5. Define
$$L_j^* = \begin{cases} L_j & \text{if } L_j = \{\mathcal{L}_j^{undec}\} \\ L_j \setminus \{\mathcal{L}_j^{undec}\} & \text{otherwise} \end{cases}$$

6. Verification: Reject, if there is no $\mathcal{L} \in L_j^*$ such that for all vertices a in S_j it holds that $\mathcal{L}(a) = in$ iff $a \in E$; otherwise continue with the next SCC.

If the above algorithm terminates without rejecting E, then E is a preferred extension.

To show the correctness of the verification algorithm we exploit the following result from [8] (Prop. 41): For $E \subseteq A$ it holds that $E \in prf(F)$ iff for each SCC S of F it holds that $E \cap S$ is the $\subseteq$-maximal admissible set of $(S \setminus (E \setminus S)^{\oplus}, R \cap S \times S)$ satisfying that no argument is attacked by $A \setminus (S \cup E^{\oplus})$.

Our algorithm iterates over all SCCs S_i testing the above conditions for E being preferred. In (1) for each argument $b \in S \cap (E \setminus S)^{\oplus}$ we set $M_j(b) := \{def\}$ excluding argument b from having any effect in the subsequent computation, i.e. restricting the set S to $S \setminus (E \setminus S)^{\oplus}$. Moreover, for each argument b attacked by $A \setminus (S \cup E^{\oplus})$ we exclude in from the possible labels for b, implementing that b can not be contained in the admissible sets.

The steps (2) - (5) are used to compute these $\subseteq$-maximal admissible sets. In step (2) we use the fact that, by definition, $cr(F) \leq 1$ iff for each SCC $S = (A_S, R_S)$ of F, there is an argument $x \in A_S$, such that $S|_{A_S \setminus \{x\}}$ is acyclic. We note that this can be easily done in polynomial time. In steps (3) & (4) we compute all labelings that are (i) admissible and (ii) are candidates for being $\subseteq$-maximal. By definition of a labeling we have that an argument is labeled def iff it is attacked by an argument labeled in (the condition $\mathcal{M}_j(a) = \{\mathit{def}\}$ just ignores arguments in $S \cap (E \setminus S)^{\oplus}$). By (i) we have that if an argument is labeled in it has to be defended, i.e. all of its attackers have to be labeled def [27]. Further by (ii) we have that each argument defended by an extension and not attacked by $A \setminus (S \cup E^{\oplus})$, actually has to be in an extension. We obtain the conditions to compute $\mathcal{L}^l_j(a)$. Note that given the label of the selected argument x, we can compute the labels of all other arguments in S_j by a finite recursion (due to the fact that the SCC without x is acyclic).

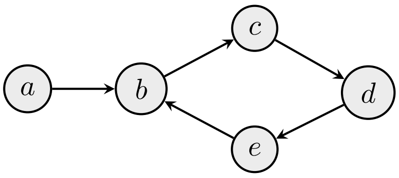
However, in step (4) we have to check whether the computed labels are compatible with the label of x, i.e. we have to verify if the computed label is really an admissible labeling. For the case where we labeled x with in, we have to check whether x is defended with respect to labeling $\mathcal{L}^{in}_j$. Similar, for $l = \mathit{def}$, we have to check whether x is attacked.

In step (5) we address the $\subseteq$-maximality of the valid labelings. The extensions corresponding to $\mathcal{L}^{in}_j$, $\mathcal{L}^{def}_j$ are clearly not in $\subseteq$-relation as one contains x and the other extension contains at least one attacker of x. On the other hand we have that in the recursion in step 3, switching an def or in label to undec never gives rise to a new in or def label. Hence, the extension corresponding to $\mathcal{L}^{undec}_j$ is in $\subseteq$-relation to the extensions corresponding to $\mathcal{L}^{in}_j$ and resp. $\mathcal{L}^{def}_j$ and as it neither contains the argument x nor an attacker of x they are also in $\subset$-relation. Hence if $\mathcal{L}^{in}_j$ and $\mathcal{L}^{def}_j$ are admissible labelings then $\mathcal{L}^{undec}_j$ is not maximal.

Finally in (6) we test whether $E \cap S$ corresponds to one of the maximal admissible sets. □

As we will see in Section 4.3, reasoning under semi-stable and stage remains hard for AFs of cycle-rank 1.

Example 4. *For illustration of the algorithm presented in the proof of Proposition 21, consider the AF* $F = (\{a, b, c, d, e\}, \{(a, b), (b, c), (c, d), (d, e), (e, b)\})$ *and the set* $E = \{a, c\}$.



We have two SCCs $S_1 = F|_{\{a\}}$ *and* $S_2 = F|_{\{b,c,d,e\}}$. *First we apply our algorithm to* S_1. *Since* S_1 *is an initial SCC, its multi-labeling is given by* $\mathcal{M}_1(a) = \{\mathit{in}, \mathit{def}, \mathit{undec}\}$. S_1 *has only one argument, we thus select* $x = a$ *in Step 2 and get the following three labelings* $\mathcal{L}^{in}_1(a) = \mathit{in}$, $\mathcal{L}^{def}_1(a) = \mathit{def}$ *and* $\mathcal{L}^{undec}_1(a) = \mathit{in}$ *in Step 3. As there is no argument attacking* a, $\mathcal{L}^{def}_1(a)$ *is not valid (Step 4). In Step 5, we obtain* $L^*_1 = \{\mathcal{L}^{in}_1, \mathcal{L}^{undec}_1\} \setminus \{\mathcal{L}^{undec}_1\} = \{\mathcal{L}^{in}_1\}$. *As* $a \in E$ *and* $\mathcal{L}^{in}_1(a) = \mathit{in}$, *we now have that* E *is valid on* S_1 *and we thus continue the algorithm with SCC* S_2.

For the multi-labeling $\mathcal{M}_2$ *we have that* $\mathcal{M}_2(c) = \mathcal{M}_2(d) = \mathcal{M}_2(e) = \{\mathit{in}, \mathit{def}, \mathit{undec}\}$ *and* $\mathcal{M}_2(b) = \{\mathit{def}\}$. *The latter equality holds because* $a \in E$ *and* $a \rightarrowtail b$. *In the next step we have four options for argument* x *to make* S_2 *acyclic. Let us consider* $x = d$. *We compute three* *labelings $\mathcal{L}_2^{in}$, $\mathcal{L}_2^{def}$ and $\mathcal{L}_2^{undec}$. They are given as follows:*

$$\begin{array}{llll}
\mathcal{L}_2^{in}(b) = def & \mathcal{L}_2^{in}(c) = in & \mathcal{L}_2^{in}(d) = in & \mathcal{L}_2^{in}(e) = def; \\
\mathcal{L}_2^{def}(b) = def & \mathcal{L}_2^{def}(c) = in & \mathcal{L}_2^{def}(d) = def & \mathcal{L}_2^{def}(e) = in; \\
\mathcal{L}_2^{undec}(b) = def & \mathcal{L}_2^{undec}(c) = in & \mathcal{L}_2^{undec}(d) = undec & \mathcal{L}_2^{undec}(e) = undec.
\end{array}$$

The labeling $\mathcal{L}_2^{in}$ is not valid, because of the fact that $c \rightarrowtail d$ and $\mathcal{L}_2^{in}(c) = in$. Hence we have that $L_2^* = \{\mathcal{L}_2^{def}\}$. Now, since $\mathcal{L}_2^{def}(e) = in$ but $e \notin E$, E is rejected by the algorithm.

It is easy to see that $\{a, c, e\}$ is the only set that would be accepted by the algorithm, which mirrors the fact that $\{a, c, e\}$ is the only preferred extension of F.

The problem $Skept_{prf}$ is closely related to the problem of deciding whether an AF is coherent [48], that is checking whether the stable and preferred extensions of the AF coincide. In general, deciding whether an AF is coherent is Π_2^P-complete, and the hardness proof is via the same reduction from [48] we will use to show that $Skept_{prf}$ is hard for AFs of cycle rank 2 (see Theorem 35 and Figure 4.5). Hence the problem clearly remains Π_2^P-hard for AFs of cycle-rank 2, but one might be interested whether this problems also becomes easier for cycle-rank 1.

In the proof of Proposition 21 we have shown that for AFs of cycle-rank 1 one can decide in polynomial time whether a given set is a preferred extension. This gives rise to a simple coNP algorithm for deciding coherence of an AF. First, non-deterministically guess a set E and then perform a polynomial-time test whether E is a preferred extension and whether E is a stable extension. If E is preferred and not stable one has found a counter-example for coherence. Moreover, coNP-hardness can be easily shown by using the reduction $F_\Phi^1 \cup (\{b\}, \{(\Psi, b), (b, b)\})$, where F_Φ^1 as in the proof of Theorem 26.

Finally, note that one can easily show NP (resp. coNP) hardness for all of the fragments presented in this section. These follow from the observation that the standard hardness proof for stable semantics is free of odd-length cycles (and thus coherent and stable-consistent) and Theorem 36 in Section 4.3.

4.2 Fixed Parameter Tractability

First investigations on fixed parameter tractability in the field of abstract argumentation where undertaken in [44], where several problems concerning preferred and stable semantics where shown to be fixed parameter tractable w.r.t. the graph-parameter tree-width of the AF. These results build on corresponding MSO encodings and Courcelle's meta-theorem (cf. Section 2.3.4).

Here we extend these results in three directions: First, we extend them to all semantics under our considerations. Second, we bring the parameter clique-width into play and extend the fixed parameter tractability to it. Third, we cover all computational tasks we are interested in.

4.2.1 MSO - Characterisations of Argumentation Semantics

In this section we use MSO to characterise all semantics under our considerations. We will exploit these characterisations later on to obtain fixed parameter tractability results via the meta-theorems presented in Section 2.3.4.

Building Blocks We first introduce some shorthands simplifying notation when dealing with subset relations and the range of extensions.

$$\begin{aligned}
x \notin X &= \neg(x \in X)\\
X \subseteq Y &= \forall x\,(x \in X \rightarrow x \in Y)\\
X \subset Y &= X \subseteq Y \wedge \neg(X \subseteq Y)\\
X \not\subseteq Y &= \neg(X \subseteq Y)\\
X \not\subset Y &= \neg(X \subset Y)\\
x \in X_R^+ &= x \in X \vee \exists y(y \in X \wedge (y,x) \in R)\\
X \subseteq_R^+ Y &= \forall x\,(x \in X_R^+ \rightarrow x \in Y_R^+)\\
X \subset_R^+ Y &= X \subseteq_R^+ Y \wedge \neg(X \subseteq_R^+ Y)
\end{aligned}$$

Another important notion that underlies argumentation semantics is the notion of a set being conflict-free. The following MSO formula encodes that a set X is conflict-free w.r.t. the attack relation R:

$$cf_R(X) = \forall x,y\,((x,y) \in R \rightarrow (\neg x \in X \vee \neg y \in X))$$

Next we give a building block for maximising extensions using an (MSO expressible) ordering $\sqsubseteq$ of sets:

$$max_{A,P(.),\sqsubseteq}(X) = P(X) \wedge \neg\exists Y \sqsubseteq A(P(Y) \wedge X \sqsubset Y)$$

Clearly we can also implement minimisation by inverting the ordering, i.e. $min_{A,P(.),\sqsubseteq}(X) = max_{A,P(.),\sqsupseteq}(X)$.

Standard Encodings In the following we provide MSO$_1$-characterisations for the different argumentation semantics. The characterisations for admissible, stable and preferred semantics are borrowed from [44].

$$\begin{aligned}
cf_R(X) &= \forall x,y\,(x,y) \in R \rightarrow (\neg x \in X \vee \neg y \in X)\\
naive_R(X) &= cf_R(X) \wedge \neg\exists Y\,(cf_R(Y) \wedge X \subset Y)\\
adm_R(X) &= cf_R(X) \wedge \forall x,y\,((x,y) \in R \wedge y \in X) \rightarrow \exists z(z \in X \wedge (z,x) \in R)\\
com_R(X) &= adm_R(X) \wedge \forall x(x \notin X \rightarrow \exists y((y,x) \in R \wedge \nexists z(z \in X \wedge (z,y) \in R)))\\
grd_R(X) &= com_R(X) \wedge \neg\exists Y\,(com_R(Y) \wedge Y \subset X)\\
stb_R(X) &= cf_R(X) \wedge \forall y(y \notin X \rightarrow \exists z \in X(z,x) \in R)\\
prf_R(X) &= adm_R(X) \wedge \neg\exists Y\,(adm_R(Y) \wedge X \subset Y)\\
sem_R(X) &= adm_R(X) \wedge \neg\exists Y\,(adm_R(Y) \wedge X \subset_R^+ Y)\\
stg_R(X) &= cf_R(X) \wedge \neg\exists Y\,(cf_R(Y) \wedge X \subset_R^+ Y)
\end{aligned}$$

As these encodings directly follow the definitions, they can be easily verified to be correct. Now, based on these encodings we build up encodings for the parameterized semantics. We start with encoding resolution-based semantics, and in a first step characterise the resolved attack relations $R \setminus \beta$ for the resolutions β.

$$\begin{aligned} \mathit{res}_R(X^E) = \forall x, y\big(X^E \subseteq R \wedge (x,x) \in R \to (x,x) \in X^E \wedge \\ (x \neq y \wedge (x,y) \in R) \to ((x,y) \in X^E \leftrightarrow (y,x) \notin X^E)\big) \end{aligned}$$

Now, following the definition, resolution based semantics are characterised by

$$\sigma^*_{A,R}(X) = \exists X^E \mathit{res}_R(X^E) \wedge \sigma_{A,X^E}(X) \wedge \forall Y \forall Y^E(\mathit{res}_R(Y^E) \wedge \sigma_{A,Y^E}(Y) \to Y \not\subset X)$$

However, as this encoding uses second order quantification over edges it does not yield an MSO$_1$-encoding for resGr. Hence, having in mind the meta-theorems for obtaining fixed parameter tractability results, we are interested whether it is possible to express resGr in MSO$_1$. We devote the next subsection to this problem.

Finally we consider parameterized ideal semantics. Here we first characterise ideal sets and then use them to encode the ideal extension.

$$\begin{aligned} \mathit{idealset}^\sigma_R(X) &= \mathit{adm}_R(X) \wedge \forall Y(\sigma_R(Y) \to X \subseteq Y) \\ \mathit{idealext}^\sigma_R(X) &= \mathit{idealset}^\sigma_R(X) \wedge \neg\exists Y \ (\mathit{idealset}^\sigma_R(Y) \wedge X \subset Y) \end{aligned}$$

Notice that the encoding for ideal sets and ideal extensions are in MSO$_1$ if the encoding of the corresponding base semantics is in MSO$_1$. So far all of the presented MSO encodings, except those for resolution-based semantics, omit quantification over edge sets and are thus MSO$_1$ encodings.

Towards an MSO$_1$ encoding of resolution-based grounded semantics

As the definition of resolution based semantics explicitly makes use of quantification over sets of attacks it is not amenable for a direct translation in MSO$_1$. Also all our attempts for implicitly quantifying over resolutions by using quantification over certain sets of arguments failed. Hence we go for a characterisation of resolution-based grounded semantics which eliminates the quantification over sets of attacks, exploiting results from [12].

To this end we first restrict the class of resolutions we have to consider when showing that a set of arguments is a complete extension of some resolved AF.

Lemma 9. *Given AF $F = (A, R)$ and a set $E \subseteq A$. If $E \in \mathit{resGr}(F)$ then there exists a resolution β with $\{(b,a) \mid a \in E, b \notin E, \{(a,b),(b,a)\} \subseteq R\} \subseteq \beta$ such that $E \in \mathit{com}(A, R \setminus \beta)$.*

Proof. As $E \in \mathit{resGr}(F)$ we have that there exists a resolution β' such that $E \in \mathit{grd}(A, R \setminus \beta')$. Now let us define β as $\{(b,a) \mid a \in E, \{(a,b),(b,a)\} \subseteq R\} \cup (\beta' \cap (A \setminus E \times A \setminus E))$. Clearly E is conflict-free in $(A, R \setminus \beta)$. Next we show that (i) $E^\oplus_{R \setminus \beta'} = E^\oplus_{R \setminus \beta}$ and (ii) $E^\ominus_{R \setminus \beta'} \supseteq E^\ominus_{R \setminus \beta}$.

For (i), let us first consider $b \in E^\oplus_{R \setminus \beta'}$. Then there exists $(a,b) \in R \setminus \beta'$ with $a \in E$ and by construction also $(a,b) \in R \setminus \beta$ and thus $b \in E^\oplus_{R \setminus \beta}$. Now let us consider $b \in E^\oplus_{R \setminus \beta}$. Then there exists $(a,b) \in R \setminus \beta$ with $a \in E$ and by construction either $(a,b) \in R \setminus \beta'$ or $(b,a) \in R \setminus \beta'$. In the first case clearly $b \in E^\oplus_{R \setminus \beta'}$. In the latter case b attacks E and as E is admissible in

$(A, R \setminus \beta')$ there exists $c \in E$ such that $(c, b) \in R \setminus \beta'$, hence $b \in E^{\oplus}_{R\setminus\beta'}$. For (ii) consider $b \in E^{\ominus}_{R\setminus\beta}$, i.e. exists $a \in E$ such that $(b, a) \in R \setminus \beta$. By the construction of β we have that $(a, b) \notin R$ and therefore $(b, a) \in R \setminus \beta'$. Hence also $b \in E^{\ominus}_{R\setminus\beta'}$.

As $E \in adm(A, R\setminus\beta')$ we have that $E^{\ominus}_{R\setminus\beta'} \subseteq E^{\oplus}_{R\setminus\beta'}$ and by the above observations then also $E^{\ominus}_{R\setminus\beta} \subseteq E^{\oplus}_{R\setminus\beta}$. Thus E is an admissible set. Finally let us consider an argument $a \in A \setminus E^{\oplus}_{R\setminus\beta}$. In the construction of β the incident attacks of a are not effected and hence $\{a\}^{\ominus}_{R\setminus\beta'} = \{a\}^{\ominus}_{R\setminus\beta}$. That is E defends a in $(A, R \setminus \beta)$ iff E defends a in $(A, R \setminus \beta')$. Now as $E \in com(A, R \setminus \beta')$ we have that a is not defended and hence $E \in com(A, R \setminus \beta)$. □

Now, using Lemma 9 we can give a first alternative characterisation for resolution-based grounded semantics.

Lemma 10. *Given AF $F = (A, R)$ and $E \subseteq A$. We have that $E \in resGr(F)$ iff the following conditions hold*

1. *there exists a resolution β with $\{(b, a) \mid a \in E, \{(a, b), (b, a)\} \subseteq R\} \subseteq \beta$ and $E \in com(A, R \setminus \beta)$*
2. *E is $\subseteq$-minimal w.r.t. (1).*

Proof. Let us first recall that by definition the grounded extension is the $\subseteq$-minimal complete extension and hence $resGr = com^*$.

$\Rightarrow$: Assume that $E \in resGr(F)$, then by Lemma 9 E fulfills condition (1). Further we have that each set E satisfying (1) is a complete extensions of a resolved AF. As by definition E is $\subseteq$-minimal in the set of all complete extensions of all resolved AFs it is also minimal for those satisfying (1).

$\Leftarrow$: As E satisfies (1) it is a complete extensions of a resolved AF. Now towards a contradiction let us assume it is not a resolution-based grounded extension. Then there exists $G \in resGr(F)$ with $G \subset E$. But by Lemma 9 G fulfills condition (1) and thus $G \subset E$ contradicts (2). □

In the next step we look for an easier characterisation of condition (1) from Lemma 10.

Lemma 11. *For an AF $F = (A, R)$ and $E \subseteq A$ the following statements are equivalent*

1. *There exists a resolution β with $\{(b, a) \mid a \in E, \{(a, b), (b, a)\} \subseteq R\} \subseteq \beta$ and $E \in com(A, R \setminus \beta)$*
2. *$E \in com(A, R \setminus \{(b, a) \mid a \in E, \{(a, b), (b, a)\} \subseteq R\})$ and $resGr(A \setminus E^{+}_{R}, R \cap ((A \setminus E^{+}_{R}) \times (A \setminus E^{+}_{R}))) = \{\emptyset\}$.*

Proof. In the following we will make use of the following shorthands, $R^* = R \setminus \{(b, a) \mid a \in E, \{(a, b), (b, a)\} \subseteq R\}$ and $(A', R') = (A \setminus E^{+}_{R}, R \cap ((A \setminus E^{+}_{R}) \times (A \setminus E^{+}_{R})))$.

(1) $\Rightarrow$ (2): Consider a resolution β such that $E \in com(A, R \setminus \beta)$. We first show that then also $E \in com(A, R^*)$. By construction we have that for arbitrary $b \in A$ that (a) $E \rightarrowtail^{R} b$ iff $E \rightarrowtail^{R\setminus\beta} b$ iff $E \rightarrowtail^{R^*} b$, and (b) $b \rightarrowtail^{R\setminus\beta} E$ iff $b \rightarrowtail^{R^*} E$. Hence we have that (i)

$E \in adm(A, R \setminus \beta)$ iff $E \in adm(A, R^*)$ and (ii) $E^+_R = E^+_{R\setminus\beta} = E^+_{R^*}$. By definition of complete semantics, $E \in com(A, R \setminus \beta)$ is equivalent to for each argument $b \in A \setminus E$ there exists an argument $c \in A$ such that $c \rightarrowtail^{R\setminus\beta} b$ and $E \not\rightarrowtail^{R\setminus\beta} c$. As $R^* \supseteq R \setminus \beta$ we obtain that $(c, b) \in R \setminus \beta$ implies $(c, b) \in R^*$. Using (a) we obtain that $E \in com(A, R \setminus \beta)$ implies for each argument $b \in A \setminus E$ there exists an argument $c \in A$ such that $(c, b) \in R^*$ and $E \not\rightarrowtail^{R^*} c$, i.e. $E \in com(A, R^*)$.

Now addressing $resGr(A', R') = \{\emptyset\}$ we again use the assumption $E \in com(A, R \setminus \beta)$, i.e. each argument which is defended by E is already contained in E, we have that $grd(A \setminus E^+_{R\setminus\beta}, R \setminus \beta \cap ((A \setminus E^+_R) \times (A \setminus E^+_R))) = grd(A', R' \setminus \beta) = \{\emptyset\}$. Note that $\beta' = \beta \cap R'$ is a resolution of (A', R') and that $grd(A', R' \setminus \beta) = grd(A', R' \setminus \beta') = \{\emptyset\}$. We can conclude that $resGr(A', R') = \{\emptyset\}$.

(1) $\Leftarrow$ (2): Let β' be a resolution such that $grd(A', R' \setminus \beta') = \{\emptyset\}$; such a β' exists since $resGr(A', R') = \{\emptyset\}$. Now consider the resolution $\beta = \{(b, a) \mid a \in E, \{(a, b), (b, a)\} \subseteq R\} \cup \beta'$. Again, by construction of β we have that for arbitrary $b \in A$: (a) $E \rightarrowtail^R b$ iff $E \rightarrowtail^{R\setminus\beta} b$ iff $E \rightarrowtail^{R^*} b$, and (b) $b \rightarrowtail^{R\setminus\beta} E$ iff $b \rightarrowtail^{R^*} E$. Hence we obtain that $E \in adm(A, R \setminus \beta)$. Using $R = E^+_{R\setminus\beta} = E^+_{R^*}$ we have $grd(A \setminus E^+_{R\setminus\beta}, (R \setminus \beta) \cap ((A \setminus E^+_R) \times (A \setminus E^+_R))) = grd(A', R' \setminus \beta') = \{\emptyset\}$. Thus, $E \in com(A, R \setminus \beta)$. □

Next we make use of a result by Baroni et al. [12].

Proposition 22. *[12] For an AF $F = (A, R)$, $resGr(F) = \{\emptyset\}$ iff for each minimal SCC S of F one of the following conditions holds:*

- *S contains a self-attacking argument*
- *S contains a non-symmetric attack*
- *S contains an undirected cycle*

Based on the above observations we obtain the following characterisation of resolution-based grounded semantics

Theorem 33. *Given AF $F = (A, R)$. The resolution-based grounded extensions are the $\subseteq$-minimal sets $E \subseteq A$ such that:*

- *$E \in com(A, R')$ with $R' = R \setminus \{(b, a) \mid a \in E, \{(a, b), (b, a)\} \subseteq R\})$*
- *Each minimal SCC S of $\hat{F} = (A \setminus E^+_R, R \cap A \setminus E^+_R \times A \setminus E^+_R)$ satisfies one of the following conditions:*
  - *S contains a self-attacking argument*
  - *S contains a non-symmetric attack*
  - *S contains an undirected cycle*

Proof. Immediate by Lemma 9, Lemma 10 and Proposition 22. □

In the following we encode the characterisation of resolution based grounded semantics obtained by Theorem 33 as MSO$_1$ formula. We start with encoding the relation R'.

$$R'_E(x,y) = (x,y) \in R \wedge \neg(x \in E \wedge y \notin E \wedge (x,y) \in R \wedge (y,x) \in R)$$

Now the AF $\hat{F} = (\hat{A}, \hat{R})$ is given by the following encodings.

$$\begin{aligned} \hat{A}_{A,R,E}(x) &= x \in A \wedge x \notin E \wedge \nexists y \in E : R'_E(y,x) \\ \hat{R}_{E,R}(x,y) &= (x,y) \in R \wedge A^*_{A,R,E}(x) \wedge A^*_{A,R,E}(y) \end{aligned}$$

To characterise reachability in a digraphs we borrow the following MSO$_1$ encoding from [34], encoding that there is path from x to y in R

$$\mathit{reachable}_R(x,y) = \forall X(x \in X \wedge [\forall u, v(u \in X \wedge R(u,v) \rightarrow v \in X)] \rightarrow y \in X)$$

Based on reachability we can easily specify a relation $SC_R(x,y)$ encoding whether the arguments x, y are strongly connected or not and a predicate $\mathit{minSCC}_{A,R}(x)$ that captures all arguments x in minimal SCCs.

$$\begin{aligned} SC_R(x,y) &= \mathit{reachable}_R(x,y) \wedge \mathit{reachable}_R(y,x) \\ \mathit{minSCC}_{A,R}(x) &= A(x) \wedge \neg\exists y\ (A(y) \wedge \mathit{reachable}_R(y,x) \wedge \neg\mathit{reachable}_R(x,y)) \end{aligned}$$

It remains to encode the check for each minimal SCC, which we implement by a check for each argument in a minimal SCC.

$$\begin{aligned} \mathit{C1}_R(x) &= \exists y(SC_R(x,y) \wedge (y,y) \in R) \\ \mathit{C2}_R(x) &= \exists y, z(SC_R(x,y) \wedge SC_R(x,z) \wedge (y,z) \in R \wedge (z,y) \notin R) \\ \mathit{C3}_R(x) &= \exists X(\exists y \in X \wedge \forall y \in X[SC_R(x,y) \wedge \\ &\qquad \exists u, v \in X : u \neq v \wedge (u,y) \in R \wedge (y,v) \in R]) \\ C_R(x) &= \mathit{C1}_R(x) \vee \mathit{C2}_R(x) \vee \mathit{C3}_R(x) \end{aligned}$$

Finally using Theorem 33 we obtain an MSO$_1$ encoding for resolution-based grounded semantics:

$$\begin{aligned} \mathit{cand}_{A,R}(X) &= \mathit{com}_{A,R'_X}(X) \wedge \forall x(\mathit{minSCC}_{\hat{A}_{A,R,E},\hat{R}_{E,R}}(x) \rightarrow C_{\hat{R}_{E,R}}(x)) \\ \mathit{resGr}_{A,R}(X) &= \mathit{cand}_{A,R}(X) \wedge \nexists Y(\mathit{cand}_{A,R}(Y) \wedge Y \subset X) \end{aligned}$$

4.2.2 Fixed-Parameter Tractability Results

Given the MSO$_1$ characterisations for all semantics under our considerations we are now prepared to state our fixed parameter tractability results. Given the meta-theorems from Section 2.3.4 we have that graph problems expressed in MSO$_1$ are fixed-parameter tractable w.r.t. the parameters tree-width and clique-width and graph problems expressed in MSO$_2$ are fixed-parameter tractable w.r.t. the tree-width. In the following we exploit these meta-theorems. To this end, let σ, σ' denote one of our semantics cf, naive, grd, adm, com, stb, $\mathit{resGr}\ \mathit{prf}$, sem, stg and resGr.

We first address the three reasoning problems and start with credulous acceptance.

Proposition 23. *Let σ be one of the above mentioned semantics.*

- Cred_σ *is fixed-parameter tractable w.r.t. the tree-width of F.*
- Cred_σ *is fixed-parameter tractable w.r.t. the clique-width of F.*

Proof. We use the following MSO encoding of credulous acceptance

$$\varphi^\sigma_{\mathit{Cred}}(x) = \exists X\ (x \in X \wedge \sigma_R(X))$$

which is MSO$_1$ if $\sigma_R(X)$ is in MSO$_1$. By the Theorems 8 and 9 we have that given an AF $F = (A, R)$ and an argument $a \in A$ one can decide whether $F \models \varphi^\sigma_{\mathit{Cred}}(a)$ is fixed-parameter tractable w.r.t. tree-width (resp. clique-width). □

Notice that the above theorem can be easily extended to the problem of deciding whether a set of arguments is simultaneously credulously accepted adapting the MSO formula as follows $\varphi^\sigma_{\mathit{Cred}}(X) = \exists Y\ (X \subseteq Y \wedge \sigma_R(Y))$. Next we consider skeptical acceptance.

Proposition 24. *Let σ be one of the above mentioned semantics.*

- Skept_σ *is fixed-parameter tractable w.r.t. the tree-width of F.*
- Skept_σ *is fixed-parameter tractable w.r.t. the clique-width of F.*

Proof. We use the following MSO encoding of skeptical acceptance

$$\varphi^\sigma_{\mathit{Skept}}(x) = \forall X\ (\sigma_R(X) \rightarrow x \in X)$$

which is MSO$_1$ if $\sigma_R(X)$ is in MSO$_1$. By the Theorems 8 and 9 we have that given an AF $F = (A, R)$ and an argument $a \in A$ one can decide whether $F \models \varphi^\sigma_{\mathit{Skept}}(a)$ is fixed-parameter tractable w.r.t. tree-width (resp. clique-width). □

Finally we consider ideal acceptance.

Proposition 25. *Let σ be one of the above mentioned semantics.*

- Ideal_σ *is fixed-parameter tractable w.r.t. the tree-width of F.*
- Ideal_σ *is fixed-parameter tractable w.r.t. the clique-width of F.*

Proof. We use the following MSO encoding of ideal acceptance

$$\varphi^\sigma_{\mathit{Ideal}}(x) = \exists X(\mathit{idealext}^\sigma_R(X) \wedge x \in X)$$

which is MSO$_1$ if $\sigma_R(X)$ is in MSO$_1$. By the Theorems 8 and 9 we have that given an AF $F = (A, R)$ and an argument $a \in A$ one can decide whether $F \models \varphi^\sigma_{\mathit{Ideal}}(a)$ is fixed-parameter tractable w.r.t. tree-width (resp. clique-width). □

We continue with the remaining decision problems.

Proposition 26. *Let σ be one of the above mentioned semantics.*

- *Ver_σ is fixed-parameter tractable w.r.t. the tree-width of F.*
- *Ver_σ is fixed-parameter tractable w.r.t. the clique-width of F.*

Proof. We use the following MSO encoding of the verification problem

$$\varphi^{\sigma}_{Ver}(X) = \sigma_R(X)$$

which is MSO$_1$ if $\sigma_R(X)$ is in MSO$_1$. By the Theorems 8 and 9 we have that given an AF $F = (A, R)$ and a set of arguments $S \in A$ one can decide whether $F \models \varphi^{\sigma}_{Ver}(S)$ is fixed-parameter tractable w.r.t. tree-width (resp. clique-width). □

Next we have that the problem $Exists_\sigma$ is trivial for all semantics except stable. Hence the next theorem is only stated for stable semantics.

Proposition 27. *For stable semantics we have that*

- *$Exists_{stb}$ is fixed-parameter tractable w.r.t. the tree-width of F, and*
- *$Exists_{stb}$ is fixed-parameter tractable w.r.t. the clique-width of F.*

Proof. We use the following MSO encoding of $Exists_\sigma$

$$\varphi^{\sigma}_{Exists} = \exists X \sigma_R(X)$$

which is MSO$_1$ if $\sigma_R(X)$ is in MSO$_1$. By the Theorems 8 and 9 we have that given an AF $F = (A, R)$ one can decide whether $F \models \varphi^{stb}_{Exists}$ is fixed-parameter tractable w.r.t. tree-width (resp. clique-width). □

Proposition 28. *Let σ be one of the above mentioned semantics.*

- *$Exists^{\neg\emptyset}_\sigma$ is fixed-parameter tractable w.r.t. the tree-width of F.*
- *$Exists^{\neg\emptyset}_\sigma$ is fixed-parameter tractable w.r.t. the clique-width of F.*

Proof. We use the following MSO encoding of $Exists_\sigma$

$$\varphi^{\sigma}_{Exists^{\neg\emptyset}} = \exists X \exists x (\sigma_R(X) \wedge x \in X)$$

which is MSO$_1$ if $\sigma_R(X)$ is in MSO$_1$. By the Theorems 8 and 9 we have that given an AF $F = (A, R)$ one can decide whether $F \models \varphi^{\sigma}_{Exists^{\neg\emptyset}}$ is fixed-parameter tractable w.r.t. tree-width (resp. clique-width). □

Finally we present an result illustrating that MSO is not only suitable for expressing reasoning problems with one semantics but also for comparing different semantics or reasoning with different semantics in parallel.

Proposition 29. *Given an AF $F = (A, R)$ and two semantics $\sigma, \sigma' \in \{cf, naive, grd, adm, com, stb, prf, sem, stg, resGr\}$.*

- *Deciding whether $\sigma(F)$=$\sigma'(F)$ is fixed-parameter tractable w.r.t. the tree-width of F.*
- *Deciding whether $\sigma(F)$=$\sigma'(F)$ is fixed-parameter tractable w.r.t. the clique-width of F.*

Proof. We use the following MSO encoding

$$\forall X(\sigma_R(X) \leftrightarrow \sigma'_R(X))$$

which is MSO_1 if both $\sigma_R(X)$ and $\sigma'_R(X)$ are in MSO_1. □

The above proposition generalises Dunne's observation that it can be efficiently decided whether an AF of bounded tree-width is coherent [44] (which is in general Π_2^P-complete [48]).

Clearly the results in this section can be generalised to all MSO_2 definable semantics when concerning tree-width and all MSO_1 definable semantics when concerning clique-width. In this work we restrict ourselves to decision problems, however as mentioned in section 2.3.4 the meta-theorems can be extended for enumerating models or counting models. Thus one can use MSO -characterisations also to obtain fixed parameter tractability results for enumerating or counting extensions [4].

As one can see, the presented meta-theorems are a powerful and elegant tool to classify argumentation problems to be fixed-parameter tractable. But when turning to algorithmic issues these theorems do not help, with the words of Niedermeier [93], MSO "*is a very elegant and powerful tool for quickly deciding about fixed-parameter tractability, but it is far from any efficient implementations*"[5]. Thus after classifying such problems to be fixed-parameter tractable the natural next step is to go for efficient algorithms, using for instance dynamic-programming techniques. For abstract argumentation prototypical dynamic programming algorithms for admissible and preferred semantics have been presented for AFs of bounded tree-width [57] and AFs of bounded clique-width [58], the former are implemented in the *dynPARTIX* system [60].

4.3 Fixed-Parameter Intractability

In this section we provide negative results for certain parametrisations of argumentation frameworks. In Section 4.2 we have shown that the graph parameter tree-width applies well to our reasoning problems. However tree-width does not take the direction of the attacks into account. As argumentation frameworks are directed graphs it seems promising to consider directed graph measures to get larger classes of tractable AFs than those captured by bounded tree-width. We have already seen the that the directed notion of clique-width applies well to argumentation. However the definition of clique-width is in a quite different manner that that of tree-width. The first one builds on algebraic expressions while the latter one uses structural decompositions of the graph. Hence one might be interest whether one can use dedicated decompositions for directed graphs to get further tractability results. As this is a frequent problem when dealing with

[4] The general complexity of counting extensions in abstract argumentation has been studied in [9].

[5] However, recently first competitive MSO model checking tools were announced [85, 88].

directed graphs there are several approaches generalising tree-width to directed graphs. Unfortunately, it turns out that the problems under our considerations remain hard when bounding typical directed graph measures. We illustrate this fact by first using cycle rank [63] as a parameter, and then we apply the meta-theorems from Section 2.3.4 to generalise these results to the other parameters.

In the following we show that for all semantics under our considerations the computational hard reasoning tasks maintain their full complexity when restricted to AFs with cycle-rank (see Definition 74) bounded by either 1 or 2. We start with semi-stable and stage semantics where cycle-rank 1 suffices for full hardness.

Theorem 34. *When restricted to AFs which have a cycle-rank of* 1 *the problems*

1. $Cred_{sem}$, $Cred_{stg}$ *remain* Σ_2^P*-hard,*
2. $Skept_{sem}$, $Skept_{stg}$ *remain* Π_2^P*-hard,*
3. $Ideal_{sem}$, $Ideal_{stg}$ *remain* Π_2^P*-hard.*

Proof. For (1) and (2) consider Reduction 1 in Section 3.3, it is easy to see that every framework of the form F_Φ has cycle-rank 1 and therefore we have an reduction from QBF formulas to an AF with cycle-rank 1. In fact, the strongly connected components of F_Φ are the following: $\{y_i, \bar{y_i}\}, \{z_i, \bar{z_i}\}$, $\{\varphi, \bar{\varphi}\}, \{y_i'\}$, $\{\bar{y}_i'\}, \{c_i\}, \{b\}$. As each of these components can be made acyclic by removing one vertex, the cycle-rank of F_Φ is thus 1.

For (3) consider the modification of F_Φ in the proof of Theorem 23, i.e. adding a new argument y mutual attacking $\bar{\varphi}$. As the new SCC $\{\varphi, \bar{\varphi}, y\}$ can be made acyclic by removing $\bar{\varphi}$ the modification still has cycle-rank 1 and the same argument as above applies. □

For preferred semantics we have that for AFs of cycle-rank 1 skeptical reasoning actually falls down to the first level of the polynomial hierarchy (see Proposition 21). However cycle-rank 2 suffices to obtain the full complexity.

Theorem 35. *The problem* $Skept_{prf}$ *remains* Π_2^P*-hard, even when restricted to AFs which have a cycle-rank of* 2*.*

Proof. Consider the following reduction from [48] mapping the Π_2^P-hard problem of deciding whether a given QBF $\Phi = \forall Y \exists Z \varphi(Y, Z)$ is valid, where $\varphi = \bigwedge_{c \in C} c$ and $X = Y \cup Z$, to $Skept_{prf}$. Given Φ one constructs the AF $F_\Phi = (A_\Phi, R_\Phi)$ as follows (Figure 4.5 illustrates the construction):

$$\begin{aligned} A_\Phi &= \{\varphi, b\} \cup C \cup X \cup \bar{X} \\ R_\Phi &= \{(c, \varphi) \mid c \in C\} \cup \{(x, \bar{x}), (\bar{x}, x) \mid x \in X\} \cup \\ &\quad \{(x, c) \mid x \text{ occurs in } c\} \cup \{(\bar{x}, c) \mid \neg x \text{ occurs in } c\} \cup \\ &\quad \{(\varphi, b), (b, b)\} \cup \{(b, z), (b, \bar{z}) \mid z \in Z\} \end{aligned}$$

We have that each model Φ is valid iff φ is skeptically accepted in F_Φ [48]. Moreover, we have that F_Φ has cycle-rank 2. Deleting argument b from F_Φ would result in an AF with the following

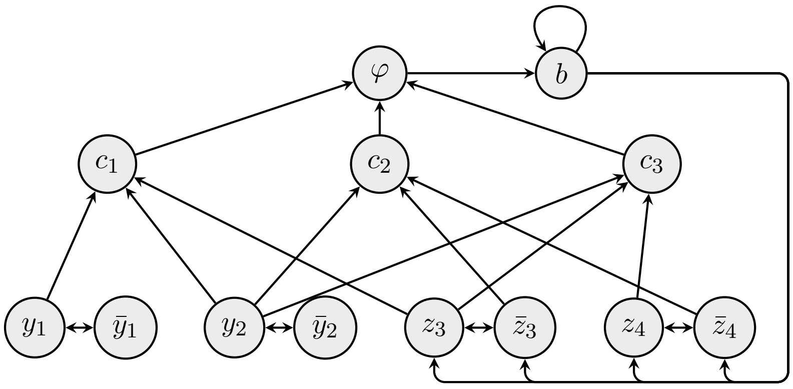


Figure 4.5: F_Φ for the QBF $\Phi = \forall y_1 y_2 \exists z_3 z_4 (y_1 \vee y_2 \vee z_3) \wedge (y_2 \vee \bar{z}_3 \vee \bar{z}_4) \wedge (y_2 \vee z_3 \vee z_4)$.

SCCs $\{y_i, \bar{y}_i\}, \{z_i, \bar{z}_i\}, \{c_i\}, \{\varphi\}$. As these SCCs can be clearly made acyclic by deleting one argument we have that F_Φ has cycle-rank ≤ 2 □

For hardness results on the first level of the polynomial hierarchy cycle-rank 1 is sufficient.

Theorem 36. *When restricted to AFs which have a cycle-rank of* 1 *the problem*

- *$Cred_{prf}$ remains NP-hard,*
- *$Skept_{prf}$ remains coNP-hard,*
- *$Ideal_{prf}$ remains coNP-hard.*

Proof. Consider the AF F^1_φ from the proof of Theorem 26. We have that the argument $\bar{\varphi}$ is skeptically / ideal accepted in F^1_φ iff the formula φ is unsatisfiable [41, 46]. The SCCs of F^1_φ are $\{x_i, \bar{x}_i\}, \{c_i\}, \{\varphi, \bar{\varphi}\}$ and hence F^1_φ has cycle-rank 1. □

We continue with stable semantics.

Theorem 37. *When restricted to AFs which have a cycle-rank of* 1 *the problem*

1. *$Cred_{stb}$ remains NP-hard,*
2. *$Skept_{stb}$ remains coNP-hard,*
3. *$Ideal_{stb}$ remains D^P-hard.*

Proof. For (1) and (2) again consider the AF F^1_φ from the proof of Theorem 26. We mention that F^1_φ has no odd length cycle and therefore is a coherent AF [42]. Thus the results from [41, 46] carry over and we have that (i) the argument $\bar{\varphi}$ is skeptically accepted w.r.t. stb in F^1_φ iff the formula φ is unsatisfiable and (ii) the argument φ is credulously accepted w.r.t. stb in F^1_φ iff the formula φ is satisfiable. As F^1_φ has cycle-rank 1 the statements (1) and (2) follow.

For (3) consider the AF $F = F^1_\psi \dot\cup F^2_\varphi$ from the proof of Theorem 26 showing D^P-hardness for $\mathit{Ideal}_{\mathit{stb}}$. It remains to show that F has cycle-rank 1. As, by definition the AFs F^1_ψ and F^2_φ are disjoint parts of F the cycle-rank of F is simple the maximum of the cycle-ranks of F^1_ψ and F^2_φ. We already obtained that the cycle rank of F^1_ψ is 1. Thus let us consider F^2_φ. The SCCs of F^2_φ are $\{x_i, \bar{x_i}\}, \{c_i\}$ and each of them can be made acyclic by deleting one argument, thus F^2_φ has cycle rank 1. Hence, $F = F^1_\psi \dot\cup F^2_\varphi$ has cycle rank 1. □

We complete our analysis of the parameter cycle-rank with resolution-based grounded semantics.

Theorem 38. *When restricted to AFs which have a cycle-rank of* 1 *the problem*

1. $\mathit{Cred}_{\mathit{resGr}}$ *remains NP-hard,*
2. $\mathit{Skept}_{\mathit{resGr}}$ *remains coNP-hard,*
3. $\mathit{Ideal}_{\mathit{resGr}}$ *remains coNP-hard.*

Proof. Recall the AF $\mathcal{G}_\varphi$ from the proof of Theorem 24. By Theorem 24 and a result from [12] we have that a formula φ is satisfiable iff the argument φ is credulously accepted in $\mathcal{G}_\varphi$ iff the argument $\bar{\varphi}$ is not skeptically accepted in $\mathcal{G}_\varphi$ iff the argument $\bar{\varphi}$ is not ideal accepted in $\mathcal{G}_\varphi$. Moreover the SCCs of $\mathcal{G}_\varphi$ are $\{x_i, \bar{x_i}\}, \{c_i\}, \{\varphi, \bar{\varphi}\}$ and hence $\mathcal{G}_\varphi$ has cycle-rank 1. □

By Theorem 10 we have that if a problem remains hard for AFs of bounded cycle-rank it is also hard when bounding one of the other directed graph measures, i.e. directed path-width, Kelly-width, DAG-width and directed tree-width. Thus, by the above hardness results for AFs of bounded cycle-rank, we have that none of these graph parameters is applicable for efficient reasoning in abstract argumentation.

Theorem 39. *For all semantics σ under our considerations. The problems Cred_σ, Skept_σ, Ideal_σ maintain there full complexity even when restricted to AFs of bounded cycle-rank, directed path-width, Kelly-width, DAG-width or directed tree-width.*

4.4 Summary

Let us first briefly resume the results of this chapter, before discussing the overall picture of tractability in abstract argumentation. In this chapter we

- complemented studies on the tractable fragments of acyclic, even-length cycle free, bipartite and symmetric AFs. That is we extend them to all of our semantics and classified tractable problems w.r.t. P-completeness.
- extended fixed parameter tractability results for the parameter tree-width.
- presented new fixed parameter tractability results for the graph parameter clique-width.

- showed that several generalisations of tree-width to directed graphs are not applicable to obtain fixed parameter tractability results for reasoning tasks in argumentation.

So for abstract argumentation we have four (known) tractable fragments, namely acyclic AFs, even-length cycle free AFs, bipartite AFs and symmetric AFs. While reasoning on acyclic AFs is tractable for each of our semantics the other fragments have some restrictions. So for even-length cycle free AFs we have that reasoning with stage semantics maintains its full complexity. For bipartite AFs credulous reasoning with resolution based grounded semantics is NP-hard. Finally in the case of symmetric AFs we have that all semantics are tractable if the AF is irreflexive, but stable, semi-stable and stage are hard if the AF is not irreflexive.

Concerning fixed parameter tractability we have that reasoning in AFs is fixed parameter tractable w.r.t. tree-width and clique-width. Further we know that the cycle-rank, directed path-width, Kelly-width, DAG-width and directed tree-width do not lead to tractability.

Ordyniak and Szeider [95] follow a different approach for parametrising AFs. They consider the distance of the AF to a specific tractable fragments, in terms of arguments that have to be deleted such that the AF falls in such a fragment. They show that reasoning problems are fixed-parameter tractable when considering the distance to the class of acyclic AFs or the class of even-length cycle free AFs[6]. Moreover the showed that small distance to symmetric or bipartite AFs does not lead to tractability.

Several negative results for possible parametrisations are given in [44], showing that considering k-partite AFs and parametrising the in- and out- degree do not lead tractability. Moreover even the combination of these parameters still yields the full complexity. However, this results are just stated for preferred semantics [7].

Finally we identify one technical issue we had to leave open in this section. We have that deciding credulous acceptance w.r.t. resolution-based grounded semantics in a symmetric AF is certainly in P but we do not know whether it is complete for P.

[6] Notice that this does not hold for stage semantics (cf. Theorem 30) and that $resGr$ was not studied in [95].

[7] Nevertheless, the author has no doubt that the results in [44] can be extended to the other (admissible-based) argumentation semantics.

CHAPTER 5

Intertranslatability of Argumentation Semantics

We already have used translations between different argumentation semantics in the previous chapters to obtain complexity results (cf. Propositions 17 & 18). In this chapter we complement this picture by formalising the concept of a translation, adding further translations and showing some impossibility results. Informally a translation from a semantics σ to a semantics σ' is a function Tr such that the σ-extensions of each AF F are in certain correspondence to the σ'-extensions of the associated AF $Tr(F)$.

These translations are not only of theoretical interest but are also of practical value considering the reduction-approach for implementing argumentation systems. This is in the spirit of well-known work on intertranslatability of other non-monotonic formalism (see, e.g., [38, 39, 74, 79, 86, 90]). For example consider the case where we have an advanced argumentation engine for a semantics σ', but for some reason we want to evaluate AFs with another semantics σ. Then instead of implementing a new system from scratch, it might be a good idea to transform the AFs such that one can use the advanced system but can still easily obtain the extensions for semantics σ. If the required transformations are efficiently computable, this is more appealing than implementing a distinguished algorithm for the σ-semantics from scratch. We illustrate this idea in Figure 5.1. The concept of a filter is required in case the transformation introduces further arguments (which thus might appear in the σ'-extensions of the transformed AF) in the course of the translation making a filtering of these new arguments necessary to obtain the desired original extensions. However, when possible we will prefer translations for which such a back-translation is not necessary. Finally let us mention that the value of translations between argumentation semantics is not limited to computational issues but also has relevance in meta-argumentation or multi-agent argumentation scenarios. For a broader discussion the interested reader is referred to [56]. Finally let us mention that we omit parameterised semantics and this chapter and hence restrict ourselves to grounded, admissible, complete, stable, preferred, semi-stable and stage semantics.

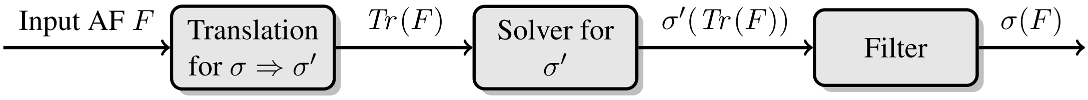


Figure 5.1: Solver for semantics σ.

The organisation of this chapter and its main contributions are as follows:

- Section 5.1 defines properties for translations basically along the lines of Janhunen [79]. In particular, we consider here as desired properties efficiency (the translation can be computed in logarithmic space w.r.t. the given AF), modularity (the translation can be done independently for certain parts of the framework) and faithfulness (there should be a clear correspondence between the extensions of the translated AF and the original AF). However, we also consider some additional features which are needed to deal with some of the argumentation semantics (for instance, the admissible semantics always yields the empty set as one solution; thus filtering such an entire solution is necessary).
- Section 5.2 contains our main results in this chapter, in particular we provide translations between grounded, stable, admissible, complete, preferred, semi-stable and stage stage when possible. We analyze these translations w.r.t. the properties mentioned above using as minimal desiderata efficiency and (a particular form of) faithfulness.
- As already mentioned, Section 5.3 then provides negative results, i.e. we show that certain translations between semantics are not possible. Some of these impossibility results make use of typical complexity-theoretic assumptions together with results from Chapter 3; others are genuine due to the different properties of the compared semantics.
- Finally, in Section 5.4 we conclude this chapter with a summary and discussion of the presented results. As well, an outlook to potential future work is given there.

This chapter is based on [56] (a short version was published in [55]).

5.1 Properties for Translations

In what follows, we understand as a translation Tr a function which maps AFs to AFs. In particular, we seek translations, such that for given semantics σ, σ', the extensions $\sigma(F)$ are in a certain relation to extensions $\sigma'(Tr(F))$ for each AF F. To start with, we introduce a few additional properties which seem desirable for such translations. To this end, we recall that, for AFs $F = (A, R)$, $F' = (A', R')$, the union $F \cup F'$ is defined as $(A \cup A', R \cup R')$, and inclusion $F \subseteq F'$ holds iff jointly $A \subseteq A'$ and $R \subseteq R'$ (see Definition 60).

Definition 77. *A translation Tr is called*

- efficient *if for every AF F, the AF $Tr(F)$ can be computed using logarithmic space w.r.t. to $|F|$;*

- covering *if for every AF* F*,* $F \subseteq Tr(F)$*;*

- embedding *if for every AF* F*,* $A_F \subseteq A_{Tr(F)}$ *and* $R_F = R_{Tr(F)} \cap (A_F \times A_F)$*;*

- monotone *if for any AFs* F, F'*,* $F \subseteq F'$ *implies* $Tr(F) \subseteq Tr(F')$*;*

- modular *if for any AFs* F, F'*,* $Tr(F) \cup Tr(F') = Tr(F \cup F')$.

As we mentioned we want to use translations as part of a reasoner (cf. Figure 5.1) hence it should be efficiently computable. Moreover if the translation already provides the complexity of the reasoning problems of a semantics we can use it to directly solve the problem. Thus the computational cost of a translation should be less than the computational cost of any semantics under our focus, i.e. less than P. Thus using the class of logarithmic space computable functions is appropriate for our purposes. This also allows us to compare semantics w.r.t. their expressiveness. In addition, one could seek translations which are minimal w.r.t. certain parameters (for instance, number of additional arguments and attacks). However, we decided not to design our translations towards such aims, since this would partly hide the main intuitions underlying the translations.

While the property of efficiency is clearly motivated by our computational issues, let us spend a few words on the other properties. Covering holding ensures that the translation does not hide some original arguments or conflicts. Being embedding, in addition, ensures that no additional attacks between the original arguments are pretended. While efficiency is motivated by the reduction approach and comparing expressiveness the properties of covering and embedding can be motivated by the meta-argumentation scenario. Translations which are covering or embedding preserve the arguments and conflicts we (meta)-argue about, an assumption one usually has in mind in the context of meta-argumentation. To put it in other words, having an embedding translation, the original framework and the meta-level part are clearly separated in the translated framework.

Monotonicity and modularity are crucial when extending the source AF after translation. Let us first consider monotonicity. In multi-agent scenarios it may be impossible for one agent to withdraw already interchanged arguments and attacks, as the other agents may not agree to forget arguments and conflicts they already know about; hence, re-translating the augmented source AF should respect the already existing translation. Now let us consider modularity and adding only a few arguments/attacks to a huge AF. When updating the translation it suffices to only consider the new arguments/attacks, instead of the whole source AF, which indeed can be of computational value. In the field of meta-argumentation, modular translations are in particular interesting as they are compatible with merging AFs. Thus one can interchange merge- and translation-operations, i.e. it does not make a difference if one first merges two AFs and then translates the union or first translates both AFs and then merges the translations. Moreover, as it can be easily checked each modular transformation is also monotone.

Next, we give two properties which refer to semantics. We note that our concept of faithfulness follows the definition used by Janhunen [79]; while exactness is in the spirit of bijective faithfulness w.r.t. equivalence as used by Liberatore [89].

Definition 78. *For semantics σ, σ' we call a translation Tr*

- exact *for $\sigma \Rightarrow \sigma'$ if for every AF F, $\sigma(F) = \sigma'(Tr(F))$;*
- faithful *for $\sigma \Rightarrow \sigma'$ if for every AF F, $\sigma(F) = \{E \cap A_F \mid E \in \sigma'(Tr(F))\}$ and $|\sigma(F)| = |\sigma'(Tr(F))|$.*

However, due to the very nature of the different semantics we want to consider, we need some less restricted notions. For instance, if we consider a translation from stable to some other semantics, we have to face the fact that some AFs do not possess a stable extension, while other semantics always yield at least one extension. The following definition takes care of this issue.

Definition 79. *For semantics σ, σ', we call a translation Tr*

- weakly exact *for $\sigma \Rightarrow \sigma'$ if there exists a collection $\mathcal{S}$ of sets of arguments, such that for any AF F, $\sigma(F) = \sigma'(Tr(F)) \setminus \mathcal{S}$;*
- weakly *faithful for $\sigma \Rightarrow \sigma'$ if there exists a collection $\mathcal{S}$ of sets of arguments, such that for any AF F, $\sigma(F) = \{E \cap A_F \mid E \in \sigma'(Tr(F)) \setminus \mathcal{S}\}$ and $|\sigma(F)| = |\sigma'(Tr(F)) \setminus \mathcal{S}|$.*

We sometimes refer to the elements from $\mathcal{S}$ as remainder sets. Note that $\mathcal{S}$ depends only on the translation, but not on the input AF. Thus, by definition, each $S \in \mathcal{S}$ only contains arguments which never occur in AFs subject to translation. In other words, we reserve certain arguments for introduction in weak translations.

Finally, we mention that the properties from Definition 77 as well as being exact, weakly exact and faithful are transitive, i.e. for two transformations satisfying one of these properties, also the concatenation satisfies the respective property. However, transitivity is not guaranteed for being weakly faithful.

5.2 Translations

In this section, we provide numerous faithful translations between the semantics introduced in Definition 4. As minimal desiderata, we want the translations to be efficient, monotone, and covering (see Definition 77). Thus, in this section when speaking about translations we tacitly assume that they satisfy at least these three properties.

5.2.1 Exact Translations

We start with a rather simple such translation, which we will show to be exact for $prf \Rightarrow sem$ and $adm \Rightarrow com$.

Translation 1. *The translation Tr_1 is defined as $Tr_1(F) = (A^*, R^*)$, where*

$$\begin{array}{rcl} A^* & = & A_F \cup A'_F \\ R^* & = & R_F \cup \{(a, a'), (a', a), (a', a') \mid a \in A_F\}, \end{array}$$

with $A'_F = \{a' \mid a \in A_F\}$.

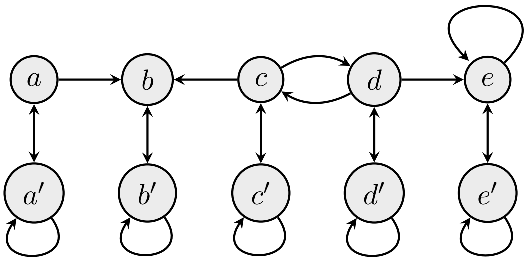


Figure 5.2: $Tr_1(F)$ for the AF F from Example 1.

A few words about the intuition behind the above translation (for illustration see Figure 5.2 which depicts the translation of our example AF from Example 1): the new arguments $a' \in A'_F$ are all self-attacking and thus never appear in any extension of the resulting framework. However, each a' attacks the original argument a (and a attacks a'), thus an argument a is only defended by a set E in $Tr_1(F)$ if $a \in E$. Consequently, we have that in $Tr_1(F)$ each admissible set is also a complete one.

Lemma 12. *For an AF F and a set E of arguments, the following propositions are equivalent:*

1. $E \in adm(F)$
2. $E \in adm(Tr_1(F))$
3. $E \in com(Tr_1(F))$

Proof. As all arguments in A'_F are self-conflicting, every conflict-free set E of $Tr_1(F)$ satisfies $E \subseteq A_F$. Further, since Tr_1 is embedding, E is conflict-free in F iff E is conflict-free in $Tr_1(F)$. Moreover, since Tr_1 only adds symmetric attacks against arguments $a \in A_F$, we have that E defends its arguments in F iff E defends its arguments in $Tr_1(F)$. Thus, $adm(F) = adm(Tr_1(F))$ and (1)⇔(2) follows. For (2)⇒(3), let $a \in A$ be an arbitrary argument and $E \subseteq A$. In $Tr_1(F)$ the argument a is attacked by a' and a is the only attacker (except a' itself) of a'. Hence, for each $a \in A$, E defends a only if $a \in E$ and thus every admissible set of $Tr_1(F)$ is also a complete one. Finally, (2)⇐(3) holds since $com(F) \subseteq adm(F)$ is true for any AF F. □

Concerning Tr_1 we observe another side effect. As already mentioned $a \in A$ is the only argument attacking a'. Thus different preferred extensions of $Tr_1(F)$ have incomparable range (recall Definition 12), and therefore each preferred extension of $Tr_1(F)$ is also a semi-stable extension of $Tr_1(F)$.

Lemma 13. *For an AF F and a set E of arguments, the following propositions are equivalent:*

1. $E \in prf(F)$

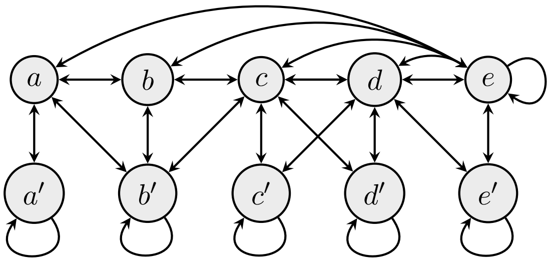


Figure 5.3: $Tr_2(F)$ for the AF F from Example 1.

2. $E \in \mathit{prf}(Tr_1(F))$
3. $E \in \mathit{sem}(Tr_1(F))$

Proof. For (1)⇔(2), it is sufficient to show that $E \in \mathit{adm}(F)$ iff $E \in \mathit{adm}(Tr_1(F))$ holds for each E. This is captured by Lemma 12. For (2)⇒(3), let $D, E \in \mathit{prf}(Tr_1(F))$ and, towards a contradiction, assume that $D^+_{R^*} \subset E^+_{R^*}$, i.e. $D \notin \mathit{sem}(Tr_1(F))$. As both D and E are preferred extensions, we have $D \not\subseteq E$. Thus, there exists an argument $a \in D \setminus E$. By construction of $Tr_1(F)$, we get $a' \in D^+_{R^*}$ but $a' \notin E^+_{R^*}$, a contradiction to $D^+_{R^*} \subset E^+_{R^*}$. (2)⇐(3) follows from the fact $\mathit{sem}(F) \subseteq \mathit{prf}(F)$ for any AF F. □

Obviously Tr_1 is an embedding translation and as the introduction of a new argument or attack in Tr_1 only depends on one original argument it is also modular. Together with the results from Lemma 12 and 13 we thus get our first main result.

Theorem 40. *Tr_1 is a modular, embedding, and exact translation for $\mathit{prf} \Rightarrow \mathit{sem}$ and $\mathit{adm} \Rightarrow \mathit{com}$.*

Next we will restate the translation already used in the proof of Proposition 17, but now using it for a different purpose, namely translating stage to semi-stable semantics. In addition to Tr_1, we make all attacks symmetric (thus Tr_2 will not be embedding) and add for each original attack (a, b) also an attack (a, b').

Translation 2. *The translation Tr_2 is defined as $Tr_2(F) = (A^*, R^*)$, where*

$$\begin{array}{rcl} A^* & = & A_F \cup A'_F \\ R^* & = & R_F \cup \{(b,a),(a,b'),(b',a) \mid (a,b) \in R_F\} \\ & & \cup\, \{(a,b),(b,a) \mid a \in A_F, (b,b) \in R_F\} \\ & & \cup\, \{(a,a'),(a',a),(a',a') \mid a \in A_F\} \end{array}$$

The symmetric attacks in $Tr_2(F)$ mirror the fact that we do not mind the orientation of attacks when considering conflict-freeness. In other words, we exploit the well known property that for symmetric frameworks conflict-free and admissible sets coincide. However, making attacks symmetric destroys the original range of extensions. Thus we make use of arguments $a' \in A'_F$ in the sense that, for a given set E of arguments, an argument a' is contained in $E^+_{R^*}$ iff a is contained in E^+_R. Likewise, we have to add attacks into self-defeating arguments. The technical reason for this is that we require that each original argument is attacked by each maximal conflict-free non-empty set in $Tr_2(F)$ (see also the proof of Proposition 17). For illustration we refer to Figure 5.3.

By results of Gaggl and Woltran [72], we can remove attacks of an AF where the source argument is self-attacking without changing the stage extensions of the AF. So in case the number of attacks in crucial one might omit the attacks $\{(b,a) \mid a \in A_F, (b,b) \in R_F\} \cup \{(a',a) \mid a \in A_F\}$ in translation Tr_2.

Lemma 14. *For an AF F and any set E of arguments, the following propositions are equivalent:*

1. $E \in stg(F)$
2. $E \in stg(Tr_2(F))$
3. $E \in sem(Tr_2(F))$

Proof. See proof of Proposition 17. □

By definition the translation Tr_2 is covering, but not embedding. Moreover, as each self-attacking argument of the original AF is attacked by all of the other original arguments Tr_2 is not modular. Together with the above lemma, we thus obtain the following result.

Theorem 41. *Tr_2 is an exact translation for $stg \Rightarrow sem$.*

The next translations consider the stable semantics as source formalism. Recall that not all AFs possess a stable extension, while this holds for all other semantics (also recall we excluded empty AFs for our considerations). Thus we have to use weak translations as introduced in Definition 79. Our first such translation is weakly exact and uses a single remainder set $\{t\}$ (recall the definition of remainder sets as given in Definition 79).

Translation 3. *The translation $Tr_3(F)$ is defined as $Tr_3(F) = (A^*, R^*)$ where*

$$\begin{aligned} A^* &= A_F \cup \{t\} \\ R^* &= R_F \cup \{(t,a),(a,t) \mid a \in A_F\} \end{aligned}$$

Here the intuition is rather simple, see also Figure 5.4. In fact, the new argument t in $Tr_3(F)$ encodes that there might not exist a stable extension for F. Thus none of the (other) arguments in $Tr_3(F)$ is accepted, whenever t is accepted. Since the argument t guards that there exists at least one stable extension of $Tr_3(F)$ (for any AF F), namely $\{t\}$, we can make use of the fact that stable, semi-stable and stage semantics thus coincide for $Tr_3(F)$.

Lemma 15. *Let $F = (A,R)$ be an AF and $E \subseteq A$. Then the following statements are equivalent:*

1. $E \in stb(F)$

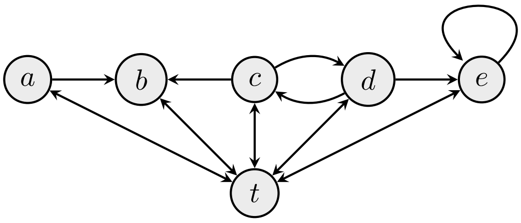


Figure 5.4: $Tr_3(F)$ for the AF F from Example 1.

2. $E \in stb(Tr_3(F))$
3. $E \in sem(Tr_3(F))$
4. $E \in stg(Tr_3(F))$

Further for each $E \in \sigma(Tr_3(F))$ with $\sigma \in \{stb, sem, stg\}$ either $E = \{t\}$ or $t \notin E$ holds.

Proof. As the translation does not modify the original AF F, i.e. Tr_3 is embedding, we have that for each $E \subseteq A_F$, E is conflict-free in F iff E is conflict-free in $Tr_3(F)$.

(1)⇒(2): Each $E \in stb(F)$ by definition is non-empty, conflict-free and satisfies $E^+_{R_F} = A_F$. By construction it also holds that $E \rightarrowtail^{R^*} t$ and thus $E^+_{R^*} = A^*$, i.e. $E \in stb(Tr_3(F))$. For (1)⇐(2) consider $E \in stb(Tr_3(F)), E \subseteq A_F$. Then by definition we have that E is conflict-free in $Tr_3(F))$ and thus in F; moreover, $E^+_{R^*} = A^*$ and as Tr_3 is embedding also $E^+_R = A_F$. Hence $E \in stb(F)$.

For (2)⇔(3)⇔(4), we mention that $\{t\}$ is a stable extension of $Tr_3(F)$ for any AF F. Furthermore, we know that if there exists a stable extension for an AF, then stable, semi-stable and stage extensions coincide.

Finally as the argument t is in conflict with all of the other arguments the only extension E with $t \in E$ is the set $\{t\}$. □

Adding argument t and the corresponding attacks to the source AF is a modular operation and as no further attacks are added Tr_3 is also embedding.

Theorem 42. *Tr_3 is modular, embedding and weakly exact for $stb \Rightarrow \sigma$, $\sigma \in \{sem, stg\}$.*

Proof. The result follows from Lemma 15, which states that $sem(Tr_3(F)) = stg(Tr_3(F)) = stb(F) \cup \{\{t\}\}$. Thus by taking as remainder set $\mathcal{S} = \{\{t\}\}$, Tr_3 is weakly exact. □

We continue with a different translation from stable to other semantics.

Translation 4. *Tr_4 is defined as $Tr_4(F) = (A^*, R^*)$ where*

$$\begin{aligned} A^* &= A_F \cup A'_F \\ R^* &= R_F \cup \{(b', a) \mid a, b \in A_F\} \\ &\quad \cup \{(a', a'), (a, a') \mid a \in A_F\} \\ &\quad \cup \{(a, b') \mid (a, b) \in R_F\}. \end{aligned}$$

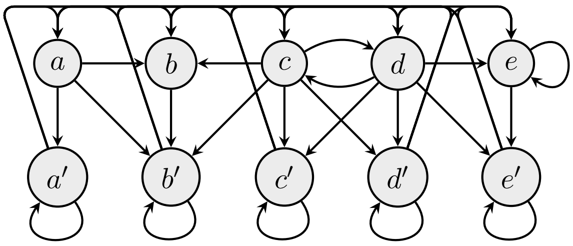


Figure 5.5: $Tr_4(F)$ for the AF F from Example 1.

As before in translation Tr_2, new arguments $a' \in A'_F$ are used to encode the range of an extension in the sense that a' is attacked by a set E in $Tr_4(F)$ only if a is in the range of E in F. However, given the fact that each $a' \in A'_F$ attacks back all original arguments $a \in A$, we can now accept an argument in a set E only if all arguments are in the range of E. For illustration on our running example, see Figure 5.5. Observe that in our example each of the arguments a', b', c', d', e' attacks each of the arguments a, b, c, d, e.

Lemma 16. *Let $F = (A, R)$ be an AF and $E \subseteq A$ with $E \neq \emptyset$. Then, the following statements are equivalent:*

1. $E \in stb(F)$

2. $E \in stb(Tr_4(F))$

3. $E \in adm(Tr_4(F))$

4. $E \in prf(Tr_4(F))$

5. $E \in com(Tr_4(F))$

6. $E \in sem(Tr_4(F))$

Further for each conflict-free set E of $Tr_4(F)$ it holds that $E \subseteq A$.

Proof. First, as all arguments $a' \in A'$ are self-attacking, for each conflict-free set E in $Tr_4(F)$ it holds that $E \subseteq A$. Since the translation is embedding, any set E is conflict-free in F iff it is conflict-free in $Tr_4(F)$. To show (1)⇒(2), let $E \in stb(F)$. Hence, for all $a \in A \setminus E$, $E \rightarrowtail^R a$. We now claim that each argument in $A^* \setminus E$ is attacked by E in $Tr_4(F)$. We distinguish between two cases for the different arguments in $A^* \setminus E$:

(i) $a \in A \setminus E$: The construction of $Tr_4(F)$ preserves all attacks in R. Thus as each $a \in A \setminus E$ satisfies $E \rightarrowtail^R a$, we obtain that $E \rightarrowtail^{R^*} a$

(ii) $a' \in A'$: In case $a \in E$ we have $E \rightarrowtail^{R^*} a'$, since $(a, a') \in R^*$ In case $a \in A \setminus E$, by the assumption $E \in stb(F)$, there exists an argument $b \in E$ such that $(b, a) \in R$. But then by construction $(b, a') \in R^*$ and thus $E \rightarrowtail^{R^*} a'$.

Together with our observations about conflict-free sets, we get $E \in \mathit{stb}(\mathit{Tr}_4(F))$.

Vice versa, to show (1)⇐(2) we get, for $E \in \mathit{stb}(\mathit{Tr}_4(F))$, $E \rightarrowtail^{R^*} a$, for each $a \in A^* \setminus E$, and thus, in particular, for each $a \in A \setminus E$. By definition of Tr_4, we also have $E \rightarrowtail^{R} a$ for each $a \in A \setminus E$. Thus $E \in \mathit{stb}(F)$ follows.

To show (2)⇐(3), let E be a nonempty admissible extension of $\mathit{Tr}_4(F)$ and $a \in E$. By construction, we have that $a^{\ominus}_{R^*} = \{b \in A^* \mid (b,a) \in R^*\} \supseteq A'$. As $E \in \mathit{adm}(\mathit{Tr}_4(F))$, $E \rightarrowtail^{R^*} a'$ for each $a' \in A'$. But $E \rightarrowtail a'$ only if either $a \in E$ or $E \rightarrowtail^{R^*} a$. Thus for every $a \in A^*$ it holds that either $a \in E$ or $E \rightarrowtail^{R^*} a$; hence, $E \in \mathit{stb}(\mathit{Tr}_4(F))$.

The remaining implications follow by well-known relations between the semantics, i.e. $\mathit{stb}(G) \subseteq \mathit{sem}(G) \subseteq \mathit{prf}(G) \subseteq \mathit{com}(G) \subseteq \mathit{adm}(G)$, for each AF G. Hence, in particular, since for $\mathit{Tr}_4(F)$, stable extensions and non-empty admissible sets coincide, the claim follows. □

Clearly Tr_4 is an embedding translation, but as for each new argument we add attacks to all original arguments, Tr_4 is not modular.

Theorem 43. *Tr_4 is an embedding and weakly exact translation for $\mathit{stb} \Rightarrow \sigma$ with $\sigma \in \{\mathit{adm}, \mathit{com}, \mathit{prf}, \mathit{sem}\}$.*

Proof. By Lemma 16, we in particular have that $\mathit{stb}(F) = \sigma(\mathit{Tr}_4(F)) \setminus \{\emptyset\}$, for any AF F. Thus taking $\emptyset$ as a remainder set, we obtain that Tr_4 is weakly exact for the involved semantics. □

Thus we have that both Tr_3 and Tr_4 are weakly exact translations for $\mathit{stb} \Rightarrow \mathit{sem}$, of course with different remainder sets. Due the to different properties of two translations it depends on the concrete application which of them would be the better choice.

5.2.2 Faithful Translations

So far, we have only introduced exact and weakly exact translations. We now present translations which relax this semantical property, i.e. we switch to faithful translations. As a first example, we consider a translation for $\mathit{stg} \Rightarrow \mathit{sem}$ which is faithful and embedding, but not exact. This is in contrast to translation Tr_2 which is exact for $\mathit{stg} \Rightarrow \mathit{sem}$ but not embedding. As we will see in Section 5.3 it is impossible to give a translation that is both embedding and exact for $\mathit{stg} \Rightarrow \mathit{sem}$, thus one has to decide which property is more important for a concrete application scenario.

Translation 5. *The translation $\mathit{Tr}_5(F)$ is defined as $\mathit{Tr}_5(F) = (A^*, R^*)$ where*

$$\begin{aligned} A^* &= A_F \cup \bar{A}_F \cup A'_F \\ R^* &= R_F \cup \{(a,\bar{a}), (\bar{a},a) \mid a \in A_F\} \\ &\quad \cup \{(a,a'), (a',a') \mid a \in A_F\} \\ &\quad \cup \{(a,b') \mid (a,b) \in R_F\} \end{aligned}$$

As in $\mathit{Tr}_2(F)$ the arguments $a' \in A'_F$ handle the range of the original extensions. But instead of making original attacks symmetric (as in Tr_2) we add the arguments $\bar{a} \in \bar{A}_F$ to

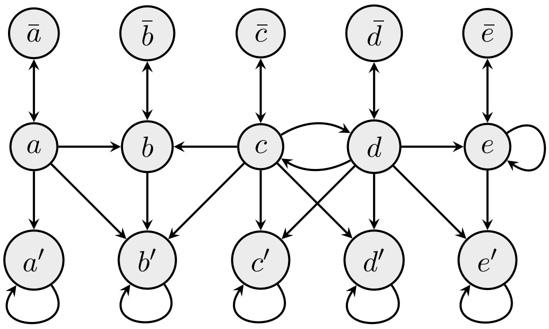


Figure 5.6: $Tr_5(F)$ for the AF F from Example 1.

encode that an argument is not in the extension (also compare Figures 5.3 and 5.6). In fact, such meta-arguments indicating that some a is out of an extension will be used in all faithful translations presented in this subsection.

Lemma 17. *Let $F = (A, R)$ be an AF, $E \subseteq A$ and $E^* = E \cup (\overline{A \setminus E})$. The following statements are equivalent:*

1. *$E \in stg(F)$*
2. *$E^* \in stg(Tr_5(F))$*
3. *$E^* \in sem(Tr_5(F))$*

Moreover for each $S \in sem(Tr_5(F))$ there exists a set $E \subseteq A$ such that $S = E \cup (\overline{A \setminus E})$.

Proof. First we prove that each $S \in stg(Tr_5(F))$ is of the form $S = E \cup (\overline{A \setminus E})$. As S is conflict-free we have that $A'_F \cap S = \emptyset$ (each $a' \in A'$ is self-attacking) and for each $a \in A$ that $\{a, \bar{a}\} \not\subseteq E^*$ (as a attacks $\bar{a}$ and vice versa). Further as each stage extension is also a $\subseteq$-maximal conflict-free set we have that for each $a \in A$ either $a \in S$ or $\bar{a} \in S$. Hence there exists an $E \subseteq A$ such that $S = E \cup (\overline{A \setminus E})$.

(1)⇒(2): Let $E \in stg(F)$. It is easy to see that E^* is conflict-free in $Tr_5(F)$. By construction for each argument $a \in A$ either $a \in E^*$ or $\bar{a} \in E^*$ holds and there are mutual attacks between a and $\bar{a}$, hence we have that $A \cup \bar{A} \subseteq (E^*)^+_{R^*}$. Next we observe that each $a' \in A'$ is self-attacking and thus $a' \in (E^*)^+_{R^*}$ iff $E^* \rightarrowtail a'$. Further by the definition of $Tr_5(F)$ each argument a' is attacked by a and all arguments b such that $(b, a) \in R$. That is $a' \in (E^*)^+_{R^*}$ iff either $a \in E$ or there exists a $b \in A$ such that $(b, a) \in R$ iff $a \in (E)^+_R$. By assumption E is a stage extension of F and thus we have that $(E)^+_R$ is $\subseteq$-maximal. Using the above observation we have that also $(E^*)^+_{R^*}$ is $\subseteq$-maximal in $Tr_5(F)$ and therefore $E^* \in stg(Tr_5(F))$.

(1)⇐(2): Let $E^* \in stg(Tr_5(F))$. We recall that E^* is of the form $S = E \cup (\overline{A \setminus E})$, for some $E \subseteq A$. It can be easily checked that E is conflict-free in F. By the above observation that $a' \in (E^*)^+_{R^*}$ iff $a \in (E)^+_R$ and the fact that $(E^*)^+_{R^*}$ is $\subseteq$-maximal in $Tr_5(F)$ we get that also E^+_R is $\subseteq$-maximal in F. Hence, $E \in stg(F)$.

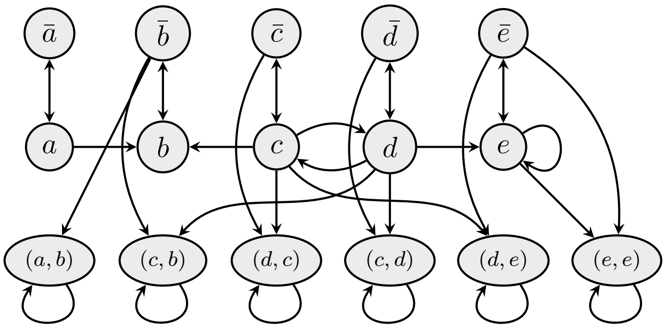


Figure 5.7: $Tr_6(F)$ for the AF F from Example 1.

(2)⇔(3): Let us consider $E^* \in stg(Tr_5(F))$. As we have already observed, E^* is of the desired form and for each $a \in A_F \cup \bar{A}_F$ either $a \in E^*$ or $E^* \rightarrowtail a$. Further by construction an argument $b \in A'_F$ does not attack E^*. We can conclude that each stage extension defends itself against all attackers, i.e. is an admissible set. Hence, stage and semi-stable extensions of $Tr_5(F)$ coincide. □

By above lemma and construction of Tr_5, the following result is immediate.

Theorem 44. *Tr_5 is a modular, embedding and faithful translation for $stg \Rightarrow sem$.*

Next we give a faithful translation from admissible semantics to stable, semi-stable and stage semantics.

Translation 6. *The translation $Tr_6(F)$ is defined as $Tr_6(F) = (A^*, R^*)$ where*

$$\begin{aligned} A^* &= A_F \cup \bar{A}_F \cup R_F \\ R^* &= R_F \cup \{(a, \bar{a}), (\bar{a}, a) \mid a \in A_F\} \\ &\quad \cup \{(r, r) \mid r \in R_F\} \\ &\quad \cup \{(\bar{a}, r) \mid r = (y, a) \in R_F\} \\ &\quad \cup \{(a, r) \mid r = (z, y) \in R_F, (a, z) \in R_F\} \end{aligned}$$

The main idea is to use additional arguments $(a, b) \in A^*$ which represent the attack relations from the source framework in order to capture admissibility as follows: (a, b) is attacked by an extension E^* in $Tr_6(F)$ if (a, b) is not critical w.r.t. the corresponding extension E in F, meaning that either $b \notin E$ or there exists a $c \in E$ such $(c, a) \in R_F$, i.e. E defends b against the attack (a, b). For instance, consider the argument (c, b) in the translation of our example framework as depicted in Figure 5.7. Then, we have that (1) $\bar{b}$ attacks (c, b) since if b is chosen to be out (i.e. $\bar{b}$ is chosen in), there is no need to defend b; (2) d attacks (c, b) since if d is chosen in, d defends b against attacker c (recall that (d, c) is present in the source AF). Thus, as long as (c, b) is attacked by some argument, b is treated corrected in terms of admissibility (w.r.t. attacker c). Note that in our example b cannot be defended against a, thus the only way to get (a, b) into the range is to select b to be out.

Lemma 18. *Let $F = (A, R)$ be an AF, $E \subseteq A$ and $E^* = E \cup (\overline{A \setminus E})$. The following statements are equivalent:*

1. $E \in adm(F)$
2. $E^* \in stb(Tr_6(F))$
3. $E^* \in sem(Tr_6(F))$
4. $E^* \in stg(Tr_6(F))$

Moreover for each $E^* \in \sigma(Tr_6(F))$ ($\sigma \in \{stb, sem, stg\}$) there exists a set $E \subseteq A$ such that $E^* = E \cup (\overline{A \setminus E})$.

Proof. (1)⇒(2): Let $E \in adm(F)$. It is easy to see that E^* is conflict-free in $Tr_6(F)$ and further that $A \cup \bar{A} \subseteq (E^*)^+_{R^*}$. It remains to show that each argument $r \in A^*$ for $r \in R$ is attacked by E^*. Let (a, b) be such an argument r. If $b \notin E$ then $\bar{b} \in E^*$ and thus $E^* \rightarrowtail^{R^*} r$. Otherwise, $b \in E$ (thus $b \in E^*$) and, by assumption, E defends b in F, i.e. $(c, a) \in R$ for some $c \in E$ (thus $c \in E^*$). By construction, $(c, r) \in R^*$ and $E^* \rightarrowtail^{R^*} r$.

(1)⇐(2): Let $E^* \in stb(Tr_6(F))$. E^* is conflict-free, thus $R \cap E^* = \emptyset$ and $\{a, \bar{a}\} \not\subseteq E^*$ for all $a \in A$. By construction, E is conflict-free in F. It remains to show that E defends all its arguments in F. Let $b \in A \setminus E$ such that $b \rightarrowtail^R a$ for some $a \in E$. Then there exists an argument (b, a) in $Tr_6(F)$ attacked by E. As $a \in E$ we have that $\bar{a} \notin E^*$ and thus there exists an argument $c \in E$ such that $(c, b) \in R$.

(2)⇔(3)⇔(4): As the empty set is always admissible we have that $\bar{A}$ is always a stable extension of $Tr_6(F)$. Hence, stable, semi-stable and stage extensions coincide in $Tr_6(F)$, for any AF F. □

Observe that in the construction of Tr_6 drawing attacks $\{(a, r) \mid r = (z, y) \in R_F, (a, z) \in R_F\}$ depends on two attacks and three arguments from the original framework. Hence Tr_6 is not modular. By Lemma 18 the next result follows quite easily.

Theorem 45. *Translation Tr_6 is embedding and faithful for $adm \Rightarrow \sigma$ ($\sigma \in \{stb, sem, stg\}$).*

In our faithful translation from complete to stable semantics which we present next, we extend the given AF by arguments that represent whether an argument is attacked in the corresponding extension or not. Further we add arguments that ensure admissibility and completeness. The entire translation is thus slightly more complicated; see also Figure 5.8 which depicts the translated framework for our running example.

Translation 7. *The translation $Tr_7(F)$ is defined as $Tr_7(F) = (A^*, R^*)$ where*

$$\begin{aligned}
A^* &= A_F \cup \bar{A}_F \cup A^\circ_F \cup \bar{A}^\circ_F \cup A'_F \cup R_F \\
R^* &= R_F \cup \{(x, x) \mid x \in A'_F \cup R_F\} \\
&\quad \cup \{(a, \bar{a}), (\bar{a}, a), (\bar{a}^\circ, a^\circ), (a, a') \mid a \in A_F\} \\
&\quad \cup \{(a, \bar{b}^\circ), (\bar{a}^\circ, b') \mid (a, b) \in R_F\} \\
&\quad \cup \{(\bar{a}, r), (b^\circ, r) \mid r = (b, a) \in R_F\}
\end{aligned}$$

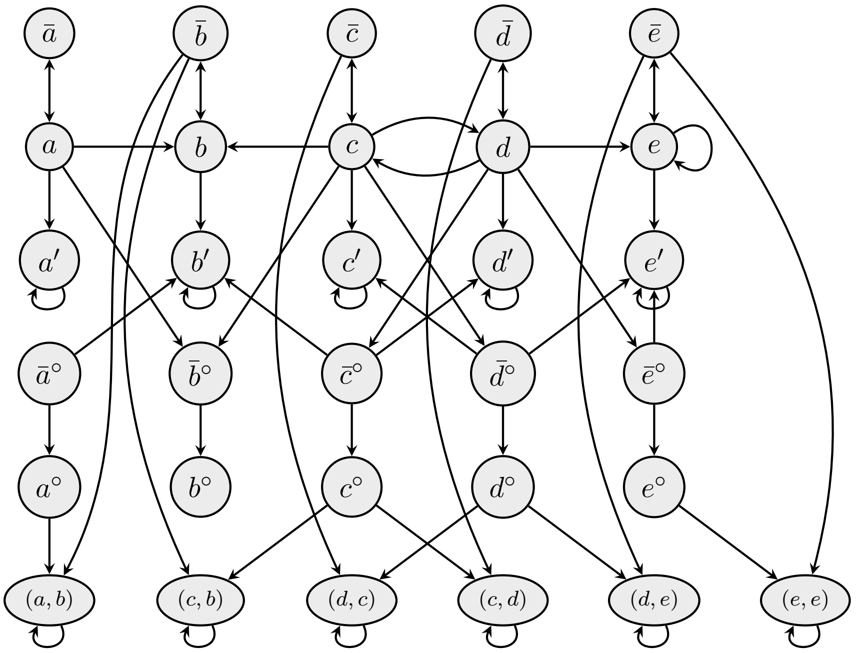


Figure 5.8: $Tr_7(F)$ for the AF F from Example 1.

The intuition behind arguments A'_F, $\bar{A}_F$, and R_F is similar as in previous translations. An argument $a^\circ \in A^\circ_F$ indicates that a is attacked by an extension E of F, i.e. $a \in E^\oplus$, while $\bar{a}^\circ \in \bar{A}^\circ_F$ says that a is not attacked by E, i.e. $a \notin E^\oplus$.

Lemma 19. *Let $F = (A, R)$ be an AF, $E \subseteq A$ and $E^* = E \cup \overline{(A \setminus E)} \cup \{a^\circ \mid E \rightarrowtail^R a\} \cup \{\bar{a}^\circ \mid E \not\rightarrowtail^R a\}$. Then the following statements are equivalent:*

1. $E \in com(F)$
2. $E^* \in stb(Tr_7(F))$
3. $E^* \in sem(Tr_7(F))$
4. $E^* \in stg(Tr_7(F))$

Moreover for each $E^* \in \sigma(Tr_6(F))$ ($\sigma \in \{stb, sem, stg\}$) there exists a set $E \subseteq A$ such that $E^* = E \cup \overline{(A \setminus E)} \cup \{a^\circ \mid E \rightarrowtail^R a\} \cup \{\bar{a}^\circ \mid E \not\rightarrowtail^R a\}$.

Proof. To show (1)⇒(2), let $E \in com(F)$. Then by construction E^* is conflict-free in $Tr_7(F)$ (for $x, y \in E$ we have $x \rightarrowtail^R y \Leftrightarrow x \rightarrowtail^{R^*} y$). Moreover, by definition of E^*, it can be verified that $A \cup \bar{A} \cup A^\circ \cup \bar{A}^\circ \subseteq (E^*)^+_{R^*}$. Thus it remains to show that (i) $A' \subseteq (E^*)^+_{R^*}$ and (ii) $R \subseteq (E^*)^+_{R^*}$.

(i) Let $a \in A$ be an arbitrary argument of F. As E is a complete extension we have that either $a \in E$, and thus $a \in E^*$, or there exists an attack $(b, a) \in R$ with $E \not\rightarrowtail^R b$, and thus $\bar{b}^\circ \in E^*$. As by construction $(\bar{b}^\circ, a') \in R^*$ we thus have that $E^* \rightarrowtail^{R^*} a'$.

(ii) Let $r = (b, a) \in R$ be an arbitrary attack of F. As E is admissible it holds that either $a \notin E$, and thus $\bar{a} \in E^*$, or $E \rightarrowtail^R b$, and thus $b^\circ \in E^*$. In both cases $E^* \rightarrowtail^{R^*} r$.

Putting things together, we get that $A \cup \bar{A} \cup A^\circ \cup \bar{A}^\circ \cup A' \cup R = A^* \subseteq (E^*)^+_{R^*}$ which is equivalent to E^* being a stable extension of $Tr_7(F)$.

To show (1)⇐(2), let $E^* \in stb(Tr_7(F))$. First we prove that E^* is of the desired form. As E^* is both conflict-free and $\subseteq$-maximal we clearly have that $E^* \cap (A \cup \bar{A}) = E \cup \overline{A \setminus E}$ for some $E \subseteq A$. Let now $a \in A$ be an arbitrary argument. We have that $a^\circ \in E^*$ iff $\bar{a}^\circ \notin E^*$. But as E^* is stable $\bar{a}^\circ \notin E^*$ iff there exists an attack $(b, \bar{a}^\circ)$ such that $b \in E^*$. By construction of $Tr_7(F)$ this is equivalent to $b \in E$ and therefore $E \rightarrowtail^R a$. Thus E^* is of the desired form, it remains to show that E is complete. As mentioned before we have for $x, y \in E : x \rightarrowtail^R y \Leftrightarrow x \rightarrowtail^{R^*} y$ and thus E is conflict-free in F. Thus it remains to show that (i) E defends each of its arguments in F and (ii) E contains each argument defended by E in F.

(i) Let us assume there exists an argument $a \in E$ not defended by E. Thus there exists $r = (b, a) \in R, E \not\rightarrowtail b$. By construction we also have that $\bar{a} \notin E^*$ (as $a \in E$) and $b^\circ \notin E^*$ (as $E \not\rightarrowtail b$). But in $Tr_7(F)$ the self-attacking argument r is only attacked by the arguments $\bar{a}, b^\circ$ (and itself). Hence, this is in contradiction to E^* being a stable extension.

(ii) Let $a \in A$ be an argument defended by E. Then for all arguments $b \rightarrowtail^R a$ we have that $E \rightarrowtail^R b$ and thus $b^\circ \in E^*$ and $\bar{b}^\circ \notin E^*$. Recall that in $Tr_7(F)$ the argument a' is self-attacking and thus does not belong to E^* and is only attacked by the arguments $\bar{b}^\circ$ and a. As E^* is a stable extension and $a' \notin E^*$ we have that $a \in E^*$ and $a \in E$.

(2) ⇔ (3) ⇔ (4): As there always exists a complete extension we know that any framework $Tr_7(F)$ has a stable extension. But then stable, stage and semi-stable extensions coincide. □

Translation Tr_7 introduces a huge number of new arguments, despite this the introduction of a concrete argument or attack only depends on a single argument or attack. Hence Tr_7 is modular. It is easily checked that Tr_7 is also embedding. Together with Lemma 19 we thus can state the following result for Tr_7.

Theorem 46. *Tr_7 is a modular, embedding and faithful translation for $com \Rightarrow \sigma$ with $\sigma \in \{stb, sem, stg\}$.*

Finally we present a translation from grounded semantics to most of the other semantics under our focus, i.e. to all semantics except admissible semantics. The main idea is to simulate the computation of the least fixed-point of the characteristic function $\mathcal{F}_F(S) = \{x \in A_F \mid x \text{ is defended by } S\}$ of an AF F within the target AF.

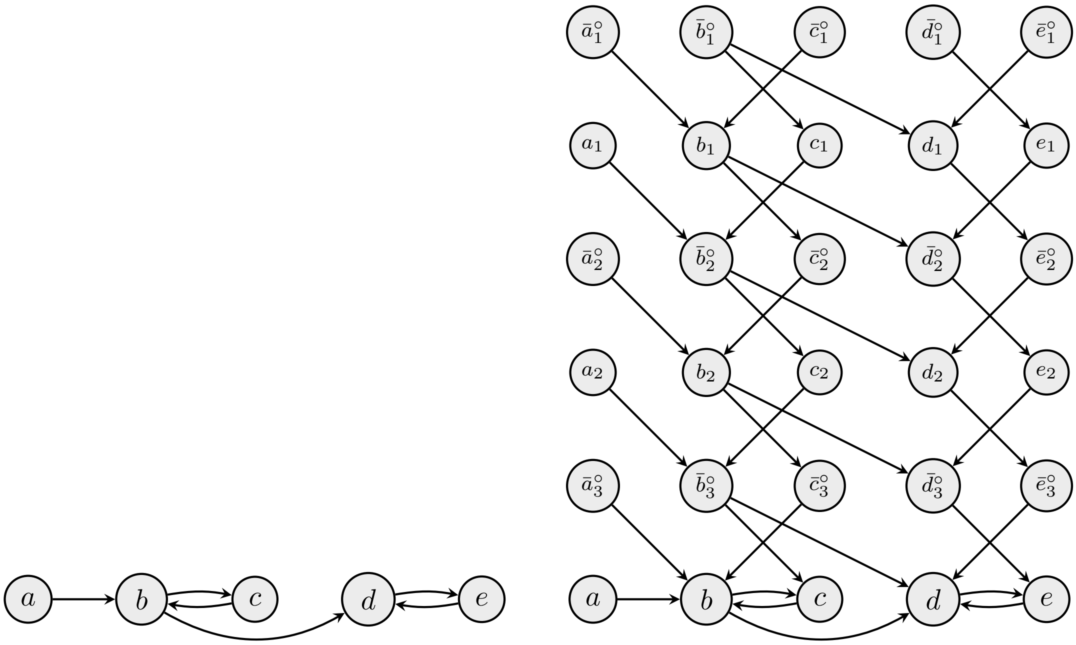


Figure 5.9: An example for Tr_8.

Translation 8. *The translation* $Tr_8(F)$ *is defined as* $Tr_8(F) = (A^*, R^*)$ *where*

$$\begin{array}{rcl} A^* & = & A_{F,1} \cup \bar{A}^\circ_{F,1} \cup \cdots \cup A_{F,l} \cup \bar{A}^\circ_{F,l} \\ R^* & = & R_F \cup \{(\bar{a}^\circ_i, b_i) \mid (a,b) \in R, i \in [l]\} \\ & & \cup \{(a_i, \bar{b}^\circ_{i+1}) \mid (a,b) \in R, i \in [l-1]\} \end{array}$$

with $A_F = A_{F,l}$ *and* $l = \lceil \frac{|A_F|}{2} \rceil$.

For illustration, we use here a slightly different example depicted in Figure 5.9(a). Observe that this AF has $\{a, c\}$ as its grounded extension. The translated framework is given in Figure 5.9(b).

The intuition behind arguments $a_i \in A_{F,i}$ is that $a \in \mathcal{F}^i_F(\emptyset)$, while the intuition of $\bar{a}^\circ_i \in \bar{A}^\circ_{F,i}$ is that $\mathcal{F}^{(i-1)}_F(\emptyset) \not\mapsto a$. The integer l is an upper bound for the number of iterations we need to reach the least fixed-point, i.e. the grounded extension.

Lemma 20. *Let* $F = (A, R)$ *be an AF and* E^* *the grounded extension of* $Tr_8(F)$*. Then* $E^* \cap A$ *is the grounded extension of* F*. We further have that on* $Tr_8(F)$ *the grounded, stable, complete, preferred, semi-stable and stage extensions coincide.*

Proof. We recall the definition of the characteristic function $\mathcal{F}_F$ of an AF F (see Definition 20), defined as $\mathcal{F}_F(S) = \{x \in A_F \mid x \text{ is defended by } S\}$, and that the grounded extension of F is the least fix-point of $\mathcal{F}_F$. Further we use as a shorthand $F^* = Tr_8(F)$. One can show that for arbitrary $a \in A$ we have

(i) $a_i \in E^*$ iff $a \in \mathcal{F}_F^i(\emptyset)$;

(ii) $\bar{a}_i^\circ \in E^*$ iff $\mathcal{F}_F^{i-1}(\emptyset) \not\rightarrowtail^R a$; and

(iii) $A_{F,i} \subseteq (E^*)^+_{R^*}$.

(iv) $\bar{A}^\circ_{F,i} \subseteq (E^*)^+_{R^*}$.

We prove this by structural induction. As induction base we show (ii) and (iv) for the arguments $\bar{a}_1^\circ$. For $a \in A$ we have that $\bar{a}_1^\circ \in E^*$ as they are not attacked by any argument. This coincides with the fact that $\mathcal{F}_F^0(\emptyset) = \emptyset$ doesn't attack any argument and thus (ii) and (iv) holds.

We have two induction steps: (1) Showing that (i) and (iii) hold for arbitrary n iff (ii) and (iv) hold for n; and (2) showing that (ii) and (iv) hold for arbitrary n iff (i) and (iii) hold for $n-1$.

(1) We assume (ii) and (iv) hold for all $\bar{a}_n^\circ$. By the definition of $\mathcal{F}_F$ we have that $a \in \mathcal{F}_F^n(\emptyset)$ iff all $b \in a^\ominus = \{b \in A \mid b \rightarrowtail a\}$ are attacked by $\mathcal{F}_F^{n-1}(\emptyset)$. Applying the induction hypothesis (ii) to $b \in a^\ominus$ we obtain that $a \in \mathcal{F}_F^n(\emptyset)$ iff each $\bar{b}_i^\circ \in \{\bar{b}_i^\circ \mid (b,a) \in R\}$ is attacked by E^*. Further, as by the construction of $Tr_8(F)$ these are the only attackers of a, this is equivalent to argument a_i being defended by E^*. Now recall that the each argument defended by the grounded extension is indeed contained in the grounded extension. Hence, $a \in \mathcal{F}_F^n(\emptyset)$ iff $a_i \in E^*$ and (i) holds.

To show (iii) we consider $a_i \in A_{F,i}$. If $a_i \in E^*$ then clearly $a_i \in (E^*)^+_{R^*}$. Thus let us consider $a_i \notin E^*$. Then, by the above observations, there exists a $\bar{b}_i^\circ$ such that $\bar{b}_i^\circ \rightarrowtail a_i$ and $E^* \not\rightarrowtail \bar{b}_i^\circ$. Using the latter and the induction hypothesis (iv) we obtain that $\bar{b}_i^\circ \in E^*$. Now we have that $E^* \rightarrowtail a_i$, hence $a_i \in (E^*)^+_{R^*}$ and we obtain (iii).

(2) Now let us assume that (i) and (iii) hold for all a_{n-1}. We have that $\mathcal{F}_F^{n-1}(\emptyset) \rightarrowtail a$ iff there exists $b \in \mathcal{F}_F^{n-1}(\emptyset) \cap \{b \mid (b,a) \in R^*\}$. By induction hypothesis this holds iff there exists a $b_{i-1} \in E^*$ such that $(b,a) \in R$. In other words there exists $b_{i-1} \in E^*$ such that $b_{i-1} \rightarrowtail^{R^*} \bar{a}_i^\circ$, which implies that $\bar{a}_i^\circ \notin E^*$. Moreover if there is no $b_{i-1} \in E^*$ such that $b_{i-1} \rightarrowtail^{R^*} \bar{a}_i^\circ$, by assumption (iii) we have that E^* defends $\bar{a}_i^\circ$ and thus $\bar{a}_i^\circ \in E^*$. Hence (ii) and (iv) hold.

Furthermore when applying the $\mathcal{F}_F$ operator we either add a new argument to the set and attack an additional argument or we reach the fixed-point. So in each step we make a decision about at least two arguments and thus $\mathcal{F}_F^l(\emptyset) = grd(F)$. In combination with (i), we get that $a_l \in E^*$ iff $a \in grd(F)$. Moreover by (iii) and (iv) it holds that E^* is also a stable extension and thus $grd(F^*) = stb(F^*) = com(F^*) = prf(F^*) = sem(F^*) = stg(F^*)$. □

As in Tr_8 the integer value l depends on the size S of the source AF, Tr_8 is not modular. However, it can be verified that the computation of the translation only requires logarithmic space w.r.t. S and that Tr_8 is embedding (the original AF is indeed contained in the resulting AF; see also the bottom layer in Figure 5.9(b)). Our final result concerning translations thus follows immediately from Lemma 20.

Theorem 47. *Tr_8 is an embedding and faithful translation for $grd \Rightarrow \sigma$ ($\sigma \in \{stb$, com, $prf, stg, sem\}$).*

5.3 Impossibility Results

In this section, we present results fortifying that for several semantics there does not exist any translation with the desired properties. The first result, which is rather straight forward, relies on the fact that the grounded semantics is a unique-status semantics.

Proposition 30. *There is no (weakly) faithful translation for $\sigma \Rightarrow grd$ with $\sigma \in \{sem,\ stg, prf,$ $com, stb, adm\}$.*

Proof. For instance consider the AF $F = (\{a, b\}, \{(a, b), (b, a)\})$. We have that $\{\{a\}, \{b\}\} \subseteq \sigma(F)$ for $\sigma \in \{sem,\ stg, prf, com, stb, adm\}$ but the grounded semantics always proposes a unique extension. □

We observe that in general it holds that if σ is a multiple status semantics and σ' is a unique status semantics then there is no (weakly) faithful translation for $\sigma \Rightarrow \sigma'$.

Further results are based on complexity gaps between different semantics (see Table 3.3) and the fact that certain translations preserve some decision problem. We start with cases where it is impossible to find efficient faithful translations; even if we allow for weakly faithful translations, cf. Definition 79. Afterwards, we give some negative results concerning (weakly) exact translations.

The following theorem concerns the intertranslatability of preferred, semi-stable and stage semantics, i.e. the semantics where skeptical acceptance is Π_2^P-complete. The underlying reason for the impossibility result is the complexity gap for the credulous acceptance problems.

Theorem 48. *There is no efficient (weakly) faithful translation for $sem \Rightarrow prf$ or $stg \Rightarrow prf$ unless $\Sigma_2^P = NP$.*

Proof. $\sigma \in \{sem, stg\}$ to prf. By definition this translation is L-computable and as we show next reduces $Cred_\sigma$ to $Cred_{prf}$: Let $F = (A, R)$ be an arbitrary AF, $x \in A$ an argument. First let us assume that x is credulously accepted w.r.t. to σ. Hence, there exists an $E \in \sigma(F)$ with $x \in E$. As Tr is a weakly faithful translation, there is an $E^* \in prf(Tr(F))$, such that $E^* \cap A = E$. Thus $x \in E^*$, i.e. x is credulously accepted w.r.t. preferred semantics in $Tr(F)$.

So assume x is credulously accepted in $Tr(F)$ w.r.t. to prf, i.e. $x \in E^*$ for some $E^* \in prf(Tr(F))$. By $x \in E^* \cap A$ we can conclude that E^* is not a remainder set of Tr. As Tr is a weakly faithful translation we have that $E = E^* \cap A$ is in $\sigma(F)$, and thus x is credulously accepted in F w.r.t. σ. Thus, Tr is a L-reduction from the Σ_2^P-hard problem $Cred_\sigma$ to the NP-easy problem $Cred_{prf}$. □

The following theorem makes use of complexity gaps for the skeptical acceptance.

Theorem 49. *There is no efficient (weakly) faithful translation for $\sigma \Rightarrow \sigma'$, where $\sigma \in \{sem,$ $stg, prf\}$ and $\sigma' \in \{com, stb, adm\}$, unless $\Sigma_2^P = NP$.*

Proof. Given an efficient weakly faithful translation Tr with remainder set $\mathcal{S}$ for $\sigma \Rightarrow \sigma'$ we have that Skept_σ is translated to the problem $\mathit{Skept}_{\sigma'}^{\mathcal{S}}$, that is deciding whether an argument is in each σ'-extension which is not in the set $\mathcal{S}$. Next we show that the problem $\mathit{Skept}_{\sigma'}^{\mathcal{S}}$ remains in coNP. One can disprove $\mathit{Skept}_{\sigma'}^{\mathcal{S}}$, by guessing a set $E \subseteq A$, such that $a \notin E$ and verify that $E \in \sigma'(F)$ and $E \notin \mathcal{S}$. As $\mathit{Ver}_{\sigma'} \in$ P and the set $\mathcal{S}$ is fixed, i.e. $\mathcal{S}$ does not depend on the input, this is an NP-algorithm. Hence proving $\mathit{Skept}_{\sigma'}^{\mathcal{S}}$ is in coNP. Thus Tr would be an L-reduction from the Π_2^P-hard problem Skept_σ to the coNP-easy problem $\mathit{Skept}_{\sigma'}^{\mathcal{S}}$, which implies $\Sigma_2^P =$ NP. □

One might prefer (weakly) exact over (weakly) faithful translations. As we have seen in Section 5.2, several of our translations are not exact but only faithful. In these cases we are interested in either finding an exact translation or an evidence that an exact translation is not possible. The following theorems approve that it was appropriate to have given only a (weakly) faithful translation in Section 5.2, as there cannot be any exact such translation.

Theorem 50. *There is no (weakly) exact translation for $\sigma \Rightarrow \sigma'$ where $\sigma \in \{adm, com\}$ and $\sigma' \in \{stb, prf, sem, stg\}$.*

Proof. This is basically by the fact that admissible resp. complete extensions may be in a $\subset$-relation; consider e.g. $F = (\{a, b\}, \{(a, b), (b, a)\})$ with $\sigma(F) = \{\{a\}, \{b\}, \emptyset\}$. Let us now assume there exists a (weakly) exact translation Tr for $\sigma \Rightarrow \sigma'$. By definition, $\sigma(F) = \{\{a\}, \{b\}, \emptyset\} \subseteq \sigma'(\mathit{Tr}(F))$, but as $\emptyset \subset \{a\}$ this contradicts $\sigma' \in \{stb, prf, sem, stg\}$. □

Theorem 51. *There is no (weakly) exact translation for $com \Rightarrow adm$.*

Proof. We observe that for every AF F it holds that $\emptyset \in adm(F)$, but there are AFs where $\emptyset \notin com(F)$. Thus for a weakly exact translation Tr, with the collection $\mathcal{S}$ of remainder sets, it holds that $\emptyset \in \mathcal{S}$. But then, given an AF F with $\emptyset \in com(F)$, e.g. $F = (\{a, b\}, \{(a, b), (b, a)\})$, we can conclude that $\emptyset \in adm(\mathit{Tr}(F)) \setminus \mathcal{S}$, a contradiction. □

Theorem 52. *There is no efficient (weakly) exact translation for $grd \Rightarrow \sigma$ where $\sigma \in \{stb, adm, com\}$, unless L = P.*

Proof. Let us, towards a contradiction, assume that there exists an efficient (weakly) exact translation Tr for $grd \Rightarrow \sigma$, with the collection $\mathcal{S}$ of remainder sets. For a given AF $F = (A, R)$ with a set $E \subseteq A$ it holds that $E \in grd(F)$ iff $E \in \sigma(\mathit{Tr}(F)) \setminus \mathcal{S}$. As $E \subseteq A$ and $\mathcal{S}$ is independent of F, we have that $E \in \mathcal{S}$ implies $E = \emptyset$. Thus for $E \neq \emptyset$[1] the translation Tr would be an L-reduction from the P-hard problem Ver_{grd} (see Proposition 6) to Ver_σ ($\sigma \in \{stb, adm, com\}$) which is in L. □

In Section 5.2 we presented two translations for $stg \Rightarrow sem$: Tr_2 which is an exact translation, but not embedding, and Tr_5 which is an embedding and faithful translation, but not exact. Let us also mention at this point that Tr_2 was the only translation presented in Section 5.2 that is not embedding. Hence a natural question that occurs is whether a translation that is embedding and exact for $stg \Rightarrow sem$ is possible. We give a negative answer to this question.

[1] The empty set can be easily verified, in L, to be the grounded extension by testing that there is no unattacked argument in F.

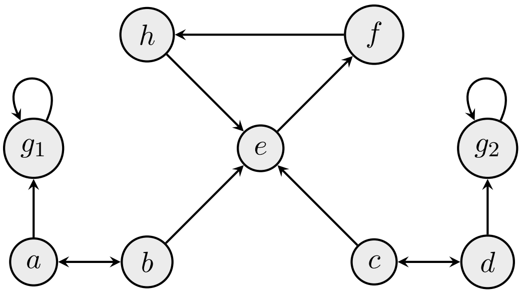


Figure 5.10: Counterexample for exact translations $\sigma \Rightarrow stg$ ($\sigma \in \{sem, prf\}$).

Theorem 53. *There is no embedding and (weakly) exact translation for stg $\Rightarrow$ sem.*

Proof. Let us assume there exists an embedding and (weakly) exact translation Tr for $stg \Rightarrow sem$. Consider the AF $F = (\{a, b\}, \{(a, a), (a, b)\})$ with $stg(F) = \{\{b\}\}$. As Tr is a (weakly) exact translation we have that $\{b\} \in sem(Tr(F))$ and thus $\{b\} \in adm(Tr(F))$. Further we have that $(a, b) \in R_{Tr(F)}$ ($Tr(F)$ is embedding) and thus $\{b\}$ must attack a. But then we have $(b, a) \in R_{Tr(F)}$ which is contradiction to Tr being an embedding translation. □

Finally we present an impossibility result for $prf \Rightarrow stg$ and $sem \Rightarrow stg$.

Theorem 54. *There is no (weakly) exact translation for $\sigma \Rightarrow stg$ ($\sigma \in \{sem, prf\}$).*

Proof. Consider the AF[2] $F = (\{a, b, c, d, e, f, g_1, g_2, h\}, \{(g_1, g_1), (g_2, g_2), (a, b), (b, a), (c, d), (d, c), (a, g_1), (b, e), (c, e), (d, g_2), (e, f), (f, h), (h, e)\})$ illustrated in Figure 5.10. We have that $sem(F) = \{\{b, d, f\}, \{a, c, f\}, \{a, d\}\}$ and $prf(F) = sem(F) \cup \{\{b, c, f\}\}$.

To prove that there is no weakly exact translation for $\sigma \Rightarrow stg$ ($\sigma \in \{sem, prf\}$), we will show that there exists no AF F' with $sem(F) \subseteq stg(F')$. To this end, let us assume that $F' = (A', R')$ is such an AF with $\{\{b, d, f\}, \{a, c, f\}, \{a, d\}\} \subseteq stg(F')$. Using the fact that $\{b, d, f\}$ is conflict-free in F' we obtain that $(d, f), (f, d) \notin R'$ and similar by using that $\{a, c, f\}$ is conflict-free in F' we get that $(a, f), (f, a) \notin R'$. By assumption $\{a, d\} \in stg(F')$ and thus $\{a, d\}$ is a maximal conflict-free set of F', but by the above observations the set $\{a, d, f\}$ is also conflict-free in F', a contradiction. □

5.4 Summary

In this chapter, we investigated intertranslations between different semantics for abstract argumentation. We focused on translations which are efficiently computable and faithful (with a few relaxations due to certain differences implicit to the semantics). An overview of our results is given in Table 5.1.[3] The entry in row σ and column σ' is to read as follows: "–" states that we

[2] The author is grateful to Christof Spanring for pointing at this counter-example.

[3] One may notice that Tr_5 does not appear in the table. Recall that Tr_5 was proposed as an alternative to Tr_2 satisfying slightly different properties for $stg \Rightarrow sem$; see also the discussion before Theorem 53.

	grd	adm	stb	com	prf	sem	stg
grd	id	$Tr_4 \circ Tr_8$ / -	Tr_8 / -	Tr_8 / -	Tr_8 / ?	Tr_8 / ?	Tr_8 / ?
adm	–	id	Tr_6 / -	Tr_1	$Tr_4 \circ Tr_6$ / -	Tr_6 / -	Tr_6 / -
stb	–	Tr_4	id	Tr_4	Tr_4	Tr_3, Tr_4	Tr_3
com	–	$Tr_4 \circ Tr_7$ / -	Tr_7 / -	id	$Tr_4 \circ Tr_7$ / -	Tr_7 / -	Tr_7 / -
prf	–	–	–	–	id	Tr_1	? / -
sem	–	–	–	–	–	id	? / -
stg	–	–	–	–	–	Tr_2	id

Table 5.1: Results about (weakly) faithful / exact translations.

have shown (Section 5.3) that no efficient faithful (even weakly faithful) translation for $\sigma \Rightarrow \sigma'$ exists. If the entry refers to a translation (or a concatenation of translations), we have found an efficient (weakly) exact translation for $\sigma \Rightarrow \sigma'$. An entry which is split into two parts, e.g. "Tr_8 / -", means that we have found an efficient (weakly) faithful translation, but there is no such exact translation. "?" indicates an open problem. We mention that all the concatenated translations are weakly faithful as they are built from a weakly exact translation Tr_4 (which has as only remainder set the empty set) and a faithful translation (either Tr_6, Tr_7, or Tr_8).

Figure 5.11 illustrates our intertranslatability results at one glance. Here, a solid arrow expresses that there is an efficient faithful translation while a dotted arrow depicts that there

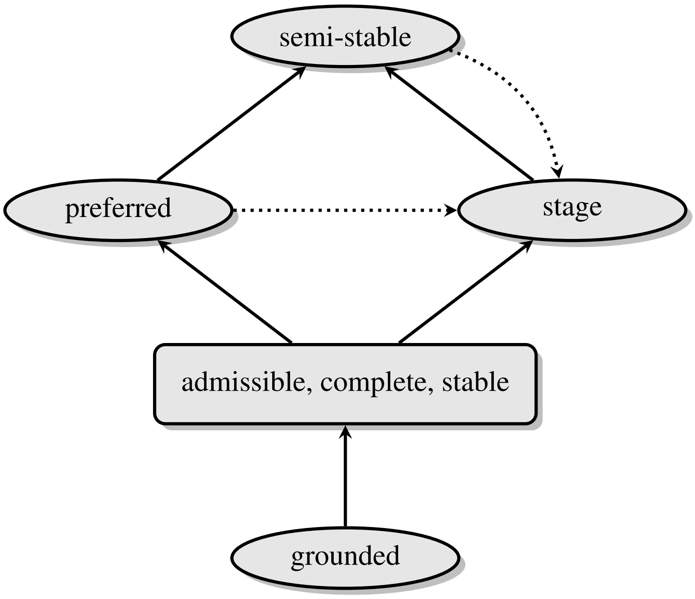


Figure 5.11: Intertranslatability of argumentation semantics w.r.t. weakly faithful translations.

may exist such a translation, but so far we have neither found one nor have an argument against its existence. Furthermore, if for two semantics σ, σ' there is no path from σ to σ' then it is proven (partly under typical complexity theoretical assumptions) that there is no efficient faithful translation for $\sigma \Rightarrow \sigma'$. If we consider the relations between the semantics w.r.t. exactness rather than just faithfulness, the overall picture changes; see Figure 5.12. Here, we get a more detailed picture about the relations between stable, admissible, and complete semantics. One conclusion, we can draw from these pictures is that semi-stable semantics is the most expressive one, since each of the other investigated semantics can be efficiently embedded. Moreover, we believe that our investigations complements recent results about comparisons between the different semantics proposed for argumentation frameworks.

Let us at this point also mention that, instead of considering different properties for the translations, we could also have used slightly revised semantics. The notion of remainder sets (as given in Definition 79) can partly be circumvented by, for instance, using a quasi-admissible semantics instead of admissible semantics, where the quasi-admissible extensions of an AF are all non-empty admissible extensions (in case such ones exist), or is only the empty set otherwise. Also it is obvious that the more restricted the properties for a translation are, the less such translations exist (compare Figures 5.11 and 5.12). Hence, we observe a certain trade-off between translation criteria and comparability between semantics.

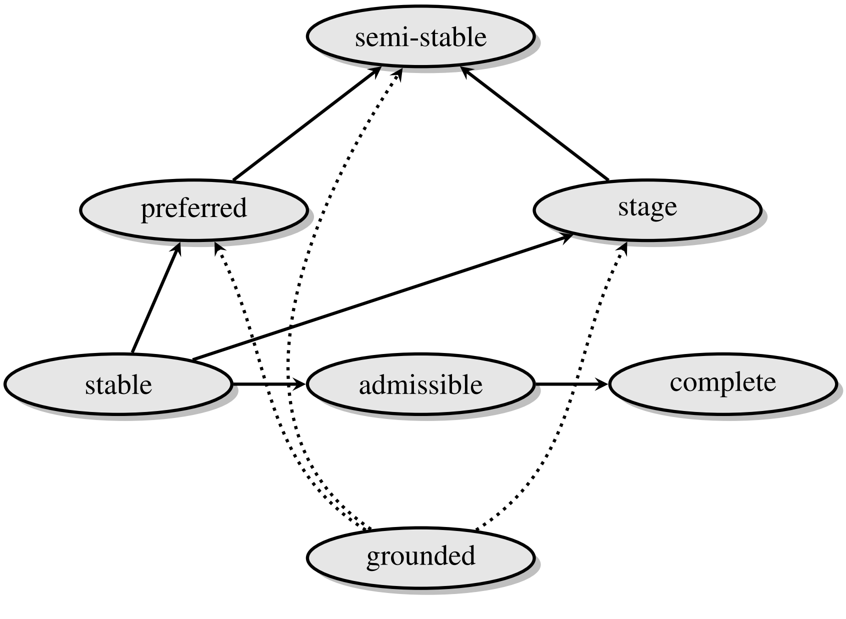


Figure 5.12: Intertranslatability of argumentation semantics w.r.t. weakly exact translations.

CHAPTER 6

Conclusion

Here we briefly summarise and discuss the achievements of this thesis, and give an outlook to potential future research directions.

6.1 Summary

In this work we provided a comprehensive analysis of computational issues in abstract argumentation, addressing computational complexity in general, several approaches to obtain tractable subclasses and finally intertranslatability as an elegant method to generalise existing systems to several semantics. In the following we discuss our contributions in more detail.

In Chapter 3 we complemented existing complexity results in several directions. Firstly, we gave exact complexity bounds for standard reasoning problems with semi-stable or stage semantics. Secondly, we provided a comprehensive complexity analysis of ideal reasoning presenting generic complexity results referring to the complexity of classical reasoning tasks as well as exact complexity classifications for the semantics under our considerations. Moreover, we classified the tractable problems w.r.t. P-completeness (resp. L-membership). Finally, together with existing results from the literature, we obtained the complexity landscape of abstract argumentation as summarised in Table 6.1.

The general picture is that there are three tractable semantics, namely cf, $naive$ and grd. We have that the problems concerning cf are all on the L level and thus cf lacks expressibility. When considering credulous and skeptical reasoning $naive$ semantics are also in L, but when considering ideal acceptance with the heavy use of self-attacks one gets P-completeness. For grounded semantics we have that the three reasoning modes coincide and acceptance is P-complete. So we have that on the one hand grounded semantics is not amenable for efficient parallelisation but on the other hand it is more expressible than cf and also than $naive$ if it is used in a reasonable way.

Further we have that stb, adm, com, $resGr$ are on the first level of the polynomial hierarchy, i.e. on the NP, coNP layer, while prf, sem, stg are located at the second level of the polynomial

σ	$Cred_\sigma$	$Skept_\sigma$	$Ideal_\sigma$	Ver_σ	$Exists_\sigma$	$Exists_\sigma^{\neg\emptyset}$
cf	**in L**	trivial	trivial	**in L**	trivial	**in L**
$naive$	**in L**	**in L**	**P-c**	**in L**	trivial	**in L**
grd	**P-c**	**P-c**	**P-c**	**P-c**	trivial	**in L**
stb	NP-c	coNP-c	$\mathbf{D}^{P}$**-c**	**in L**	NP-c	NP-c
adm	NP-c	trivial	trivial	**in L**	trivial	NP-c
com	NP-c	**P-c**	**P-c**	**in L**	trivial	NP-c
$resGr$	NP-c	coNP-c	**coNP-c**	**P-c**	trivial	in P
prf	NP-c	Π_2^P-c	in Θ_2^P	coNP-c	trivial	NP-c
sem	$\mathbf{\Sigma_2^P}$**-c**	$\mathbf{\Pi_2^P}$**-c**	$\mathbf{\Pi_2^P}$**-c**	coNP-c	trivial	NP-c
stg	$\mathbf{\Sigma_2^P}$**-c**	$\mathbf{\Pi_2^P}$**-c**	$\mathbf{\Pi_2^P}$**-c**	**coNP-c**	trivial	**in L**

Table 6.1: Complexity of abstract argumentation ($\mathcal{C}$-c denotes completeness for class $\mathcal{C}$). Novel results are highlighted in boldface.

hierarchy. Hence we have that most of the interesting problems are in general intractable while the necessity of efficient reasoners for argumentation systems is obvious.

Therefore, in Chapter 4, we explored the range of tractable instances. That is we studied so called tractable fragments, i.e. graph classes that allow for efficient reasoning methods. We complemented existing results on acyclic, even-cycle free, bipartite, and symmetric AFs by extending them to all semantics under our considerations whenever possible and showing hardness otherwise. Moreover we classified the obtained tractability results w.r.t. P-completeness. We also studied fragments beyond tractability, that is graph classes reducing the complexity of reasoning problems for the semantics at the second level of the polynomial hierarchy down to either NP or coNP. In a second approach towards tractability we studied fixed-parameter tractability, i.e. parameterisations of problems such that the problem is hard w.r.t. a parameter but can be solved in polynomial w.r.t. the size if the parameter is bounded by a fixed constant. We considered several graph parameters. First building on work of Dunne [44] we considered the parameter tree-width and extended existing results to all of our semantics and reasoning problems using characterisations in monadic second order logic and meta-theorems by Courcelle [32, 33] and Bodlaender [20]. In a similar manner we gave fixed-parameter tractability results for the more general parameter of clique-width using MSO_1 encodings and meta-theorems by Courcelle et al. [36] and Oum and Seymour [96]. Finally we studied generalisations of tree-width for directed graphs and showed that all of these parameters are not applicable to abstract argumentation in order to obtain fixed parameter tractability.

A high-level overview of tractability results for abstract argumentation, combing novel results from this work together with existing results from the literature is given in Table 6.2. We have that acyclic AFs are tractable for all semantics, while the more general class of even-cycle free AFs is tractable for all semantics except stage semantics where the reasoning tasks maintain

	stb	adm	com	$resGr$	prf	sem	stg
acyclic	✓	✓	✓	✓	✓	✓	✓
noeven	✓	✓	✓	✓	✓	✓	✗
bipartite	✓	✓	✓	✗	✓	✓	✓
symmetric	✗	✓	✓	✓	✓	✗	✗
bounded tree-width	✓	✓	✓	✓	✓	✓	✓
bounded clique-width	✓	✓	✓	✓	✓	✓	✓
bounded cycle-rank	✗	✗	✗	✗	✗	✗	✗
bounded directed path-width	✗	✗	✗	✗	✗	✗	✗
bounded Kelly-width	✗	✗	✗	✗	✗	✗	✗
bounded DAG-width	✗	✗	✗	✗	✗	✗	✗
bounded directed tree-width	✗	✗	✗	✗	✗	✗	✗

Table 6.2: Tractability for abstract argumentation: An entry ✓means the main reasoning problems are tractable for AFs of the given graph class. While an entry ✗ mains that at least one of the main reasoning problem remains hard for such AF.

their full complexity on the second level of the polynomial hierarchy. For bipartite AFs the only exception from tractability is $resGr$ where credulous acceptance remains hard. Notice also that by results in [12, 44] deciding whether two (or more) arguments are simultaneously credulously accepted in a bipartite AF is NP-hard. For symmetric AFs we have that stable, semi-stable and stage semantics remain hard, but when additionally assuming that the AFs are irreflexive also these semantics are tractable [31]. For the fragments of acyclic AFs and AFs without even length cycles one can compute the unique extension efficiently (except for admissible semantics). In contrast in bipartite and symmetric AFs there is in general a exponential number of extensions and thus they can not be handled efficiently.

Concerning fixed-parameter tractability we have that all semantics under our considerations are tractable w.r.t. graph parameters tree-width and clique-width. This was shown by MSO_1 characterisations of the semantics and extends to any new semantics that is expressible in MSO_1. Moreover we have shown that the graph parameter cycle-rank does not lead to tractability which, by results from the literature [16, 77, 78], also implies intractability for the parameters directed path-width, Kelly-width, DAG-width, and directed tree-width.

However, notice that, beside the fragments studied here, there are further negative results for preferred semantics given in [44], i.e. for planar graphs and parameters vertex degree and k-partite, as well as there is work towards so called backdoors for tractable fragments by Ordyniak and Szeider [95], providing both fixed-parameter tractability and negative results. Roughly speaking the idea behind the backdoor approach is to parameterise the distance to a tractable fragments.

Finally we studied how semantics can be translated into each other by modifying the AF,

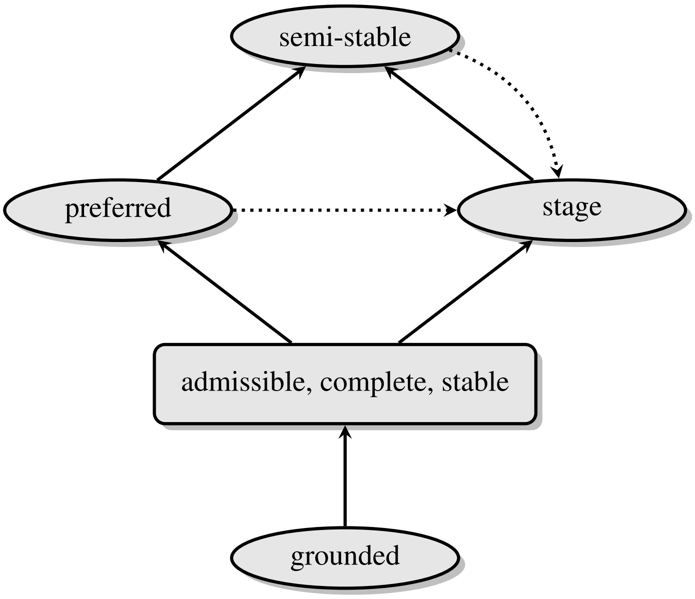


Figure 6.1: Intertranslatability of argumentation semantics.

which gives rise to reduction-based implementations. That is, given such a translation from a semantics σ to a semantics σ', we can use an existing solver for σ' to compute the σ-extensions by first applying the translation. The overall picture of our results is given in Figure 6.1. Semantics embraced in one box can be translated to each other, that is admissible, complete and stable semantics can be translated to each other. A path of solid arrows from a semantic σ to a semantics σ' expresses that σ can be translated to σ'. Dotted arrow (and paths containing a dotted arrow) depicts that there may exist such a translation, but so far we have neither found one nor have an argument against its existence. Furthermore, if for two semantics σ, σ' there is no path from σ to σ' then we have shown (partly under typical complexity theoretical assumptions) that there is no efficient translation for $\sigma \Rightarrow \sigma'$.

6.2 Open Problems & Future Research Directions

We already identified several open problems during the last chapters which we briefly summarise next. In Chapter 3 we have to leave the exact complexity of the problems $Ideal_{prf}$ and $Exists^{\neg\emptyset}_{resGr}$ open. However the first can be shown to be Θ_2^P-complete under randomized reductions [44]. In Chapter 4 we mentioned that it is open whether $Cred_{resGr}$ restricted to symmetric AFs is P-complete. In our studies on intertranslatability in Chapter 5 we left open whether there exist efficient weakly faithful translations for $prf \Rightarrow stg$ and $sem \Rightarrow stg$ and whether there exist efficient weakly exact translations for $grd \Rightarrow \{prf, sem, stg\}$. However these problems are all of rather technical nature, and will only have minor effects. In the following we discuss open issues on a higher level, which are crucial towards efficient reasoning systems for practical argumentation.

First, recall that we have discussed a lot of graph parameters for obtaining fixed-parameter tractability results for argumentation. However such results only help if instances in practise provide low values for the parameter. So one important issue is to actually collecting benchmarks argumentation frameworks of applications, study their structure and identify parameters that apply to them (and offer fixed-parameter tractability).

Also without having information about the typical structure argumentation frameworks provide one can think of building complexity-sensitive reasoning systems building on our complexity results. For instance consider semi-stable semantics which in general is on the second level of the polynomial hierarchy. A good reasoner should work in polynomial time for acyclic AFs, should be in NP is the AF is free of odd cycles, i.e. it might use a SAT solver, and only if none of the easier fragments apply using a general procedure. First work on complexity-sensitive procedures was recently published in [62].

Finally let us mention that abstract argumentation frameworks are not the holy grail of formal argumentation. While they convince by their simplicity when studying conflicts between arguments, even this simplicity sometimes causes cumbersome encodings in the instantiation process, for example when dealing with preference. Hence several generalisations of Dung's abstract argumentation frameworks have been proposed, most prominently value-based argumentation frameworks (VAFs) [13], extend argumentation frameworks (EAFs) [91], argumentation frameworks with recursive attacks (AFRAs) [11], and abstract dialectical frameworks (ADFs) [21]. Clearly similar computational issues as for standard AFs also arise for each of these generalisations. So one open issue would be to complement existing complexity analysis on these generalisations, in particular by studies on tractable sub-classes. The general complexity of VAFs was studied [15, 49] as well as there are investigations for fixed-parameter tractability [44, 84]. For EAFs the general complexity has been studied in [52] and the complexity of AFRAs is implicitly given in [11] by a translation to abstract argumentation frameworks. However as far as we know there is no work on tractable subclasses for EAFs and AFRAs. For ADFs even the general complexity of reasoning is partly still open.

For future work on intertranslatability, we identify the following tasks: Further properties for translations could be of interest. For instance, one could even strengthen the property of being exact (which is defined in terms of the extensions) to the requirement that the labelings [27] of the source and target framework coincide. Labelings provide additional information, in particular for arguments not contained in an extension. A promising starting point for obtaining (negative) results in that direction would be the work on labeling based justification statuses of arguments [53, 106]. Likewise, it would be interesting to investigate intertranslatability in the more general approach of equational semantics for argumentation frameworks [71].

Further properties for translations could be also given in terms of graph properties. As an example, acyclic AFs should remain acyclic after the translations, or parameters as tree-width should remain unchanged. Requirements of such a form are also termed "structural preservation" [80]. Such properties are of interest from a computational point of view in the sense that, in case the source AF is easy to evaluate (because of its structure), this advantage should not be lost during the translation or if we translate a semantics on the first level of the polynomial hierarchy to a semantics on the second level of the polynomial hierarchy we gain for a graph structure that allows for NP (resp. coNP) algorithms; recall here Figure 5.1 where we suggested to use our translations for a rapid prototyping approach to compute the extensions of a semantics via an argumentation engine based on a different semantics.

Another open issue is to studying translations between semantics for generalisations of Dung-style AFs as EAFs, AFRAs or ADFs.

Bibliography

[1] Leila Amgoud and Caroline Devred. Argumentation frameworks as constraint satisfaction problems. In Salem Benferhat and John Grant, editors, *Scalable Uncertainty Management - 5th International Conference, SUM 2011, Dayton, OH, USA, October 10-13, 2011. Proceedings*, volume 6929 of *Lecture Notes in Computer Science*, pages 110–122. Springer, 2011.

[2] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Algebraic Discrete Methods*, 8:277–284, April 1987.

[3] Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991.

[4] Jørgen Bang-Jensen and Gregory Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer Monographs in Mathematics. Springer, 2010.

[5] János Barát. Directed path-width and monotonicity in digraph searching. *Graphs and Combinatorics*, 22(2):161–172, 2006.

[6] Pietro Baroni and Massimiliano Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artif. Intell.*, 171(10-15):675–700, 2007.

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

[8] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. Scc-recursiveness: A general schema for argumentation semantics. *Artif. Intell.*, 168(1-2):162–210, 2005.

[9] Pietro Baroni, Paul E. Dunne, and Massimiliano Giacomin. On extension counting problems in argumentation frameworks. In Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Guillermo R. Simari, editors, *Proceedings of the 3th Conference on Computational Models of Argument (COMMA 2010)*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 63–74, Desenzano del Garda, Italy, September 8-10 2010. IOS Press.

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

[11] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. AFRA: Argumentation framework with recursive attacks. *Int. J. Approx. Reasoning*, 52(1):19–37, 2011.

[12] 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.

[13] Trevor J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *J. Log. Comput.*, 13(3):429–448, 2003.

[14] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.

[15] Trevor J. M. Bench-Capon, Sylvie Doutre, and Paul E. Dunne. Audiences in argumentation frameworks. *Artif. Intell.*, 171(1):42–71, 2007.

[16] Dietmar Berwanger, Anuj Dawar, Paul Hunter, and Stephan Kreutzer. Dag-width and parity games. In Bruno Durand and Wolfgang Thomas, editors, *STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science, Marseille, France, February 23-25, 2006, Proceedings*, volume 3884 of *Lecture Notes in Computer Science*, pages 524–536. Springer, 2006.

[17] 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.

[18] Hans L. Bodlaender. On linear time minor tests with depth-first search. *J. Algorithms*, 14 (1):1–23, 1993.

[19] Hans L. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11:1–21, 1993.

[20] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[21] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*, pages 780–785. AAAI Press, 2010.

[22] Martin Caminada. Semi-stable semantics. In Paul E. Dunne and Trevor J. M. Bench-Capon, editors, *Proceedings of the 1st Conference on Computational Models of Argument (COMMA 2006)*, volume 144 of *Frontiers in Artificial Intelligence and Applications*, pages 121–130. IOS Press, 2006.

[23] Martin Caminada. On the issue of reinstatement in argumentation. In Michael Fisher, Wiebe van der Hoek, Boris Konev, and Alexei Lisitsa, editors, *Logics in Artificial Intelligence, 10th European Conference, JELIA 2006, Liverpool, UK, September 13-15,*

2006, Proceedings, volume 4160 of *Lecture Notes in Computer Science*, pages 111–123. Springer, 2006.

[24] 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.

[25] Martin Caminada. An algorithm for stage semantics. In Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Guillermo Ricardo Simari, editors, *Computational Models of Argument: Proceedings of COMMA 2010, Desenzano del Garda, Italy, September 8-10, 2010*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 147–158. IOS Press, 2010.

[26] Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.

[27] Martin Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2):109–145, 2009.

[28] Martin Caminada, Walter A. Carnielli, and Paul E. Dunne. Semi-stable semantics. *Journal of Logic and Computation*, 2011.

[29] 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.

[30] Derek G. Corneil and Udi Rotics. On the relationship between clique-width and treewidth. *SIAM J. Comput.*, 34(4):825–847, 2005.

[31] 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.

[32] Bruno Courcelle. Recognizability and second-order definability for sets of finite graphs. Technical Report I-8634, Université de Bordeaux, 1987.

[33] Bruno Courcelle. Graph rewriting: an algebraic and logic approach. In *Handbook of Theoretical Computer Science, Vol. B*, pages 193–242. Elsevier, Amsterdam, 1990.

[34] Bruno Courcelle and Joost Engelfriet. *Graph structure and monadic second-order logic, a language theoretic approach*, volume 2012. Cambridge University Press, 2011.

[35] Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.

[36] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

[37] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discrete Applied Mathematics*, 108(1-2):23–52, 2001.

[38] Jos de Bruijn, Thomas Eiter, and Hans Tompits. Embedding approaches to combining rules and ontologies into autoepistemic logic. In Gerhard Brewka and Jérôme Lang, editors, *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR2008)*, pages 485–495. AAAI Press, 2008.

[39] Marc Denecker, Wiktor Marek, and Miroslaw Truszczyński. Uniform semantic treatment of default and autoepistemic logics. *Artif. Intell.*, 143(1):79–122, 2003.

[40] Artan Dermaku, Tobias Ganzow, Georg Gottlob, Benjamin J. McMahan, Nysret Musliu, and Marko Samer. Heuristic methods for hypertree decomposition. In Alexander F. Gelbukh and Eduardo F. Morales, editors, *MICAI 2008: Advances in Artificial Intelligence, 7th Mexican International Conference on Artificial Intelligence, Atizapán de Zaragoza, Mexico, October 27-31, 2008, Proceedings*, volume 5317 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 2008.

[41] Yannis Dimopoulos and Alberto Torres. Graph theoretical structures in logic programs and default theories. *Theor. Comput. Sci.*, 170(1-2):209–244, 1996.

[42] 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):321–358, 1995.

[43] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. *Artif. Intell.*, 171(10-15):642–674, 2007.

[44] Paul E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artif. Intell.*, 171(10-15):701–729, 2007.

[45] Paul E. Dunne. The computational complexity of ideal semantics I: Abstract argumentation frameworks. In *Proceedings of the 2nd Conference on Computational Models of Argument COMMA'08*, volume 172 of *Frontiers in Artificial Intelligence and Applications*, pages 147–158. IOS Press, 2008.

[46] Paul E. Dunne. The computational complexity of ideal semantics. *Artif. Intell.*, 173(18): 1559–1591, 2009.

[47] Paul E. Dunne and Trevor J. M. Bench-Capon. Complexity and combinatorial properties of argument systems. Technical report, Dept. of Computer Science, University of Liverpool, 2001.

[48] Paul E. Dunne and Trevor J. M. Bench-Capon. Coherence in finite argument systems. *Artif. Intell.*, 141(1/2):187–203, 2002.

[49] Paul E. Dunne and Trevor J. M. Bench-Capon. Complexity in value-based argument systems. In José Júlio Alferes and João Alexandre Leite, editors, *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA 2004)*, volume 3229 of *Lecture Notes in Computer Science*, pages 360–371. Springer, 2004.

[50] Paul E. Dunne and Martin Caminada. Computational complexity of semi-stable semantics in abstract argumentation frameworks. In Steffen Hölldobler, Carsten Lutz, and Heinrich Wansing, editors, *Proceedings of the 11th European Conference on Logics in Artificial Intelligence JELIA 2008*, volume 5293 of *Lecture Notes in Computer Science*, pages 153–165. Springer, 2008.

[51] Paul E. Dunne and Michael Wooldridge. Complexity of abstract argumentation. In Guillermo Simari and Iyad Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 85–104. Springer US, 2009.

[52] Paul E. Dunne, Sanjay Modgil, and Trevor J. M. Bench-Capon. Computation in extended argumentation frameworks. In *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 119–124. IOS Press, 2010.

[53] Wolfgang Dvořák. On the complexity of computing the justification status of an argument. In *First International Workshop on the Theory and Applications of Formal Argumentation (TAFA-11)*, Barcelona, Catalonia, Spain,, July 16-17 2011.

[54] Wolfgang Dvořák and Stefan Woltran. Complexity of semi-stable and stage semantics in argumentation frameworks. *Inf. Process. Lett.*, 110(11):425–430, 2010.

[55] Wolfgang Dvořák and Stefan Woltran. On the intertranslatability of argumentation semantics. In *Proceedings of the Conference on Thirty Years of Nonmonotonic Reasoning (NonMon@30)*, Lexington, KY, USA, October 22-25 2010.

[56] Wolfgang Dvořák and Stefan Woltran. On the intertranslatability of argumentation semantics. *J. Artif. Intell. Res. (JAIR)*, 41:445–475, 2011.

[57] Wolfgang Dvořák, Reinhard Pichler, and Stefan Woltran. Towards fixed-parameter tractable algorithms for argumentation. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR'10)*, pages 112–122. AAAI Press, 2010.

[58] Wolfgang Dvořák, Stefan Szeider, and Stefan Woltran. Reasoning in argumentation frameworks of bounded clique-width. In Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Guillermo R. Simari, editors, *Proceedings of the 3th Conference on Computational Models of Argument (COMMA 2010)*, Frontiers in Artificial Intelligence and Applications, pages 219–230, Desenzano del Garda, Italy, September 8-10 2010. IOS Press.

[59] Wolfgang Dvořák, Paul E. Dunne, and Stefan Woltran. Parametric properties of ideal semantics. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 851–856. IJCAI/AAAI, 2011.

[60] Wolfgang Dvořák, Michael Morak, Clemens Nopp, and Stefan Woltran. dynpartix - a dynamic programming reasoner for abstract argumentation. *CoRR*, abs/1108.4804, 2011.

[61] Wolfgang Dvořák, Reinhard Pichler, and Stefan Woltran. Towards fixed-parameter tractable algorithms for abstract argumentation. Technical Report DBAI-TR-2011-74, Technische Universität Wien, 2011.

[62] Wolfgang Dvořák, Matti Järvisalo, Johannes Wallner, and Stefan Woltran. Complexity-sensitive decision procedures for abstract argumentation. Accepted for KR'12, 2012.

[63] Lawrence C. Eggan. Transition graphs and the star height of regular events. *Michigan Math. J.*, 10:385–397, 1963.

[64] Uwe Egly and Stefan Woltran. Reasoning in argumentation frameworks using quantified boolean formulas. In Paul E. Dunne and Trevor J. M. Bench-Capon, editors, *Proceedings of the 1st Conference on Computational Models of Argument (COMMA 2006)*, volume 144 of *Frontiers in Artificial Intelligence and Applications*, pages 133–144. IOS Press, 2006.

[65] Uwe Egly, Sarah Gaggl, and Stefan Woltran. Answer-set programming encodings for argumentation frameworks. *Argument and Computation*, 1(2):147–177, 2010.

[66] Thomas Eiter and Georg Gottlob. Propositional circumscription and extended closed-world reasoning are Π_2^P-complete. *Theor. Comput. Sci.*, 114(2):231–245, 1993.

[67] Thomas Eiter and Georg Gottlob. The complexity class Θ_2^{p}: Recent results and applications in AI and modal logic. In Bogdan S. Chlebus and Ludwik Czaja, editors, *Fundamentals of Computation Theory, 11th International Symposium, FCT '97, Kraków, Poland, September 1-3, 1997, Proceedings*, volume 1279 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 1997.

[68] Michael R. Fellows, Frances A. Rosamond, Udi Rotics, and Stefan Szeider. Clique-width is NP-complete. *SIAM J. Discrete Math.*, 23(2):909–939, 2009.

[69] Jörg Flum and Martin Grohe. *Parameterized complexity theory*. Texts in theoretical computer science. Springer, 2006.

[70] Jörg Flum, Markus Frick, and Martin Grohe. Query evaluation via tree-decompositions. *J. ACM*, 49(6):716–752, 2002.

[71] Dov M. Gabbay. Equational approach to argumentation networks. Unpublished draft, 2011.

[72] Sarah Alice Gaggl and Stefan Woltran. Strong equivalence for argumentation semantics based on conflict-free sets. In Weiru Liu, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 11th European Conference, ECSQARU 2011, Belfast, UK, June 29-July 1, 2011. Proceedings*, volume 6717 of *Lecture Notes in Computer Science*, pages 38–49. Springer, 2011.

[73] Leslie M. Goldschlager. The monotone and planar circuit value problems are log space complete for P. *ACM SIGACT News*, 9(2):25–29, 1977.

[74] Georg Gottlob. Translating default logic into standard autoepistemic logic. *J. ACM*, 42 (4):711–740, 1995.

[75] Guido Governatori, Michael J. Maher, Grigoris Antoniou, and David Billington. Argumentation semantics for defeasible logic. *J. Log. Comput.*, 14(5):675–702, 2004.

[76] Raymond Greenlaw, H.James Hoover, and Walter L. Ruzzo. *Limits to parallel computation: P-completeness theory*. Oxford University Press, 1995.

[77] Hermann Gruber. Digraph complexity measures and applications in formal language theory. In *Proceedings of the 4th Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS 2008)*, pages 60–67, 2008.

[78] Paul Hunter and Stephan Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theor. Comput. Sci.*, 399(3):206–219, 2008.

[79] Tomi Janhunen. On the intertranslatability of non-monotonic logics. *Ann. Math. Artif. Intell.*, 27(1-4):79–128, 1999.

[80] Tomi Janhunen, Ilkka Niemelä, Dietmar Seipel, Patrik Simons, and Jia-Huai You. Unfolding partiality and disjunctions in stable model semantics. *ACM Trans. Comput. Log.*, 7(1):1–37, 2006.

[81] Thor Johnson, Neil Robertson, Paul D. Seymour, and Robin Thomas. Directed tree-width. *J. Comb. Theory, Ser. B*, 82(1):138–154, 2001.

[82] Mamadou Moustapha Kanté. The rank-width of directed graphs. *CoRR*, abs/0709.1433, 2007.

[83] Simon Kasif. On the parallel complexity of some constraint satisfaction problems. In Tom Kehler, editor, *Proceedings of the 5th National Conference on Artificial Intelligence. Philadelphia, PA, August 11-15, 1986. Volume 1: Science*, pages 349–353. Morgan Kaufmann, 1986.

[84] Eun Jung Kim, Sebastian Ordyniak, and Stefan Szeider. Algorithms and complexity results for persuasive argumentation. *Artif. Intell.*, 175(9-10):1722–1736, 2011.

[85] Joachim Kneis, Alexander Langer, and Peter Rossmanith. Courcelle's theorem - a game-theoretic approach. *Discrete Optimization*, 8(4):568–594, 2011.

[86] Kurt Konolige. On the relation between default and autoepistemic logic. *Artif. Intell.*, 35 (3):343–382, 1988.

[87] Richard E. Ladner. The circuit value problem is log space complete for P. *SIGACT News*, 7:18–20, January 1975.

[88] Alexander Langer, Felix Reidl, Peter Rossmanith, and Somnath Sikdar. Evaluation of an MSO-solver. In *ALENEX 2012 (accepted for publication)*, 2012.

[89] Paolo Liberatore. Bijective faithful translations among default logics. *CoRR*, abs/0707.3781, 2007.

[90] Wiktor Marek and Miroslaw Truszczyński. *Nonmonotonic Logic: Context Dependent Reasoning*. Springer, 1993.

[91] Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artif. Intell.*, 173(9-10):901–934, 2009.

[92] Sanjay Modgil and Martin Caminada. Proof theories and algorithms for abstract argumentation frameworks. In Iyad Rahwan and Guillermo Simari, editors, *Argumentation in Artificial Intelligence*, pages 105–132. 2009.

[93] Rolf Niedermeier. *Invitation to fixed-parameter algorithms*, volume 31. Oxford University Press, USA, 2006.

[94] Emilia Oikarinen and Stefan Woltran. Characterizing strong equivalence for argumentation frameworks. *Artif. Intell.*, 175(14-15):1985–2009, 2011.

[95] Sebastian Ordyniak and Stefan Szeider. Augmenting tractable fragments of abstract argumentation. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 1033–1038. IJCAI/AAAI, 2011.

[96] Sang-il Oum and Paul D. Seymour. Approximating clique-width and branch-width. *J. Comb. Theory, Ser. B*, 96(4):514–528, 2006.

[97] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, Massachusetts, 1994.

[98] Reinhard Pichler, Stefan Rümmele, and Stefan Woltran. Counting and enumeration problems with bounded treewidth. In Edmund M. Clarke and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning - 16th International Conference, LPAR-16, Dakar, Senegal, April 25-May 1, 2010, Revised Selected Papers*, volume 6355 of *Lecture Notes in Computer Science*, pages 387–404. Springer, 2010.

[99] Henry Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

[100] Raymond Reiter. On closed world data bases. In *Logic and Data Bases*, pages 55–76, 1977.

[101] Neil Robertson and Paul D. Seymour. Graph minors. III. Planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984.

[102] Alan Mathison Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 2(42):230–265, 1936.

[103] Bart Verheij. Two approaches to dialectical argumentation: admissible sets and argumentation stages. In J. Meyer and L. van der Gaag, editors, *Proceedings of the 8th Dutch Conference on Artificial Intelligence (NAIC'96)*, pages 357–368, 1996.

[104] Bart Verheij. A labeling approach to the computation of credulous acceptance in argumentation. In Manuela M. Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 623–628, 2007.

[105] Yining Wu and Martin Caminada. A labelling-based justification status of arguments. *Studies in Logic*, 3(4):12–29, 2010.

[106] Yining Wu, Martin Caminada, and Mikolaj Podlaszewski. A labelling based justification status of arguments. In *13th International Workshop on Nonmonotonic Reasoning (NMR)*, 2010.

APPENDIX A

Curriculum Vitae

Wolfgang Dvořák

Vienna University of Technology
Institute of Information Systems
Database and Artificial Intelligence Group
Favoritenstraße 9
A-1040 Wien, Austria

Phone:	+43 (1) 58801 58437
Fax:	+43 (1) 58801 18493
Email:	dvorak@dbai.tuwien.ac.at
Homepage:	www.dbai.tuwien.ac.at/staff/dvorak

Personal

Date of birth: June 24, 1984

Birthplace: Vienna, Austria

Citizenship: Austrian

Gender: male

Education

Student of Technical Mathematics, Vienna University of Technology, October 2003 - March 2009,
With a concentration in Mathematical Computer Science

24.March 2009: Master of Science (Msc) in Technical Mathematics with distinction.

Student of Computational Intelligence, Vienna University of Technology, April 2009 - December 2009

02.December 2009: Master of Science (Msc) in Computational Intelligence with distinction.

Ph.D. student in Computer Science, Vienna University of Technology, April 2009 - Present
Supervisor: Ass.Prof.Dr. Stefan Woltran

Participation in Summer Schools

European Summer School in Logic, Language and Information (ESSLLI'09), Bordeaux France, July 20-31, 2009.

Advanced Course in Artificial Intelligence (ACAI'09), Belfast GB, August 23-29, 2009.

European Summer School in Logic, Language and Information (ESSLLI'10), Copenhagen Denmark, August 9-20, 2010.

Employment

Teaching Assistant at the Database and AI Group, Vienna UT, March 2006 - February 2009
for the courses: Data Modeling, Database Systems, Semistructured Data, Database Theory

Research Assistant at the Database and AI Group, Vienna University of Technology, April 2009 - Present
Project title: *New Methods for Analyzing, Comparing, and Solving Argumentation Problems*
(supported by the Vienna Science and Technology Fund (WWTF) under grant ICT 08-028)
Project leader: Ass.Prof.Dr. Stefan Woltran

Publications

Journal Articles

[J2] Wolfgang Dvořák, Stefan Woltran. On the Intertranslatability of Argumentation Semantics. Journal of Artificial Intelligence Research, Volume 41 (2011), Pages 445-475.

[J1] Wolfgang Dvořák, Stefan Woltran. Complexity of Semi-Stable and Stage Semantics in Argumentation Frameworks, Information Processing Letters, Volume 110, Issue 11, 16 May 2010, Pages 425-430

Conference Papers and Posters

[C10] Wolfgang Dvořák, Matti Järvisalo, Johannes Wallner and Stefan Woltran. Complexity-Sensitive Decision Procedures for Abstract Argumentation, Accepted for KR'12

[C9] Wolfgang Dvořák, Sarah Alice Gaggl, Johannes Wallner and Stefan Woltran. Making Use of Advances in Answer-Set Programming for Abstract Argumentation Systems. In Proceedings of the 19th International Conference on Applications of Declarative Programming and Knowledge Management (INAP 2011)

[C8] Wolfgang Dvořák, Michael Morak, Clemens Nopp, and Stefan Woltran. dynPARTIX - A Dynamic Programming Reasoner for Abstract Argumentation. In Proceedings of the 19th International Conference on Applications of Declarative Programming and Knowledge Management (INAP 2011)

[C7] Wolfgang Dvořák. On the Complexity of Computing the Justification Status of an Argument. First International Workshop on the Theory and Applications of Formal Argumentation (TAFA)

[C6] Wolfgang Dvořák, Paul E. Dunne, Stefan Woltran. Parametric Properties of Ideal Semantics. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011), 851-856 , 2011.

[C5] Wolfgang Dvořák, Stefan Woltran. On the Intertranslatability of Argumentation Semantics, In Proceedings of the Conference on Thirty Years of Nonmonotonic Reasoning(NONMON@30), 2010

[C4] Wolfgang Dvořák, Stefan Szeider, Stefan Woltran. Reasoning in Argumentation Frameworks of Bounded Clique-Width, In Proceedings of COMMA 2010, Third International Conference on Computational Models of Argument, 2010, pp. 219-230

[C3] Wolfgang Dvořák, Reinhard Pichler, Stefan Woltran. Towards Fixed-Parameter Tractable Algorithms for Argumentation, In Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR'10), 2010, pp. 112-122

[C2] Wolfgang Dvořák - Poster: Argumentation with Bounded Tree-Width, ACAI'09 (Belfast) Poster Session

[C1] Wolfgang Dvořák, Georg Gottlob, Reinhard Pichler, Stefan Woltran. Alternation as a programming paradigm, In Proceedings of the 11th ACM SIGPLAN Conference on Principles and Practice of Declarative Programming (Coimbra, Portugal, September 07 - 09, 2009). PPDP '09. ACM, New York, NY, 61-72

Technical Reports

[R5] Wolfgang Dvořák. Technical Note: Exploring Σ_2^P / Π_2^P -hardness for Argumentation Problems with fixed distance to tractable classes, CoRR, 2012, abs/1201.0478

[R4] Wolfgang Dvořák, Reinhard Pichler, Stefan Woltran. Towards Fixed-Parameter Tractable Algorithms for Abstract Argumentation, Technical Report DBAI-TR-2008-74, Technische Universität Wien, Database and Artificial Intelligence Group, 2011.

[R3] Wolfgang Dvořák, Sarah Gaggl, Johannes Wallner, and Stefan Woltran. Making Use of Advances in Answer-Set Programming for Abstract Argumentation Systems. Technical Report DBAI-TR-2011-70, Technische Universität Wien, 2011.

[R2] Wolfgang Dvořák, Stefan Woltran. Technical Note: Complexity of Stage Semantics in Argumentation Frameworks, Technical Report DBAI-TR-2009-66, Technische Universität Wien, Database and Artificial Intelligence Group, 2009

[R1] Wolfgang Dvořák, Georg Gottlob, Reinhard Pichler, Stefan Woltran. Alternation as a Programming Paradigm, Technical Report DBAI-TR-2008-64, Technische Universität Wien, Database and Artificial Intelligence Group, 2009

Other Scientific Activities

Reviewing

PC-member: IJCAI 2011

Reviewer: Artificial Intelligence Journal, ECAI 2010, LPNMR 2011, ECSQARU 2011, AAAI 2011, ICAART 2012, FOIKS 2012

Scientific Talks

[T9] *Parameterized Splitting - A Simple Modification-Based Approach* - Argumentation Seminar (University Bratislava), 30th Jan. 2012

[T8] *dynPARTIX - A Dynamic Programming Reasoner for Abstract Argumentation* - INAP'11 28th Sept. 2011

[T7] *Parametric Properties of Ideal Semantics* - IJCAI'11 22th July 2011

[T6] *On the Complexity of Computing the Justification Status of an Argument* - TAFA'11 17th July 2011

[T5] *On the Intertranslatability of Argumentation Semantics* - London Argumentation Forum (LAF), 25th March 2011

[T4] *On the Intertranslatability of Argumentation Semantics* - Argumentation Christmas Meeting Vienna, 7th Dec. 2010

[T3] *Reasoning in Argumentation Frameworks of Bounded Clique-Width* - COMMA'10, 10th Sept. 2010

[T2] *Towards Fixed-Parameter Tractable Algorithms for Argumentation* - KR'10, 12th May 2010

[T1] *Alternation as a programming paradigm* - PPDP'09, 7th Sept. 2009

Teaching

Course “Abstract Argumentation” at the Vienna University of Technology (summer term 2011, winter term 2011/2012)

Co-supervised master theses:

Christof Spanring - Intertranslatability Results for Argumentation Semantics;
Günther Charwat - Tree-Decomposition based Algorithms for Abstract Argumentation.

Grants

IJCAI Travel Grant for attending IJCAI 2011

COST Travel Award for attending the London Argumentation Forum (LAF) in March 2011

ECCAI Travel Award for attending ACAI 2009

Last updated: February 16, 2012
`http://www.dbai.tuwien.ac.at/staff/dvorak/`