Analyzing the Computational Complexity of Abstract Dialectical Frameworks via Approximation Fixpoint Theory

Hannes Strass
Computer Science Institute
Leipzig University, Germany

Johannes Peter Wallner
Institute of Information Systems
Vienna University of Technology, Austria

Abstract

Abstract dialectical frameworks (ADFs) have recently been proposed as a versatile generalization of Dung's abstract argumentation frameworks (AFs). In this paper, we present a comprehensive analysis of the computational complexity of ADFs. Our results show that while ADFs are one level up in the polynomial hierarchy compared to AFs, there is a useful subclass of ADFs which is as complex as AFs while arguably offering more modeling capacities. As a technical vehicle, we employ the approximation fixpoint theory of Denecker, Marek and Truszczyński, thus showing that it is also a useful tool for complexity analysis of operator-based semantics.

Introduction

Formal models of argumentation are increasingly being recognized as viable tools in knowledge representation and reasoning (Bench-Capon and Dunne 2007). A particularly popular formalism are Dung's abstract argumentation frameworks (AFs) (1995). AFs treat arguments as abstract entities and natively represent only attacks between them using a binary relation. Typically, abstract argumentation frameworks are used as a target language for translations from more concrete languages. For example, the Carneades formalism for structured argumentation (Gordon, Prakken, and Walton 2007) has been translated to AFs (Van Gijzel and Prakken 2011); Caminada and Amgoud (2007) and Wyner et al. (2013) translate rule-based defeasible theories into AFs. Despite their popularity, abstract argumentation frameworks have limitations. Most significantly, their limited modeling capacities are a notable obstacle for applications: arguments can only attack one another. Furthermore, Caminada and Amgoud (2007) observed how AFs that arise as translations of defeasible theories sometimes lead to unintuitive conclusions.

To address the limitations of abstract argumentation frameworks, researchers have proposed quite a number of generalizations of AFs (Brewka, Polberg, and Woltran 2013). Among the most general of those are Brewka and Woltran's *abstract dialectical frameworks (ADFs)* (2010). ADFs are even more abstract than AFs: while in AFs arguments are abstract and the relation between arguments is fixed to attack, in ADFs also the relations are abstract (and called *links*). The relationship between different arguments (called *statements* in ADFs) is specified by *acceptance conditions*. These are Boolean functions indicating the conditions under which a statement s can be accepted when given the acceptance status of all statements with a direct link to s (its *parents*). ADFs have been successfully employed to address the shortcomings of AFs: Brewka and Gordon (2010) translated Carneades to ADFs and for the first time allowed cyclic dependencies amongst arguments; for rule-based defeasible theories we (Strass 2013b) showed how to deal with the problems observed by Caminada and Amgoud (2007).

There is a great number of semantics for AFs already, and many of them have been generalized to ADFs. Thus it might not be clear to potential ADF users which semantics are adequate for a particular application domain. In this regard, knowing the computational complexity of semantics can be a valuable guide. However, existing complexity results for ADFs are scattered over different papers, miss several semantics and some of them present upper bounds only. In this paper, we provide a comprehensive complexity analysis for ADFs. In line with the literature, we represent acceptance conditions by propositional formulas as they provide a compact and elegant way to represent Boolean functions.

Technically, we base our complexity analysis on the approximation fixpoint theory (AFT) by Denecker, Marek and Truszczyński (2000; 2003; 2004). This powerful framework provides an algebraic account of how monotone and non-monotone two-valued operators can be approximated by monotone three- or four-valued operators. (As an example of an operator to be approximated, think of the two-valued van Emden-Kowalski consequence operator from logic programming.) AFT embodies the intuitions of decades of KR research; we believe that this is very valuable also for relatively recent languages (such as ADFs), because we get the enormously influential formalizations of intuitions of Reiter and others for free. (As a liberal variation on Newton, we could say that approximation fixpoint theory allows us to take the elevator up to the shoulders of giants instead of walking up the stairs.) In fact, approximation fixpoint theory can be and partially has already been used to define some of the semantics of ADFs (Brewka et al. 2013; Strass 2013a). There, we generalized various AF and logic programming semantics to ADFs using AFT, which has provided us with two families of semantics, that we call – for reasons that will become clear later – *approximate* and *ultimate*, respectively. Intuitively speaking, both families approximate the original two-valued model semantics of ADFs, where the ultimate family is more *precise* in a formally defined sense. The present paper employs approximating operators for complexity analysis and thus shows that AFT is also well-suited for studying the computational complexity of formalisms.

Along with providing a comparison of the approximate and ultimate families of semantics, our main results can be summarized as follows. We show that: (1) the computational complexity of ADF decision problems is one level up in the polynomial hierarchy from their AF counterparts (Dunne and Wooldridge 2009); (2) the ultimate semantics are as complex as the approximate semantics, with the notable exception of two-valued stable models; (3) there is a certain subclass of ADFs, called *bipolar* ADFs (BADFs), which is of the same complexity as AFs. Intuitively, in bipolar ADFs all links between statements are supporting or attacking. To formalize these notions, Brewka and Woltran (2010) gave a precise semantical definition of support and attack. In our work, we assume that the link types are specified by the user along with the ADF. We consider this a harmless assumption since the existing applications of ADFs produce bipolar ADFs where the link types are known (Brewka and Gordon 2010; Strass 2013b). This attractiveness of bipolar ADFs from a KR point of view is the most significant result of the paper: it shows that BADFs offer – in addition to AF-like and more general notions of attack – also syntactical notions of support *without any increase in computational cost*.

Previously, Brewka, Dunne and Woltran (2011) translated BADFs into AFs for two-valued semantics and suggested indirectly that the complexities align. Here we go a direct route, which has more practical relevance since it immediately affects algorithm design. Our work was also inspired by the complexity analysis of assumption-based argumentation by Dimopoulos, Nebel and Toni (2002) – they derived generic results in a way similar to ours.

The paper proceeds as follows. We first provide the background on approximation fixpoint theory, abstract dialectical frameworks and the necessary elements of complexity theory. In the section afterwards, we define the relevant decision problems, survey existing complexity results, use examples to illustrate how operators revise ADF interpretations and show generic upper complexity bounds. In the main section on complexity results for general ADFs, we back up the upper bounds with matching lower bounds; the section afterwards does the same for bipolar ADFs. We end with a brief discussion of related and future work. An earlier version of this paper with more details and all proofs is available as a technical report (Strass and Wallner 2013).

Background

A *complete lattice* is a partially ordered set $(A, \sqsubseteq)$ where every subset of A has a least upper and a greatest lower bound. In particular, a complete lattice has a least and a greatest element. An operator $O : A \to A$ is *monotone* if for all $x \sqsubseteq y$ we find $O(x) \sqsubseteq O(y)$. An $x \in A$ is a *fixpoint* of O if $O(x) = x$; an $x \in A$ is a *prefixpoint* of O if $O(x) \sqsubseteq x$ and a *postfixpoint* of O if $x \sqsubseteq O(x)$. Due to a fundamental result by Tarski and Knaster, for any monotone operator O on a complete lattice, the set of its fixpoints forms a complete lattice itself (Davey and Priestley 2002, Theorem 2.35). In particular, its least fixpoint $\mathit{lfp}(O)$ exists.

In this paper, we will be concerned with more general algebraic structures: complete partially ordered sets (CPOs). A CPO is a partially ordered set with a least element where each directed subset has a least upper bound. A set is directed iff it is nonempty and each pair of elements has an upper bound in the set. Clearly every complete lattice is a complete partially ordered set, but not necessarily vice versa. Fortunately, complete partially ordered sets still guarantee the existence of (least) fixpoints for monotone operators.

Theorem 1 ((Davey and Priestley 2002, Theorem 8.22))**.** *In a complete partially ordered set $(A, \sqsubseteq)$, any $\sqsubseteq$-monotone operator $O : A \to A$ has a least fixpoint.*

Approximation Fixpoint Theory

Denecker, Marek and Truszczyński (2000) introduce the important concept of an approximation of an operator. In the study of semantics of knowledge representation formalisms, elements of lattices represent objects of interest. Operators on lattices transform such objects into others according to the contents of some knowledge base. Consequently, fixpoints of such operators are then objects that are fully updated – informally, the knowledge base can neither increase nor decrease the amount of information in a fixpoint.

To study fixpoints of operators O, DMT study their *approximation operators* $\mathcal{O}$. When O operates on a set A, its approximation $\mathcal{O}$ operates on pairs $(x, y) \in A \times A$. Such a pair (x, y) can be seen as representing a *set* of lattice elements by providing a lower bound x and an upper bound y. Consequently, (x, y) approximates all $z \in A$ such that $x \sqsubseteq z \sqsubseteq y$. We will restrict our attention to *consistent* pairs – those where $x \sqsubseteq y$, that is, the set of approximated elements is nonempty; we denote the set of all consistent pairs over A by A^c. A pair (x, y) with $x = y$ is called *exact* – it "approximates" a single element of the original lattice.

It is natural to order approximating pairs according to their information content. Formally, for $x_1, x_2, y_1, y_2 \in A$ define the *information ordering* $(x_1, y_1) \leq_i (x_2, y_2)$ iff $x_1 \sqsubseteq x_2$ and $y_2 \sqsubseteq y_1$. This ordering and the restriction to consistent pairs leads to a complete partially ordered set $(A^c, \leq_i)$, the *consistent CPO*. For example, the *trivial pair* $(\bot, \top)$ consisting of $\sqsubseteq$-least $\bot$ and $\sqsubseteq$-greatest lattice element $\top$ approximates all lattice elements and thus contains no information – it is the least element of the CPO $(A^c, \leq_i)$; exact pairs (x, x) are the maximal elements of $(A^c, \leq_i)$.

To define an approximation operator $\mathcal{O} : A^c \to A^c$, one essentially has to define two functions: a function $\mathcal{O}' : A^c \to A$ that yields a revised *lower* bound (first component) for a given pair; and a function $\mathcal{O}'' : A^c \to A$ that yields a revised *upper* bound (second component) for a given pair. Accordingly, the overall approximation is then given by $\mathcal{O}(x, y) = (\mathcal{O}'(x, y), \mathcal{O}''(x, y))$ for $(x, y) \in A^c$. The operator $\mathcal{O} : A^c \to A^c$ is *approximating* iff it is $\leq_i$-monotone and it satisfies $\mathcal{O}'(x, x) = \mathcal{O}''(x, x)$ for all $x \in A$, that is, $\mathcal{O}$ assigns exact pairs to exact pairs. Such an $\mathcal{O}$ then *ap-*

Kripke-Kleene semantics	$\mathit{lfp}(\mathcal{O})$	grounded pair
admissible/reliable pair (x, y)	$(x, y) \leq_i \mathcal{O}(x, y)$	admissible pair
three-valued supported model (x, y)	$(x, y) = \mathcal{O}(x, y)$	complete pair
M-supported model (x, y)	$(x, y) \leq_i \mathcal{O}(x, y)$ and (x, y) is $\leq_i$-maximal	preferred pair
two-valued supported model (x, x)	$(x, x) = \mathcal{O}(x, x)$	model
two-valued stable model (x, x)	$x = \mathit{lfp}(\mathcal{O}'(\cdot, x))$	stable model

Table 1: Operator-based semantical notions (and their argumentation names on the right) for a complete lattice $(A, \sqsubseteq)$ and an approximating operator $\mathcal{O} : A^c \to A^c$ on the consistent CPO. While an approximating operator always possesses three-valued (post-)fixpoints, two-valued fixpoints need not exist. Clearly, any two-valued stable model is a two-valued supported model is a preferred pair is a complete pair is an admissible pair; furthermore the grounded semantics is a complete pair.

proximates an operator $O : A \to A$ on the original lattice iff $\mathcal{O}'(x, x) = O(x)$ for all $x \in A$.

The main contribution of Denecker, Marek and Truszczyński (2000) was the association of the *stable operator* to an approximating operator. Their original definition was four-valued; in this paper we are only interested in two-valued stable models and simplified the definitions. For an approximating operator $\mathcal{O}$ on a consistent CPO, a (two-valued) pair $(x, x) \in A^c$ is a (two-valued) *stable model of* $\mathcal{O}$ iff x is the least fixpoint of the operator $\mathcal{O}'(\cdot, x)$ defined by $w \mapsto \mathcal{O}'(w, x)$ for $w \sqsubseteq x$. This general, lattice-theoretic approach yields a uniform treatment of the standard semantics of the major nonmonotonic knowledge representation formalisms – logic programming, default logic and autoepistemic logic (Denecker, Marek, and Truszczyński 2003).

In subsequent work, Denecker, Marek and Truszczyński (2004) presented a general, abstract way to define the most precise – called the *ultimate* – approximation of a given operator O. Most precise here refers to a generalisation of $\leq_i$ to operators, where for $\mathcal{O}_1, \mathcal{O}_2$, they define $\mathcal{O}_1 \leq_i \mathcal{O}_2$ iff for all $(x, y) \in A^c$ it holds that $\mathcal{O}_1(x, y) \leq_i \mathcal{O}_2(x, y)$. Denecker, Marek and Truszczyński (2004) show that the most precise approximation of O is $\mathcal{U}_O : A^c \to A^c$ that maps (x, y) to

$$\left(\bigsqcap \{O(z) \mid x \sqsubseteq z \sqsubseteq y\}, \bigsqcup \{O(z) \mid x \sqsubseteq z \sqsubseteq y\}\right)$$

where $\sqcap$ denotes the greatest lower bound and $\sqcup$ the least upper bound in the complete lattice $(A, \sqsubseteq)$.

In recent work, we defined new operator-based semantics inspired by semantics from logic programming and abstract argumentation (Strass 2013a).[1] An overview is in Table 1.

Abstract Dialectical Frameworks

An abstract dialectical framework (ADF) is a directed graph whose nodes represent statements or positions which can be accepted or not. The links represent dependencies: the status of a node s only depends on the status of its parents (denoted $par(s)$), that is, the nodes with a direct link to s. In addition, each node s has an associated acceptance condition C_s specifying the exact conditions under which s is accepted. C_s is a function assigning to each subset of $par(s)$ one of the truth values $\mathbf{t}, \mathbf{f}$. Intuitively, if for some $R \subseteq par(s)$ we have $C_s(R) = \mathbf{t}$, then s will be accepted provided the nodes in R are accepted and those in $par(s) \setminus R$ are not accepted.

[1]To be precise, we used a slightly different technical setting there. The results can however be transferred to the present setting (Denecker, Marek, and Truszczyński 2004, Theorem 4.2).

Definition 1. An *abstract dialectical framework* is a tuple $\Xi = (S, L, C)$ where

- S is a set of statements (positions, nodes),
- $L \subseteq S \times S$ is a set of links,
- $C = \{C_s\}_{s \in S}$ is a collection of total functions $C_s : 2^{par(s)} \to \{\mathbf{t}, \mathbf{f}\}$, one for each statement s. The function C_s is called *acceptance condition of* s.

It is often convenient to represent acceptance conditions by propositional formulas. In particular, we will do so for the complexity results of this paper. There, each C_s is represented by a propositional formula φ_s over $par(s)$. Then, clearly, $C_s(R \cap par(s)) = \mathbf{t}$ iff $R \models \varphi_s$. Furthermore, throughout the paper we will denote ADFs by Ξ and tacitly assume that $\Xi = (S, L, C)$ unless stated otherwise.

Brewka and Woltran (2010) introduced a useful subclass of ADFs called *bipolar*: Intuitively, in bipolar ADFs (BADFs) each link is supporting or attacking (or both). Formally, a link $(r, s) \in L$ is *supporting in* Ξ iff for all $R \subseteq par(s)$, we have that $C_s(R) = \mathbf{t}$ implies $C_s(R \cup \{r\}) = \mathbf{t}$; symmetrically, a link $(r, s) \in L$ is *attacking in* Ξ iff for all $R \subseteq par(s)$, we have that $C_s(R \cup \{r\}) = \mathbf{t}$ implies $C_s(R) = \mathbf{t}$. An ADF $\Xi = (S, L, C)$ is *bipolar* iff all links in L are supporting or attacking or both; we use L^+ to denote all supporting and L^- to denote all attacking links of L in Ξ. For an $s \in S$ we define $att_\Xi(s) = \{x \mid (x, s) \in L^-\}$ and $supp_\Xi(s) = \{x \mid (x, s) \in L^+\}$.

The semantics of ADFs can be defined using approximating operators. For two-valued semantics of ADFs we are interested in sets of statements, that is, we work in the complete lattice $(A, \sqsubseteq) = (2^S, \subseteq)$. To approximate elements of this lattice, we use consistent pairs of sets of statements and the associated consistent CPO $(A^c, \leq_i)$ – the *consistent CPO over S-subset pairs*. Such a pair $(X, Y) \in A^c$ can be regarded as a three-valued interpretation where all elements in X are true, those in $Y \setminus X$ are unknown and those in $S \setminus Y$ are false. (This allows us to use "pair" and "interpretation" synonymously from now on.) The following definition specifies how to revise a given three-valued interpretation.

Definition 2 ((Strass 2013a, Definition 3.1))**.** Let Ξ be an

ADF. Define an operator $\mathcal{G}_\Xi : 2^S \times 2^S \to 2^S \times 2^S$ by

$$\mathcal{G}_\Xi(X,Y) = (\mathcal{G}'_\Xi(X,Y), \mathcal{G}'_\Xi(Y,X))$$
$$\mathcal{G}'_\Xi(X,Y) = \{s \in S \mid B \subseteq par(s), C_s(B) = \mathbf{t}, B \subseteq X, (par(s) \setminus B) \cap Y = \emptyset\}$$

Intuitively, statement s is included in the revised lower bound iff the input pair provides sufficient reason to do so, given acceptance condition C_s. Although the operator is defined for all pairs (including inconsistent ones), its restriction to consistent pairs is well-defined since it maps consistent pairs to consistent pairs. This operator defines the *approximate* family of ADF semantics according to Table 1. Based on the three-valued operator $\mathcal{G}_\Xi$, a two-valued one-step consequence operator for ADFs can be defined by $G_\Xi(X) = \mathcal{G}'_\Xi(X,X)$. The general result of Denecker, Marek and Truszczyński (2004) (Theorem 5.6) then immediately defines the ultimate approximation of G_Ξ as the operator $\mathcal{U}_\Xi$ given by $\mathcal{U}_\Xi(X,Y) = (\mathcal{U}'_\Xi(X,Y), \mathcal{U}''_\Xi(X,Y))$ with

- $\mathcal{U}'_\Xi(X,Y) = \{s \in S \mid \text{ for all } X \subseteq Z \subseteq Y, Z \models \varphi_s\}$ and
- $\mathcal{U}''_\Xi(X,Y) = \{s \in S \mid \text{ for some } X \subseteq Z \subseteq Y, Z \models \varphi_s\}$.

Incidentally, Brewka and Woltran (2010) already defined this operator, which was later used to define the *ultimate* family of ADF semantics according to Table 1 (Brewka et al. 2013).[2] In this paper, we will refer to the two families of three-valued semantics as "approximate σ" and "ultimate σ" for σ among admissible, grounded, complete, preferred and stable. For two-valued supported models (or simply models), approximate and ultimate semantics coincide.

Finally, for a propositional formula φ over vocabulary P and $X \subseteq Y \subseteq P$ we define the *partial valuation of* φ *by* (X,Y) as $\varphi^{(X,Y)} = \varphi[p/\mathbf{t} : p \in X][p/\mathbf{f} : p \in P \setminus Y]$. This partial evaluation takes the two-valued part of (X,Y) and replaces the evaluated variables by their truth values. Naturally, $\varphi^{(X,Y)}$ is a formula over the vocabulary $Y \setminus X$.

Complexity theory

We assume familiarity with the complexity classes P, NP and coNP, as well as with polynomial reductions and hardness and completeness for these classes. We also make use of the polynomial hierarchy, that can be defined (using oracle Turing machines) as follows: $\Sigma_0^P = \Pi_0^P = \Delta_0^P = \mathsf{P}$, $\Sigma_{i+1}^P = \mathsf{NP}^{\Sigma_i^P}, \Pi_{i+1}^P = \mathsf{coNP}^{\Sigma_i^P}, \Delta_{i+1}^P = \mathsf{P}^{\Sigma_i^P}$ for $i \geq 0$.

As a somewhat non-standard polynomial hierarchy complexity class, we use D_i^P, a generalisation of the complexity class DP to the polynomial hierarchy. A language is in DP iff it is the intersection of a language in NP and a language in coNP. Generally, a language is in D_i^P iff it is the intersection of a language in Σ_i^P and a language in Π_i^P. The canonical problem of $\mathsf{DP} = \mathrm{D}_1^P$ is SAT-UNSAT, the problem to decide for a given pair (ψ_1, ψ_2) of propositional formulas whether ψ_1 is satisfiable and ψ_2 is unsatisfiable. Obviously, by definition $\Sigma_i^P, \Pi_i^P \subseteq \mathrm{D}_i^P \subseteq \Delta_{i+1}^P$ for all $i \geq 0$.

[2] Technically, Brewka et al. (2013) represented interpretations not by pairs $(X,Y) \in A^c$ but by mappings $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ into the set of truth values $\mathbf{t}$ (true), $\mathbf{f}$ (false) and $\mathbf{u}$ (unknown or undecided). Clearly the two representations are interchangeable.

Preparatory Considerations

We first introduce some notation to make precise what decision problems we will analyze. For a set S, let

- $(A^c, \leq_i)$ be the consistent CPO of S-subset pairs,
- $\mathcal{O}$ an approximating operator on $(A^c, \leq_i)$,
- $\sigma \in \{adm, com, grd, pre, 2su, 2st\}$ a semantics among admissible, complete, grounded, preferred, two-valued supported and two-valued stable semantics, respectively.

In the *verification* problem we decide whether $(X,Y) \in A^c$ is a σ-model/pair of $\mathcal{O}$, denoted by $\mathsf{Ver}_\sigma^\mathcal{O}(X,Y)$. In the *existence* problem we ask whether there exists a σ-model/pair of $\mathcal{O}$ which is non-trivial, that is, different from $(\emptyset, S)$, denoted by $\mathsf{Exists}_\sigma^\mathcal{O}$. For query reasoning and $s \in S$ we consider the problem of deciding whether there exists a σ-model/pair (X,Y) of $\mathcal{O}$ such that $s \in X$, denoted by $\mathsf{Cred}_\sigma^\mathcal{O}(s)$ (*credulous* reasoning) and the problem of deciding whether in all σ-models/pairs (X,Y) of $\mathcal{O}$ we have $s \in X$, denoted by $\mathsf{Skept}_\sigma^\mathcal{O}(s)$ (*skeptical* reasoning). Note that it is no restriction to check only for truth of a statement $s \in S$, since checking for falsity can be modeled by introducing a new statement s' that behaves like the logical negation of s, by setting its acceptance condition to $\varphi_{s'} = \neg s$.

Existing results

We briefly survey – to the best of our knowledge – all existing complexity results for abstract dialectical frameworks. For general ADFs Ξ and the ultimate family of semantics, Brewka et al. (2013) have shown the following:

- $\mathsf{Ver}_{\mathsf{2su}}^{\mathcal{U}_\Xi}$ is in P, $\mathsf{Exists}_{\mathsf{2su}}^{\mathcal{U}_\Xi}$ is NP-complete (Proposition 5)
- $\mathsf{Ver}_{\mathsf{adm}}^{\mathcal{U}_\Xi}$ is coNP-complete (Proposition 10)
- $\mathsf{Ver}_{\mathsf{grd}}^{\mathcal{U}_\Xi}$ and $\mathsf{Ver}_{\mathsf{com}}^{\mathcal{U}_\Xi}$ are DP-complete (Theorem 6, Cor. 7)
- $\mathsf{Ver}_{\mathsf{2st}}^{\mathcal{U}_\Xi}$ is in DP (Proposition 8)
- $\mathsf{Exists}_{\mathsf{2st}}^{\mathcal{U}_\Xi}$ is Σ_2^P-complete (Theorem 9)

For bipolar ADFs, Brewka and Woltran (2010) showed that $\mathsf{Ver}_{\mathsf{grd}}^{\mathcal{U}_\Xi}$ is in P (Proposition 15). So particularly for BADFs, this paper will greatly illuminate the complexity landscape.

Relationship between the operators

Since $\mathcal{U}_\Xi$ is the ultimate approximation of G_Ξ it is clear that for any $X \subseteq Y \subseteq S$ we have $\mathcal{G}_\Xi(X,Y) \leq_i \mathcal{U}_\Xi(X,Y)$. In other words, the ultimate revision operator produces new bounds that are at least as tight as those of the approximate operator. More explicitly, the ultimate new lower bound always contains the approximate new lower bound: $\mathcal{G}'_\Xi(X,Y) \subseteq \mathcal{U}'_\Xi(X,Y)$; conversely, the ultimate new upper bound is contained in the approximate new upper bound: $\mathcal{U}''_\Xi(X,Y) \subseteq \mathcal{G}''_\Xi(X,Y)$. Somewhat surprisingly, it turns out that the revision operators for the upper bound coincide.

Lemma 2. *Let* $\Xi = (S,L,C)$ *be an ADF and* $X \subseteq Y \subseteq S$.

$$\mathcal{G}''_\Xi(X,Y) = \mathcal{U}''_\Xi(X,Y)$$

The operators for computing a new lower bound are demonstrably different, since we can find Ξ and (X,Y) with $\mathcal{U}'_\Xi(X,Y) \not\subseteq \mathcal{G}'_\Xi(X,Y)$, as the following ADF shows.

Example 1. Consider the ADF $D = (\{a\}, \{(a,a)\}, \{\varphi_a\})$ with one self-dependent statement a that has acceptance formula $\varphi_a = a \vee \neg a$. In Figure 1, we show the relevant CPO and the behavior of approximate and ultimate operators: we see that $\mathcal{G}_D(\emptyset, \{a\}) <_i \mathcal{U}_D(\emptyset, \{a\})$, which shows that in some cases the ultimate operator is strictly more precise.

So in a sense the approximate operator cannot see beyond the case distinction $a \vee \neg a$. As we will see shortly, this difference really amounts to the capability of tautology checking.

Example 2. ADF $E = (\{a,b\}, \{(b,a),(b,b)\}, \{\varphi_a, \varphi_b\})$ has acceptance formulas $\varphi_a = b \vee \neg b$ and $\varphi_b = \neg b$. So b is self-attacking and the link from b to a is redundant. In Figure 1 on the next page, we show the relevant CPO and the behavior of the operators $\mathcal{U}_E$ and $\mathcal{G}_E$ on this CPO.

The examples show that the approximate and ultimate families of semantics really are different, save for one straightforward inclusion relation in case of admissible.

Corollary 3. *For any ADF Ξ, we have the following:*

1. *An approximate admissible pair is an ultimate admissible pair, but not vice versa.*
2. *With respect to their sets of pairs, the approximate and ultimate versions of preferred/complete/grounded semantics are $\subseteq$-incomparable.*

Operator complexities

We next analyze the computational complexity of deciding whether a single statement is contained in the lower or upper bound of the revision of a given pair. This then leads to the complexity of checking whether current lower/upper bounds are pre- or postfixpoints of the revision operators for computing new lower/upper bounds, that is, whether the revisions represent improvements in terms of the information ordering. Intuitively, these results describe how hard it is to "use" the operators and lay the foundation for the rest of the complexity results.

Proposition 4. *Let Ξ be an ADF, $s \in S$ and $X \subseteq Y \subseteq S$.*

1. *Deciding $s \in \mathcal{G}'_\Xi(X,Y)$ is in* P.
2. *Deciding $\mathcal{G}'_\Xi(X,Y) \subseteq X$ is in* P.
3. *Deciding $X \subseteq \mathcal{G}'_\Xi(X,Y)$ is in* P.

Now let $\mathcal{O} \in \{\mathcal{G}_\Xi, \mathcal{U}_\Xi\}$.

4. *Deciding $s \in \mathcal{O}''(X,Y)$ is* NP*-complete.*
5. *Deciding $\mathcal{O}''(X,Y) \subseteq Y$ is* coNP*-complete.*
6. *Deciding $Y \subseteq \mathcal{O}''(X,Y)$ is* NP*-complete.*

Proof. We only show 1 and 4 as the rest follows suit.

1. Since $X \subseteq Y$, we have that whenever there exists a $B \subseteq X \cap par(s)$ with $C_s(B) = \mathbf{t}$ and $(par(s) \setminus B) \cap Y = \emptyset$, we know that $B = X \cap par(s)$. Thus $s \in \mathcal{G}'_\Xi(X,Y)$ iff $C_s(X \cap par(s)) = \mathbf{t}$ and $(par(s) \setminus X) \cap Y = \emptyset$. For acceptance functions represented by propositional formulas, both conditions can be checked in polynomial time.
4. Deciding $s \in \mathcal{G}''_\Xi(X,Y)$ is NP-complete:

   in NP: By definition, $\mathcal{G}''_\Xi(X,Y) = \mathcal{G}'_\Xi(Y,X)$. To verify $s \in \mathcal{G}'_\Xi(Y,X)$, we can guess a set $M \subseteq S$ and verify that $M \subseteq Y$, $par(s) \setminus M \subseteq S \setminus X$ and $M \models \varphi_s$.

   NP-hard: For hardness, we provide a reduction from SAT. Let ψ be a propositional formula over vocabulary P. Define an ADF $\Xi = (S, L, C)$ with $S = P \cup \{z\}$ where $z \notin P$, $\varphi_z = \psi$ and $\varphi_p = p$ for all $p \in P$. Observe that $par(z) = P$, and set $X = \emptyset$ and $Y = P$. Now $z \in \mathcal{G}''_\Xi(X,Y)$ iff $z \in \mathcal{G}'_\Xi(Y,X)$ iff $z \in \mathcal{G}'_\Xi(P,\emptyset)$ iff there is an $M \subseteq P$ with $P \setminus M \cap \emptyset = \emptyset$ and $M \models \varphi_z$ iff there is an $M \subseteq P$ with $M \models \psi$ iff ψ is satisfiable. □

These results can also be formulated in terms of partial evaluations of acceptance formulas: We have $s \in \mathcal{G}'_\Xi(X,Y)$ iff the partial evaluation $\varphi_s^{(X,Y)}$ is a formula without variables that has truth value $\mathtt{t}$. Similarly, we have $s \in \mathcal{G}''_\Xi(X,Y)$ iff the partial evaluation $\varphi_s^{(X,Y)}$ is satisfiable. Under standard complexity assumptions, computing a new lower bound with the ultimate operator is harder than with the approximate operator. This is because, intuitively, $s \in \mathcal{U}'_\Xi(X,Y)$ iff the partial evaluation $\varphi_s^{(X,Y)}$ is a tautology.

Proposition 5. *Let Ξ be an ADF, $s \in S$ and $X \subseteq Y \subseteq S$.*

1. *Deciding $s \in \mathcal{U}'_\Xi(X,Y)$ is* coNP*-complete.*
2. *Deciding $\mathcal{U}'_\Xi(X,Y) \subseteq X$ is* NP*-complete.*
3. *Deciding $X \subseteq \mathcal{U}'_\Xi(X,Y)$ is* coNP*-complete.*

Proof. We only show the first item since the remaining proofs work along the same lines. The hardness proof uses the ADF from Proposition 4.

in coNP: To decide that $s \notin \mathcal{U}'_\Xi(X,Y)$, we guess a Z with $X \subseteq Z \subseteq Y$ and verify that $Z \not\models \varphi_s$.

coNP-hard: Set $X = \emptyset$ and $Y = P$. Now $z \in \mathcal{U}'_\Xi(X,Y)$ iff $z \in \mathcal{U}'_\Xi(\emptyset, P)$ iff for all $Z \subseteq P$, we have $Z \models \varphi_z$ iff for all $Z \subseteq P$, we have $Z \models \psi$ iff ψ is a tautology. □

The next result considerably simplifies the complexity analysis of deciding the existence of non-trivial pairs.

Lemma 6. *Let $(A, \sqsubseteq)$ be a complete lattice and $\mathcal{O}$ an approximating operator on A^c. The following are equivalent:*

1. *$\mathcal{O}$ has a non-trivial admissible pair.*
2. *$\mathcal{O}$ has a non-trivial preferred pair.*
3. *$\mathcal{O}$ has a non-trivial complete pair.*

Proof. "(1) ⇒ (2)": Let $(\bot, \top) <_i (x,y) \leq_i \mathcal{O}(x,y)$. We show that there is a preferred pair $(p,q) \geq_i (x,y)$. Define $D = \{(a,b) \mid (x,y) \leq_i (a,b)\}$, then the pair $(D, \leq_i)$ is a CPO on which $\mathcal{O}$ is an approximating operator. (Obviously $(a,b) \in D$ implies $(x,y) \leq_i (a,b)$ whence by presumption and $\leq_i$-monotonicity of $\mathcal{O}$ we get $(x,y) \leq_i \mathcal{O}(x,y) \leq_i \mathcal{O}(a,b)$ and $\mathcal{O}(a,b) \in D$.) Now any sequence $(a,b) \leq_i \mathcal{O}(a,b) \leq_i \mathcal{O}(\mathcal{O}(a,b)) \leq_i \ldots$ is a non-empty chain in D and therefore has an upper bound in D. By Zorn's lemma, the set of all $\mathcal{O}$-admissible pairs in A has a maximal element $(p,q) \geq_i (x,y) >_i (\bot, \top)$.

"(2) ⇒ (3)": By (Strass 2013a, Theorem 3.10), every preferred pair is complete.

"(3) ⇒ (1)": Any complete pair is admissible (Table 1). □

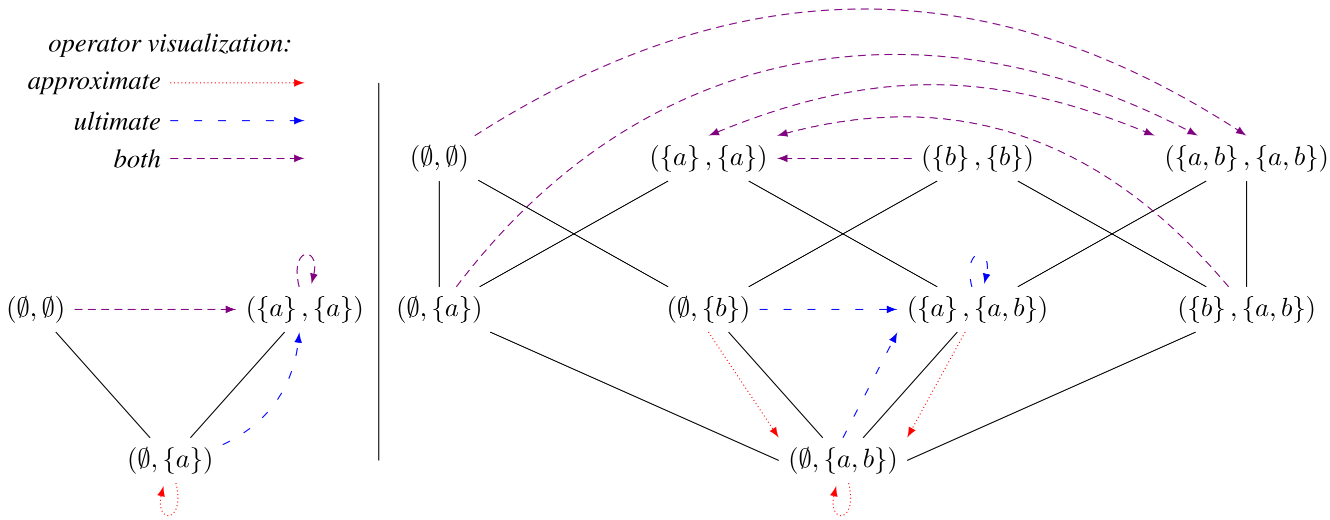

Figure 1: Hasse diagrams of consistent CPOs for the ADFs from Example 1 (left) and Example 2 (right). Solid lines represent the information ordering $\leq_i$. Directed arrows express how revision operators map pairs to other pairs. For pairs where the revisions coincide, the arrows are densely dashed and violet. When the operators revise a pair differently, we use a dotted red arrow for the ultimate and a loosely dashed blue arrow for the approximate operator. Exact (two-valued) pairs are the $\leq_i$-maximal elements. For those pairs, (and any ADF Ξ) it is clear that the operators $\mathcal{U}_\Xi$ and $\mathcal{G}_\Xi$ coincide since they approximate the same two-valued operator G_Ξ. In Example 1 on the left, we can see that the ultimate operator maps all pairs to its only fixpoint $(\{a\},\{a\})$ where a is true. The approximate operator has an additional fixpoint, $(\emptyset,\{a\})$, where a is unknown. In Example 2 on the right, the major difference between the operators is whether statement a can be derived given that b has truth value unknown. This is the case for the ultimate, but not for the approximate operator. Since there is no fixpoint in the upper row (showing the two-valued operator G_E), the ADF E does not have a two-valued model. Each of the revision operators has however exactly one three-valued fixpoint, which thus constitutes the respective grounded, preferred and complete semantics.

This directly shows the equivalence of the respective decision problems, that is, $\mathsf{Exists}^{\mathcal{O}}_{\mathsf{adm}} = \mathsf{Exists}^{\mathcal{O}}_{\mathsf{pre}} = \mathsf{Exists}^{\mathcal{O}}_{\mathsf{com}}$.

Regarding decision problems for querying, skeptical reasoning w.r.t. admissibility is trivial, i.e. $(\emptyset, S)$ is always an admissible pair in any ADF. Further credulous reasoning w.r.t. admissibility, complete and preferred semantics coincides.

Lemma 7. *Let Ξ be an ADF, $\mathcal{O} \in \{\mathcal{G}_\Xi, \mathcal{U}_\Xi\}$ and $s \in S$. Then $\mathsf{Cred}^{\mathcal{O}}_{\mathsf{adm}}(s)$ iff $\mathsf{Cred}^{\mathcal{O}}_{\mathsf{com}}(s)$ iff $\mathsf{Cred}^{\mathcal{O}}_{\mathsf{pre}}(s)$.*

Proof. Assume (X, Y) with $s \in X$ is admissible w.r.t. $\mathcal{O}$, then there exists a (X', Y') with $(X, Y) \leq_i (X', Y')$ which is preferred with respect to $\mathcal{O}$ and where $s \in X'$, see proof of Lemma 6. Since any preferred pair is also complete and any complete pair is also admissible the claim follows. □

Generic upper bounds

We now show generic upper bounds for the computational complexity of the considered problems. This kind of analysis is in the spirit of the results of Dimopoulos, Nebel and Toni (2002) (Section 4). The first item is furthermore a straightforward generalization of Theorem 6.13 in (Denecker, Marek, and Truszczyński 2004).

Theorem 8. *Let S be a finite set, define $A = 2^S$ and let $\mathcal{O}$ be an approximating operator on $(A^c, \leq_i)$, the consistent CPO of S-subset pairs. For a pair $(X, Y) \in A^c$ and an $s \in S$, let the problem of deciding whether $s \in \mathcal{O}'(X, Y)$ be in Π^P_i; let the problem of deciding $s \in \mathcal{O}''(X, Y)$ be in Σ^P_i. For $(X, Y) \in A^c$ and a statement $s \in S$, we have:*

1. *The least fixpoint of $\mathcal{O}$ can be computed in polynomial time with a polynomial number of calls to a Σ^P_i-oracle.*
2. *$\mathsf{Ver}^{\mathcal{O}}_{\mathsf{adm}}(X, Y)$ is in Π^P_i; $\mathsf{Cred}^{\mathcal{O}}_{\mathsf{adm}}(s)$ is in Σ^P_{i+1};*
3. *$\mathsf{Ver}^{\mathcal{O}}_{\mathsf{com}}(X, Y)$ is in D^P_i; $\mathsf{Cred}^{\mathcal{O}}_{\mathsf{com}}(s)$ is in Σ^P_{i+1};*
4. *$\mathsf{Ver}^{\mathcal{O}}_{\mathsf{pre}}(X, Y)$ is in Π^P_{i+1}; $\mathsf{Cred}^{\mathcal{O}}_{\mathsf{pre}}(s)$ is in Σ^P_{i+1}; $\mathsf{Skept}^{\mathcal{O}}_{\mathsf{pre}}(s)$ is in Π^P_{i+2}.*

Proof. 1. For any $(V, W) \in A^c$ we can use the oracle to compute an application of $\mathcal{O}'$ by simply asking whether $z \in \mathcal{O}'(V, W)$ for each $z \in S$. This means we can compute with a linear number of oracle calls the sets $\mathcal{O}'(V, W)$ and $\mathcal{O}''(V, W)$, thus the pair $\mathcal{O}(V, W)$. Hence we can compute the sequence $(\emptyset, S) \leq_i \mathcal{O}(\emptyset, S) \leq_i \mathcal{O}(\mathcal{O}(\emptyset, S)) \leq_i \ldots$ which converges to the least fixpoint of $\mathcal{O}$ after a linear number of operator applications.

2. We can decide $\mathcal{O}'(X, Y) \subseteq X$ and $Y \subseteq \mathcal{O}''(X, Y)$ in Σ^P_i, $X \subseteq \mathcal{O}'(X, Y)$ and $\mathcal{O}''(X, Y) \subseteq Y$ in Π^P_i; all by combining independent guesses. Then $\mathsf{Ver}^{\mathcal{O}}_{\mathsf{adm}}(X, Y)$ is in Π^P_i. For $\mathsf{Cred}^{\mathcal{O}}_{\mathsf{adm}}(s)$, we guess a pair (X_1, Y_1) with $s \in X_1$ and check if it is admissible.

3. $\mathsf{Ver}^{\mathcal{O}}_{\mathsf{com}}(X, Y)$ is in D^P_i by the same method as for admissibility. $\mathsf{Cred}^{\mathcal{O}}_{\mathsf{com}}(s) = \mathsf{Cred}^{\mathcal{O}}_{\mathsf{adm}}(s)$ by Lemma 7.

4. For $\mathsf{Ver}^{\mathcal{O}}_{\mathsf{pre}}(X,Y)$, consider the co-problem, i.e. deciding whether (X,Y) is not a preferred pair. We first check if (X,Y) is a complete pair, which is in D^P_i. If this holds, we guess an (X_1,Y_1) with $(X,Y) <_i (X_1,Y_1)$ and check if it is complete. $\mathsf{Cred}^{\mathcal{O}}_{\mathsf{pre}}(X,Y)$: coincides with credulous reasoning w.r.t. admissibility, see Lemma 7. $\mathsf{Skept}^{\mathcal{O}}_{\mathsf{pre}}(s)$: Consider the co-problem, i.e. deciding whether there exists a preferred pair (X_1,Y_1) with $X_1 \cap \{a\} = \emptyset$. We guess such a pair (X_1,Y_1) and check if it is preferred. □

Naturally, the capability of solving the functional problem of *computing* the grounded semantics allows us to solve the associated decision problems.

Corollary 9. *Under the assumptions of Theorem 8, the problems* $\mathsf{Ver}^{\mathcal{O}}_{\mathsf{grd}}$ *and* $\mathsf{Exists}^{\mathcal{O}}_{\mathsf{grd}}$ *are in* Δ^P_{i+1}.

Complexity of General ADFs

Due to the coincidence of $\mathcal{G}''_{\Xi}$ and $\mathcal{U}''_{\Xi}$, the computational complexities of decision problems that concern only the upper bound operator also coincide. This will save both work and space in the subsequent developments. Additionally, for all containment results (except for the grounded semantics), we can use Theorem 8 and need only show hardness.

Proposition 10. *Let* Ξ *be an ADF,* $X,Y \subseteq S$ *and consider any* $\mathcal{O} \in \{\mathcal{G}_{\Xi},\mathcal{U}_{\Xi}\}$. $\mathsf{Ver}^{\mathcal{O}}_{\mathsf{adm}}(X,Y)$ *is* coNP*-complete.*

Proof. Hardness follows from Proposition 4, item 5. □

Recall that a pair (X,Y) is an approximate/ultimate complete pair iff it is a fixpoint of the corresponding (approximate/ultimate) operator. Given the complexities of operator computation, it is straightforward to show the following.

Proposition 11. *Let* Ξ *be an ADF,* $X \subseteq Y \subseteq S$ *and consider any* $\mathcal{O} \in \{\mathcal{G}_{\Xi},\mathcal{U}_{\Xi}\}$. $\mathsf{Ver}^{\mathcal{O}}_{\mathsf{com}}(X,Y)$ *is* DP*-complete.*

Next, we analyze the complexity of verifying that a given pair is the approximate (ultimate) Kripke-Kleene semantics of an ADF Ξ, that is, the least fixpoint of $\mathcal{G}_{\Xi}$ ($\mathcal{U}_{\Xi}$). Although interesting, the proof is lengthy and technical, so we unfortunately have to omit it due to space constraints. Interestingly, the membership part is the tricky one, where we encode the steps of the operator computation into propositional logic.

Theorem 12. *Let* Ξ *be an ADF,* $X \subseteq Y \subseteq S$ *and consider any operator* $\mathcal{O} \in \{\mathcal{G}_{\Xi},\mathcal{U}_{\Xi}\}$. $\mathsf{Ver}^{\mathcal{O}}_{\mathsf{grd}}(X,Y)$ *is* DP*-complete.*

We next ask whether there exists a *non-trivial* admissible pair, that is, if at least one statement has a truth value other than unknown. Clearly, we can guess a pair and perform the coNP-check to show that it is admissible. The next result shows that this is also the best we can do. Again, the proof is lengthy and technical and we could not include it here.

Theorem 13. *Let* Ξ *be an ADF and consider any operator* $\mathcal{O} \in \{\mathcal{G}_{\Xi},\mathcal{U}_{\Xi}\}$. $\mathsf{Exists}^{\mathcal{O}}_{\mathsf{adm}}$ *is* Σ^P_2*-complete.*

Lemma 6 implies the same complexity for the existence of non-trivial complete and preferred pairs.

Corollary 14. *Let* Ξ *be an ADF,* $\sigma \in \{com, pre\}$ *and consider any operator* $\mathcal{O} \in \{\mathcal{G}_{\Xi},\mathcal{U}_{\Xi}\}$. $\mathsf{Exists}^{\mathcal{O}}_{\sigma}$ *is* Σ^P_2*-complete.*

By corollary to Theorem 12, the existence of a non-trivial grounded pair can be decided in DP by testing whether the trivial pair $(\emptyset,S)$ is (not) a fixpoint of the relevant operator. The following result shows that this bound can be improved. We cannot present the proof here but can say that intuitively, half of the usual subset checks can be left out due to using the trivial pair.

Proposition 15. *Let* Ξ *be an ADF and consider any operator* $\mathcal{O} \in \{\mathcal{G}_{\Xi},\mathcal{U}_{\Xi}\}$. $\mathsf{Exists}^{\mathcal{O}}_{\mathsf{grd}}$ *is* coNP*-complete.*

Using the result for existence of non-trivial admissible pairs, the verification complexity for the preferred semantics is straightforward to obtain, similarly as in the case of AFs (Dimopoulos and Torres 1996).

Proposition 16. *Let* Ξ *be an ADF,* $X \subseteq Y \subseteq S$ *and consider any* $\mathcal{O} \in \{\mathcal{G}_{\Xi},\mathcal{U}_{\Xi}\}$. $\mathsf{Ver}^{\mathcal{O}}_{\mathsf{pre}}(X,Y)$ *is* Π^P_2*-complete.*

Considering query reasoning we now show that on general ADFs credulous reasoning with respect to admissibility is harder than on AFs. By Lemma 7, the same lower bound holds for complete and preferred semantics.

Proposition 17. *Let* Ξ *be an ADF,* $\mathcal{O} \in \{\mathcal{G}_{\Xi},\mathcal{U}_{\Xi}\}$ *be an operator and* $s \in S$. $\mathsf{Cred}^{\mathcal{O}}_{\mathsf{adm}}(s)$ *is* Σ^P_2*-complete.*

For credulous and skeptical reasoning with respect to the grounded semantics, we first observe that the two coincide since there is always a unique grounded pair. Furthermore, a statement s is true in the approximate grounded pair iff s is true in the least fixpoint (of $\mathcal{G}_{\Xi}$) iff s is true in all fixpoints iff there is no fixpoint where s is unknown or false. This condition can be encoded in propositional logic and leads to the next result. For the ultimate operator we can use results for the verification problem (Brewka et al. 2013, Theorem 6). Briefly put, the problem is in coNP since the NP-hardness comes from verifying that certain arguments are undefined in the ultimate grounded pair, which is not needed for credulous reasoning. For coNP-hardness the proof of (Brewka and Woltran 2010, Proposition 13) can be easily adapted.

Proposition 18. *Let* Ξ *be an ADF,* $\mathcal{O} \in \{\mathcal{G}_{\Xi},\mathcal{U}_{\Xi}\}$, $s \in S$. *Both* $\mathsf{Cred}^{\mathcal{O}}_{\mathsf{grd}}(s)$ *and* $\mathsf{Skept}^{\mathcal{O}}_{\mathsf{grd}}(s)$ *are* coNP*-complete.*

Regarding skeptical reasoning for the remaining semantics we need only show the results for complete and preferred semantics, in all other cases the complexity coincides with credulous reasoning or is trivial. For complete semantics it is easy to see that skeptical reasoning coincides with skeptical reasoning under grounded semantics, since the grounded pair is the $\leq_i$-least complete pair.

Corollary 19. *Let* Ξ *be an ADF,* $\mathcal{O} \in \{\mathcal{G}_{\Xi},\mathcal{U}_{\Xi}\}$ *and* $s \in S$. $\mathsf{Skept}^{\mathcal{O}}_{\mathsf{com}}(s)$ *is* coNP*-complete.*

Similar to reasoning on AFs, we step up one level of the polynomial hierarchy by changing from credulous to skeptical reasoning with respect to preferred semantics, which makes skeptical reasoning under preferred semantics particularly hard. We apply proof ideas by (Dunne and Bench-Capon 2002) to prove Π^P_3-hardness.

Theorem 20. *Let* Ξ *be an ADF,* $\mathcal{O} \in \{\mathcal{G}_{\Xi},\mathcal{U}_{\Xi}\}$ *and* $s \in S$. $\mathsf{Skept}^{\mathcal{O}}_{\mathsf{pre}}(s)$ *is* Π^P_3*-complete.*

Two-valued semantics

The complexity results we have obtained so far might lead the reader to ask why we bother with the approximate operator $\mathcal{G}_\Xi$ at all: the ultimate operator $\mathcal{U}_\Xi$ is at least as precise and for all the semantics considered up to now, it has the same computational costs. We now show that for the verification of two-valued stable models, the operator for the upper bound plays no role and therefore the complexity difference between the lower bound operators for approximate (in P) and ultimate (coNP-hard) semantics comes to bear.

For the ultimate two-valued stable semantics, Brewka et al. (2013) already have some complexity results: model verification is in DP (Proposition 8), and model existence is Σ_2^P-complete (Theorem 9). We will show next that we can do better for the approximate version.

Proposition 21. *Let Ξ be an ADF and $X \subseteq Y \subseteq S$. Verifying that X is the least fixpoint of $\mathcal{G}'_\Xi(\cdot, Y)$ is in* P.

Proof sketch. Roughly, we construct the sequence defined by $X_0 = \emptyset$ and $X_{i+1} = \mathcal{G}'_\Xi(X_i, Y)$ for $i \geq 0$, as long as $X_i \subseteq Y$. By $\leq_i$-monotonicity of $\mathcal{G}_\Xi$, this sequence is monotonically $\subseteq$-increasing and so the procedure terminates after a linear number of steps. We then check if $X_{i+1} = X_i = X$, that is, the right fixpoint was reached. □

In particular, the procedure can decide whether Y is the least fixpoint of $\mathcal{G}'_\Xi(\cdot, Y)$, that is, whether (Y, Y) is a two-valued stable model of $\mathcal{G}_\Xi$. This yields the next result.

Theorem 22. *Let Ξ be an ADF and $X \subseteq S$. 1. $\mathsf{Ver}_{2st}^{\mathcal{G}_\Xi}(X, X)$ is in* P. *2. $\mathsf{Exists}_{2st}^{\mathcal{G}_\Xi}$ is* NP*-complete.*

The hardness direction of the second part is clear since the respective result from stable semantics of abstract argumentation frameworks carries over.

Brewka et al. (2013) showed that $\mathsf{Ver}_{2st}^{\mathcal{U}_\Xi}$ is in DP (Proposition 8). As one of the most surprising results of this paper, we can improve that upper bound to coNP. The proof is not at all trivial, but basically the operator for the upper bound (contributing the NP part) is not really needed. Using the complexity of the lower revision operator $\mathcal{U}'_\Xi$, we can even show completeness for coNP.

Proposition 23. *Let Ξ be an ADF and $X \subseteq S$. $\mathsf{Ver}_{2st}^{\mathcal{U}_\Xi}(X, X)$ is* coNP*-complete.*

We now turn to the credulous and skeptical reasoning problems for the two-valued semantics. We first recall that a two-valued pair (X, X) is a supported model (or model for short) of an ADF Ξ iff $\mathcal{G}_\Xi(X, X) = (X, X)$. Thus it could equally well be characterized by the two-valued operator by saying that X is a model iff $G_\Xi(X) = X$. Now since $\mathcal{U}_\Xi$ is the ultimate approximation of G_Ξ, also $\mathcal{U}_\Xi(X, X) = (X, X)$ in this case. Rounding up, this recalls that approximate and ultimate two-valued supported models coincide. Hence we get the following results for reasoning with this semantics.

Corollary 24. *Let Ξ be an ADF, $\mathcal{O} \in \{\mathcal{G}_\Xi, \mathcal{U}_\Xi\}$ be an operator and $s \in S$. The problem $\mathsf{Cred}_{2su}^{\mathcal{O}}(s)$ is* NP*-complete; $\mathsf{Skept}_{2su}^{\mathcal{O}}(s)$ is* coNP*-complete.*

For the approximate two-valued stable semantics, the fact that model verification can be decided in polynomial time leads to the next result.

$\mathcal{O}$	$\mathcal{G}_\Xi, \mathcal{U}_\Xi$				$\mathcal{G}_\Xi, \mathcal{U}_\Xi$	$\mathcal{G}_\Xi$	$\mathcal{U}_\Xi$
σ	adm	com	pre	grd	2su	2st	2st
$\mathsf{Ver}_\sigma^{\mathcal{O}}$	coNP-c	DP-c	Π_2^P-c	DP-c	in P		coNP-c
$\mathsf{Exists}_\sigma^{\mathcal{O}}$	Σ_2^P-c	Σ_2^P-c	Σ_2^P-c	coNP-c	NP-c		Σ_2^P-c
$\mathsf{Cred}_\sigma^{\mathcal{O}}$	Σ_2^P-c	Σ_2^P-c	Σ_2^P-c	coNP-c	NP-c		Σ_2^P-c
$\mathsf{Skept}_\sigma^{\mathcal{O}}$	trivial	coNP-c	Π_3^P-c	coNP-c	coNP-c		Π_2^P-c

Table 2: Complexity results for semantics of ADFs.

Corollary 25. *Let Ξ be an ADF and $s \in S$. $\mathsf{Cred}_{2st}^{\mathcal{G}_\Xi}(s)$ is* NP*-complete; $\mathsf{Skept}_{2st}^{\mathcal{G}_\Xi}(s)$ is* coNP*-complete.*

For the ultimate two-valued stable semantics, things are bit more complex. The hardness reduction in the proof of Theorem 9 in (Brewka et al. 2013) makes use of a statement z that is false in any ultimate two-valued stable model. This can be used to show the same hardness for the credulous reasoning problem for this semantics: we introduce a new statement x that behaves just like $\neg z$, then x is true in some model if and only if there exists a model.

Proposition 26. *Let Ξ be an ADF and $s \in S$. The problem $\mathsf{Cred}_{2st}^{\mathcal{U}_\Xi}(s)$ is Σ_2^P-complete.*

A similar argument works for the skeptical reasoning problem: Given a QBF $\forall P \exists Q \psi$, we construct its negation $\exists P \forall Q \neg \psi$, whose associated ADF D has an ultimate two-valued stable model (where z is false) iff $\exists P \forall Q \neg \psi$ is true iff the original QBF $\forall P \exists Q \psi$ is false. Hence $\forall P \exists Q \psi$ is true iff z is true in all ultimate two-valued stable models of D.

Proposition 27. *Let Ξ be an ADF and $s \in S$. The problem $\mathsf{Skept}_{2st}^{\mathcal{U}_\Xi}(s)$ is Π_2^P-complete.*

Complexity of Bipolar ADFs

We first note that since BADFs are a subclass of ADFs, all membership results from the previous section immediately carry over. However, we can show that many problems will in fact become easier. Intuitively, computing the revision operators is now P-easy because the associated satisfiability/tautology problems only have to treat restricted acceptance formulas. In bipolar ADFs, by definition, if in some three-valued pair (X, Y) a statement s is accepted by a revision operator ($s \in \mathcal{O}'(X, Y)$), it will stay so if we set its undecided supporters to true and its undecided attackers to false. Symmetrically, if a statement is rejected by an operator ($s \notin \mathcal{O}''(X, Y)$), it will stay so if we set its undecided supporters to false and its undecided attackers to true. This is the key idea underlying the next result.

Proposition 28. *Let Ξ be a BADF with $L = L^+ \cup L^-$, $\mathcal{O} \in \{\mathcal{G}_\Xi, \mathcal{U}_\Xi\}$, $s \in S$ and $X \subseteq Y \subseteq S$.*

1. *Deciding $s \in \mathcal{O}'(X, Y)$ is in* P.
2. *Deciding $s \in \mathcal{O}''(X, Y)$ is in* P.

Using the generic upper bounds of Theorem 8, it is now straightforward to show membership results for BADFs with known link types.

Corollary 29. *Let Ξ be a BADF with $L = L^+ \cup L^-$, consider any operator $\mathcal{O} \in \{\mathcal{G}_{\sqsubseteq}, \mathcal{U}_{\sqsubseteq}\}$ and semantics $\sigma \in \{adm, com\}$. For $X \subseteq Y \subseteq S$ and $s \in S$, we find that*

- $\mathsf{Ver}_{\sigma}^{\mathcal{O}}(X,Y)$ *and* $\mathsf{Ver}_{\mathsf{grd}}^{\mathcal{O}}(X,Y)$ *are in* P*;*
- $\mathsf{Ver}_{\mathsf{pre}}^{\mathcal{O}}(X,Y)$ *is in* coNP*;*
- $\mathsf{Exists}_{\sigma}^{\mathcal{O}}$, $\mathsf{Exists}_{\mathsf{pre}}^{\mathcal{O}}$, $\mathsf{Cred}_{\sigma}^{\mathcal{O}}(s)$ *and* $\mathsf{Cred}_{\mathsf{pre}}^{\mathcal{O}}(s)$ *are in* NP*;*
- $\mathsf{Exists}_{\mathsf{grd}}^{\mathcal{O}}$, $\mathsf{Cred}_{\mathsf{grd}}^{\mathcal{O}}(s)$, $\mathsf{Skept}_{\mathsf{grd}}^{\mathcal{O}}(s)$, $\mathsf{Skept}_{\mathsf{com}}^{\mathcal{O}}(s)$ *are in* P*;*
- $\mathsf{Skept}_{\mathsf{pre}}^{\mathcal{O}}(s)$ *is in* Π_2^P*.*

Proof. Membership is due to Theorem 8 and the complexity bounds of the operators in BADFs in Proposition 28, just note that $\Sigma_0^P = \Pi_0^P = \mathsf{P}$. $\mathsf{Ver}_{\mathsf{grd}}^{\mathcal{O}}(X,Y)$ is in $\mathsf{P}^{\mathsf{P}} = \mathsf{P}$ by Corollary 9. For the existence of non-trivial pairs we can simply guess and check in polynomial time for admissible pairs and thus also for complete and preferred semantics. □

Hardness results straightforwardly carry over from AFs.

Proposition 30. *Let Ξ be a BADF with $L = L^+ \cup L^-$, consider any operator $\mathcal{O} \in \{\mathcal{G}_{\sqsubseteq}, \mathcal{U}_{\sqsubseteq}\}$ and semantics $\sigma \in \{adm, com, pre\}$. For $X \subseteq Y \subseteq S$ and $s \in S$:*

- $\mathsf{Ver}_{\mathsf{pre}}^{\mathcal{O}}(X,Y)$ *is* coNP*-hard;*
- $\mathsf{Exists}_{\sigma}^{\mathcal{O}}$ *and* $\mathsf{Cred}_{\sigma}^{\mathcal{O}}(s)$ *are* NP*-hard;*
- $\mathsf{Skept}_{\mathsf{pre}}^{\mathcal{O}}(s)$ *is* Π_2^P*-hard.*

Proof. Hardness results from AFs for these problems carry over to BADFs as for all semantics AFs are a special case of BADFs (Brewka et al. 2013; Strass 2013a). The complexities of the problems on AFs for admissible and preferred semantics are shown by (Dimopoulos and Torres 1996), except for the Π_2^P-completeness result of skeptical preferred semantics, which is shown by (Dunne and Bench-Capon 2002). The complete semantics is studied by (Coste-Marquis, Devred, and Marquis 2005). □

We next show that there is no hope that the existence problems for approximate and ultimate two-valued stable models coincide as there are cases when the semantics differ.

Example 3. Consider the BADF $F = (S, L, C)$ with statements $S = \{a, b, c\}$ and acceptance formulas $\varphi_a = \mathbf{t}$, $\varphi_b = a \vee c$ and $\varphi_c = a \vee b$. The only two-valued supported model is (S, S) where all statements are true. This pair is also an ultimate two-valued stable model, since $\mathcal{U}_F'(\emptyset, S) = \{a\}$, and both $\varphi_b^{(\{a\},S)} = \mathbf{t} \vee c$ and $\varphi_c^{(\{a\},S)} = \mathbf{t} \vee b$ are tautologies, whence $\mathcal{U}_F'(\{a\}, S) = S$. However, (S, S) is not an approximate two-valued stable model: although $\mathcal{G}_F'(\emptyset, S) = \{a\}$, then $\mathcal{G}_F'(\{a\}, S) = \{a\}$ and we thus cannot reconstruct the upper bound S. Thus F has no approximate two-valued stable models.

So approximate and ultimate two-valued stable model semantics are indeed different. However, we can show that the respective existence problems have the same complexity.

Proposition 31. *Let Ξ be a BADF with $L = L^+ \cup L^-$, $\mathcal{O} \in \{\mathcal{G}_{\sqsubseteq}, \mathcal{U}_{\sqsubseteq}\}$ an operator and semantics $\sigma \in \{2su, 2st\}$. For $X \subseteq S$, $\mathsf{Ver}_{\sigma}^{\mathcal{O}}(X,X)$ is in P; $\mathsf{Exists}_{\sigma}^{\mathcal{O}}$ is NP-complete.*

σ	adm	com	pre	grd	2su	2st
$\mathsf{Ver}_{\sigma}^{\mathcal{O}}$	in P	in P	coNP-c	in P	in P	in P
$\mathsf{Exists}_{\sigma}^{\mathcal{O}}$	NP-c	NP-c	NP-c	in P	NP-c	NP-c
$\mathsf{Cred}_{\sigma}^{\mathcal{O}}$	NP-c	NP-c	NP-c	in P	NP-c	NP-c
$\mathsf{Skept}_{\sigma}^{\mathcal{O}}$	trivial	in P	Π_2^P-c	in P	coNP-c	coNP-c

Table 3: Complexity results for semantics of bipolar Abstract Dialectical Frameworks for $\mathcal{O} \in \{\mathcal{G}_{\sqsubseteq}, \mathcal{U}_{\sqsubseteq}\}$.

Proof. Membership carries over – for supported models from (Brewka et al. 2013, Proposition 5), for approximate stable models from Theorem 22. For membership for ultimate stable models, we can use Proposition 28 to adapt the decision procedure of Proposition 21. In any case, hardness carries over from AFs (Dimopoulos and Torres 1996). □

For credulous and skeptical reasoning over the two-valued semantics, membership is straightforward and hardness again carries over from argumentation frameworks.

Corollary 32. *Let Ξ be a BADF with $L = L^+ \cup L^-$; consider any operator $\mathcal{O} \in \{\mathcal{G}_{\sqsubseteq}, \mathcal{U}_{\sqsubseteq}\}$ and semantics $\sigma \in \{2su, 2st\}$. For $s \in S$, $\mathsf{Cred}_{\sigma}^{\mathcal{O}}(s)$ is NP-complete; $\mathsf{Skept}_{\sigma}^{\mathcal{O}}(s)$ is coNP-complete.*

Discussion

In this paper we studied the computational complexity of abstract dialectical frameworks using approximation fixpoint theory. We showed numerous novel results for two families of ADF semantics, the approximate and ultimate semantics, which are themselves inspired by argumentation and AFT. We showed that in most cases the complexity increases by one level of the polynomial hierarchy compared to the corresponding reasoning tasks on AFs. A notable difference between the two families of semantics lies in the stable semantics, where the approximate version is easier than its ultimate counterpart. For the restricted, yet powerful class of bipolar ADFs we proved that for the corresponding reasoning tasks AFs and BADFs have the same complexity, which suggests that many types of relations between arguments can be introduced without increasing the worst-time complexity. On the other hand, our results again emphasize that arbitrary (non-bipolar) ADFs cannot be compiled into equivalent Dung AFs in deterministic polynomial time, unless the polynomial hierarchy collapses to the first level. Under the same assumption, ADFs cannot be implemented directly with methods that are typically applied to AFs, for example answer-set programming (Egly, Gaggl, and Woltran 2010).

Our results lay the foundation for future algorithms and their implementation, for example augmenting the ADF system DIAMOND (Ellmauthaler and Strass 2013) to support also the approximate semantics family, as well as devising efficient methods for the interesting class of BADFs.

For further future work several promising directions are possible. Studying easier fragments of ADFs as well as parameterized complexity analysis can lead to efficient decision procedures, as is witnessed for AFs (Dvořák et al. 2014; Dvořák, Ordyniak, and Szeider 2012). We also deem it auspicious to use alternative representations of acceptance conditions, for instance by employing techniques from knowledge compilation (Darwiche and Marquis 2002).

A detailed complexity analysis of other useful AF semantics would also reveal further insights, e.g. semi-stable semantics (Caminada, Carnielli, and Dunne 2012), naive-based semantics, such as cf2 (Baroni, Giacomin, and Guida 2005), or a recently proposed extension-based semantics for ADFs (Polberg, Wallner, and Woltran 2013). For semantical analysis, it would be useful to consider principle-based evaluations for ADFs (Baroni and Giacomin 2007). Furthermore it appears natural to compare (ultimate) ADF semantics and ultimate logic programming semantics (Denecker, Marek, and Truszczyński 2004) in approximation fixpoint theory, in particular with respect to computational complexity.

Acknowledgements. Thanks to an anonymous reviewer for pointing out a notational issue in our usage of the complexity classes D_i^P. This research was supported by DFG (project BR 1817/7-1) and FWF (project I1102).

References

Baroni, P., and Giacomin, M. 2007. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence* 171(10-15):675–700.

Baroni, P.; Giacomin, M.; and Guida, G. 2005. SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence* 168(1-2):162–210.

Bench-Capon, T. J. M., and Dunne, P. E. 2007. Argumentation in Artificial Intelligence. *Artificial Intelligence* 171(10-15):619–641.

Brewka, G., and Gordon, T. F. 2010. Carneades and Abstract Dialectical Frameworks: A Reconstruction. In *Proceedings of COMMA 2010*, volume 216 of *FAIA*, 3–12. IOS Press.

Brewka, G., and Woltran, S. 2010. Abstract Dialectical Frameworks. In *Proceedings of KR 2010*, 102–111. AAAI Press.

Brewka, G.; Ellmauthaler, S.; Strass, H.; Wallner, J. P.; and Woltran, S. 2013. Abstract Dialectical Frameworks Revisited. In *Proceedings of IJCAI 2013*, 803–809. IJCAI/AAAI.

Brewka, G.; Dunne, P. E.; and Woltran, S. 2011. Relating the Semantics of Abstract Dialectical Frameworks and Standard AFs. In *Proceedings of IJCAI 2011*, 780–785. IJCAI/AAAI.

Brewka, G.; Polberg, S.; and Woltran, S. 2013. Generalizations of Dung frameworks and their role in formal argumentation. *IEEE Intelligent Systems* PP(99). Special Issue on Representation and Reasoning. In press.

Caminada, M. W., and Amgoud, L. 2007. On the evaluation of argumentation formalisms. *Artificial Intelligence* 171(5–6):286–310.

Caminada, M. W.; Carnielli, W. A.; and Dunne, P. E. 2012. Semi-stable Semantics. *Journal of Logic and Computation* 22(5):1207–1254.

Coste-Marquis, S.; Devred, C.; and Marquis, P. 2005. Symmetric Argumentation Frameworks. In *ECSQARU*, volume 3571 of *LNCS*, 317–328. Springer.

Darwiche, A., and Marquis, P. 2002. A Knowledge Compilation Map. *Journal of Artificial Intelligence Research* 17:229–264.

Davey, B., and Priestley, H. 2002. *Introduction to Lattices and Order*. Cambridge University Press, second edition.

Denecker, M.; Marek, V. W.; and Truszczyński, M. 2000. Approximations, Stable Operators, Well-Founded Fixpoints and Applications in Nonmonotonic Reasoning. In *Logic-Based Artificial Intelligence*. Kluwer Academic Publishers. 127–144.

Denecker, M.; Marek, V. W.; and Truszczyński, M. 2003. Uniform Semantic Treatment of Default and Autoepistemic Logics. *Artificial Intelligence* 143(1):79–122.

Denecker, M.; Marek, V. W.; and Truszczyński, M. 2004. Ultimate approximation and its application in nonmonotonic knowledge representation systems. *Information and Computation* 192(1):84–121.

Dimopoulos, Y., and Torres, A. 1996. Graph Theoretical Structures in Logic Programs and Default Theories. *Theoretical Computer Science* 170(1-2):209–244.

Dimopoulos, Y.; Nebel, B.; and Toni, F. 2002. On the computational complexity of assumption-based argumentation for default reasoning. *Artificial Intelligence* 141(1/2):57–78.

Dung, P. M. 1995. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence* 77(2):321–358.

Dunne, P. E., and Bench-Capon, T. J. M. 2002. Coherence in Finite Argument Systems. *Artificial Intelligence* 141(1/2):187–203.

Dunne, P. E., and Wooldridge, M. 2009. Complexity of abstract argumentation. In Simari, G., and Rahwan, I., eds., *Argumentation in Artificial Intelligence*. Springer. 85–104.

Dvořák, W.; Järvisalo, M.; Wallner, J. P.; and Woltran, S. 2014. Complexity-Sensitive Decision Procedures for Abstract Argumentation. *Artificial Intelligence* 206:53–78.

Dvořák, W.; Ordyniak, S.; and Szeider, S. 2012. Augmenting tractable fragments of abstract argumentation. *Artificial Intelligence* 186:157–173.

Egly, U.; Gaggl, S. A.; and Woltran, S. 2010. Answer-set programming encodings for argumentation frameworks. *Argument and Computation* 1(2):147–177.

Ellmauthaler, S., and Strass, H. 2013. The DIAMOND System for Argumentation: Preliminary Report. In *Proceedings of ASPOCP*, 97–107.

Gordon, T. F.; Prakken, H.; and Walton, D. 2007. The Carneades model of argument and burden of proof. *Artificial Intelligence* 171(10–15):875–896.

Polberg, S.; Wallner, J. P.; and Woltran, S. 2013. Admissibility in the Abstract Dialectical Framework. In *Proceedings of CLIMA XIV*, volume 8143 of *LNAI*, 102–118. Springer.

Strass, H., and Wallner, J. P. 2013. Analyzing the Computational Complexity of Abstract Dialectical Frameworks via Approximation Fixpoint Theory. Technical Report 2, Computer Science Institute, Leipzig University. http://nbn-resolving.de/urn:nbn:de:bsz:15-qucosa-129614.

Strass, H. 2013a. Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence* 205:39–70.

Strass, H. 2013b. Instantiating knowledge bases in abstract dialectical frameworks. In *Proceedings of CLIMA XIV*, volume 8143 of *LNAI*, 86–101. Springer.

Van Gijzel, B., and Prakken, H. 2011. Relating Carneades with abstract argumentation. In *Proceedings of IJCAI 2011*, 1113–1119. IJCAI/AAAI.

Wyner, A.; Bench-Capon, T. J. M.; and Dunne, P. E. 2013. On the instantiation of knowledge bases in abstract argumentation frameworks. In *Proceedings of CLIMA XIV*, volume 8143 of *LNAI*, 34–50. Springer.