This manuscript was published as:
Thomas Linsbichler, Marco Maratea, Andreas Niskanen, Johannes P. Wallner, and Stefan Woltran. Novel Algorithms for Abstract Dialectical Frameworks based on Complexity Analysis of Subclasses and SAT Solving. In Jérôme Lang, editor, *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 1905–1911, Stockholm, Sweden, July 2018. IJCAI. The final publication is available in the proceedings of the International Joint Conferences on Artificial Intelligence via `https://doi.org/10.24963/ijcai.2018/263`. The proceedings are available at `http://www.ijcai.org`.

Novel Algorithms for Abstract Dialectical Frameworks based on Complexity Analysis of Subclasses and SAT Solving

Thomas Linsbichler[1], **Marco Maratea**[2], **Andreas Niskanen**[3], **Johannes P. Wallner**[1], **Stefan Woltran**[1]
[1] Institute of Logic and Computation, TU Wien, Austria,
[2] DIBRIS, University of Genova, Italy,
[3] HIIT, Department of Computer Science, University of Helsinki, Finland

Abstract

Abstract dialectical frameworks (ADFs) constitute one of the most powerful formalisms in abstract argumentation. Their high computational complexity poses, however, certain challenges when designing efficient systems. In this paper, we tackle this issue by (i) analyzing the complexity of ADFs under structural restrictions, (ii) presenting novel algorithms which make use of these insights, and (iii) empirically evaluating a resulting implementation which relies on calls to SAT solvers.

1 Introduction

Over the last decade, argumentation has become a major field in AI research [Bench-Capon and Dunne, 2007] and an increasing number of applications (see [Atkinson *et al.*, 2017] for a survey) encourages the development of computational models of argumentation. Within this field, Dung's abstract argumentation frameworks (AFs for short) [Dung, 1995] have received notable attention which resulted in a multitude of methods for solving reasoning problems in such frameworks [Charwat *et al.*, 2015]. Systems implementing such methods participate in a dedicated competition [Thimm and Villata, 2017], where procedures which rely on (multiple) calls to SAT solvers have been proven successful [Dvořák *et al.*, 2014; Cerutti *et al.*, 2014].

Despite the popularity of AFs, their conceptual simplicity – an AF is just a directed graph with nodes as abstract arguments and edges representing individual attacks – has also often turned out to be overly limiting. Therefore, generalizations of AFs introducing further link types such as support or collective attacks [Nielsen and Parsons, 2007; Cayrol and Lagasquie-Schiex, 2013] or imposing quantitative aspects have been presented. Abstract dialectical frameworks (ADFs for short) [Brewka and Woltran, 2010; Brewka *et al.*, 2017b] constitute one of the most comprehensive generalizations of AFs available. By equipping every argument with an acceptance condition, ADFs are capable of modeling arbitrary relations between arguments. Potential applications in legal reasoning [Al-Abdulkarim *et al.*, 2016], online dialog systems [Neugebauer, 2017], text exploration [Cabrio and Villata, 2016], and the instantiation of defeasible theories [Strass, 2014] have already been proposed.

The power of ADFs comes with a price: compared to AFs, the computational complexity increases by one step in the polynomial hierarchy for nearly all reasoning tasks [Strass and Wallner, 2015], with the hardest problems being Π_3^{P}-complete. Therefore, the identification of fragments which yield decreased complexity, or even tractability, is of high interest. So far, only the class of bipolar ADFs [Brewka and Woltran, 2010] has been shown to be unaffected by the increase in complexity compared to AFs. Recent work [Keshavarzi, 2017] has investigated further subclasses but without an explicit complexity analysis. Existing implementations of ADFs deal with the complexity by employing reductions either to QBF [Diller *et al.*, 2015] or to ASP [Strass and Ellmauthaler, 2017; Brewka *et al.*, 2017a], but barely exploit potential shortcuts for ADFs of lower complexity.

In this work, we aim towards more efficient systems for ADFs and provide (1) a careful complexity analysis of ADF subclasses together with investigations whether ADFs being "close" to a subclass remain easier; and (2) SAT-based implementations which make use of these insights. For AFs such analyses can be found in [Dvořák *et al.*, 2012; 2014] and have been implemented in the Cegartix system.

More specifically, our main contributions are as follows:

- We investigate the complexity of two further subclasses: acyclic ADFs and concise ADFs. The first class, which restricts the graph underlying the ADF to be acyclic, yields dramatic gains in complexity for the majority of reasoning tasks. The second class is defined with respect to a semantics for which the ADFs are required to yield only a single solution. This also leads to a certain decrease in the complexity. As all these fragments impose rather strict requirements, we further show that many of the computational advantages also hold for ADFs which have constant distance to these subclasses (for this analysis we also include bipolar ADFs).
- Based on our observations we design novel algorithms for ADF reasoning, in particular, for the problems of skeptical and credulous acceptance under preferred semantics. We provide an implementation based on incremental SAT solving, which is not only novel in combining ADF solving with SAT, but is also complexity sensitive as it takes advantage of the input ADFs being in (low distance to) advantageous fragments.
- Finally, we provide an experimental evaluation of our new approach which shows that it is capable of outperforming state-of-the-art systems for ADFs.

2 Background

We recall the syntax and semantics of ADFs from [Brewka *et al.*, 2017b]. We assume a fixed and finite set of arguments A. An interpretation is a function I mapping arguments to one of the three truth values $I : A \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. That is, an interpretation maps each argument to either true ($\mathbf{t}$), false ($\mathbf{f}$), or undefined ($\mathbf{u}$). An interpretation I is *two-valued* if $I(a) \in \{\mathbf{t}, \mathbf{f}\}$ for all $a \in A$, and *trivial* (denoted $I_\mathbf{u}$) if $I(a) = \mathbf{u}$ for all $a \in A$. For a two-valued interpretation I, $I(\varphi)$ extends to the evaluation of a formula φ under I as usual. I is equally or more informative than J, denoted by $J \leq_i I$, if $J(a) \in \{\mathbf{t}, \mathbf{f}\}$ implies $J(a) = I(a)$ for all $a \in A$. We denote by $<_i$ the strict version of $\leq_i$, i.e., $J <_i I$ if $J \leq_i I$ and $\exists a \in A$ s.t. $J(a) = \mathbf{u}$ and $I(a) \in \{\mathbf{t}, \mathbf{f}\}$.

An ADF is a tuple $D = (A, L, C)$ where A is a set of arguments, $L \subseteq A \times A$ is a set of links, and $C = \{\varphi_a\}_{a \in A}$ is a collection of acceptance conditions for arguments $a \in A$, each given by a propositional formula over the parents of an argument: $par_D(a) = \{b \in A \mid (b, a) \in L\}$.

The semantics of ADFs are based on the characteristic operator Γ_D mapping interpretations to updated interpretations and defined for an ADF D as $\Gamma_D(I) = J$ with

$$J(a) = \begin{cases} \mathbf{t} \text{ if } \varphi_a[I] \text{ is a tautology} \\ \mathbf{f} \text{ if } \varphi_a[I] \text{ is unsatisfiable} \\ \mathbf{u} \text{ otherwise} \end{cases}$$

where $\varphi[I]$ is the formula obtained from φ with each argument that I assigns to either true or false being replaced by the corresponding truth constant, i.e., $\varphi[I] = \varphi[x \mapsto \top \mid I(x) = \mathbf{t}][x \mapsto \bot \mid I(x) = \mathbf{f}]$; arguments assigned to undefined are not modified in the resulting formula.

The intuition behind the characteristic operator is that an argument a is assigned true (false) if all completions of the given interpretation I satisfy (do not satisfy) the acceptance condition of a. A completion of an interpretation I is a two-valued interpretation J such that $I \leq_i J$.

Definition 2.1. *Given an ADF D, an interpretation I*

- *is admissible in D iff $I \leq_i \Gamma_D(I)$;*
- *is complete in D iff $I = \Gamma_D(I)$;*
- *is grounded in D iff I is the least fixed-point of Γ_D;*
- *is preferred in D iff I is $\leq_i$-maximal admissible in D.*

Conflict-free interpretations are defined by weakening the condition of admissible interpretations: I is conflict-free in D if, for all $a \in A$, $I(a) = \mathbf{t}$ implies $\varphi_a[I]$ is satisfiable, and $I(a) = \mathbf{f}$ implies $\varphi_a[I]$ is refutable (a formula is refutable if it is not a tautology). This differs from admissibility by requiring satisfiability instead of a tautology and refutability instead of unsatisfiability. A conflict-free interpretation I is a naive interpretation in D if there is no conflict-free interpretation J in D s.t. $I <_i J$. We refer to the set of all conflict-free, naive, admissible, complete, grounded, and preferred interpretations of an ADF D as *cf*(D), *nai*(D), *adm*(D), *com*(D), *grd*(D), and *prf*(D), respectively. In any ADF D, it holds that *prf*$(D) \subseteq$ *com*$(D) \subseteq$ *adm*$(D) \subseteq$ *cf*(D).

We denote the update of an interpretation I with truth value $\mathbf{x} \in \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ for argument b by $I|^b_\mathbf{x}$, i.e., $I|^b_\mathbf{x}(b) = \mathbf{x}$ and $I|^b_\mathbf{x}(a) = I(a)$ for $a \neq b$. In an ADF $D = (A, L, C)$, a link $(b, a) \in L$ is called (i) *supporting* (in D) if for every two-valued interpretation I, $I(\varphi_a) = \mathbf{t}$ implies $I|^b_\mathbf{t}(\varphi_a) = \mathbf{t}$; (ii) *attacking* (in D) if for every two-valued interpretation I, $I(\varphi_a) = \mathbf{f}$ implies $I|^b_\mathbf{t}(\varphi_a) = \mathbf{f}$. D is called *bipolar* if each link $(b, a) \in L$ is attacking or supporting.

σ	*Cred*$_\sigma$	*Skept*$_\sigma$	*Exists*$_\sigma$	*Ver*$_\sigma$
cf	NP-c	trivial	NP-c	NP-c
nai	NP-c	Π_2^P-c	NP-c	DP-c
adm	Σ_2^P-c	trivial	Σ_2^P-c	coNP-c
grd	coNP-c	coNP-c	coNP-c	DP-c
com	Σ_2^P-c	coNP-c	Σ_2^P-c	DP-c
prf	Σ_2^P-c	Π_3^P-c	Σ_2^P-c	Π_2^P-c

Table 1: Complexity of ADF reasoning [Strass and Wallner, 2015].

The most important reasoning problems in ADFs are credulous and skeptical acceptance, existence of a non-trivial interpretation, and verification of an interpretation. They are defined as follows for semantics σ, given ADF $D{=}(A, L, C)$:

- *Cred*$_\sigma$: Given $a \in A$, is there an $I \in \sigma(D)$ s.t. $I(a) = \mathbf{t}$?
- *Skept*$_\sigma$: Given $a \in A$, is $I(a) = \mathbf{t}$ for all $I \in \sigma(D)$?
- *Exists*$_\sigma$: Is there an $I \in \sigma(D)$ with $I \neq I_\mathbf{u}$?
- *Ver*$_\sigma$: Given an interpretation I, is $I \in \sigma(D)$?

Example 2.2. *Let $D = (\{a, b, c\}, L, C)$ be an ADF with $\varphi_a = b$, $\varphi_b = a$, and $\varphi_c = \neg a$. It holds that $I = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}\}$ and $I' = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{f}, c \mapsto \mathbf{t}\}$ are preferred in D and $I_\mathbf{u}$ is grounded in D. Further, a is credulously but not skeptically accepted under preferred semantics.*

The known complexity results are shown in Table 1.

3 Complexity of Subclasses

In this section we present our complexity results for subclasses of ADFs, which are summarized in Table 2 (top).

Bipolar ADFs. It has been shown that reasoning in bipolar ADFs is easier [Strass and Wallner, 2015], when for each link its type is known. The results are shown in Table 2 (top left).

Acyclic ADFs. This class is defined as follows.

Definition 3.1. *An ADF $D = (A, L, C)$ is acyclic if the directed graph (A, L) is acyclic.*

In acyclic ADFs we have arguments without parents (initial arguments) whose acceptance conditions are constant. This implies that $|$*com*$(D)| = 1$ [Keshavarzi, 2017], which follows from a bottom-up propagation of the acceptance conditions.

Proposition 3.2 ([Keshavarzi, 2017])**.** *Any acyclic ADF has a unique two-valued complete interpretation.*

Based on this result, one can compute the grounded (unique preferred) interpretation in polynomial time. Furthermore, credulous and skeptical reasoning coincide for complete based semantics and can be decided in polynomial time.

Corollary 3.3. *For acyclic ADFs, Cred$_\sigma$, Skept$_\sigma$, Ver$_\tau$ are in* P *for $\sigma \in \{adm, grd, com, prf\}$ and $\tau \in \{grd, com, prf\}$.*

	bipolar [Strass and Wallner, 2015]				acyclic				σ-concise			
σ	*Cred*$_\sigma$	*Skept*$_\sigma$	*Exists*$_\sigma$	*Ver*$_\sigma$	*Cred*$_\sigma$	*Skept*$_\sigma$	*Exists*$_\sigma$	*Ver*$_\sigma$	*Cred*$_\sigma$	*Skept*$_\sigma$	*Exists*$_\sigma$	*Ver*$_\sigma$
cf	in P	trivial	in P	in P	NP-c	trivial	trivial	NP-c	trivial	trivial	trivial	trivial
nai	in P	coNP-c	in P	in P	NP-c	coNP-h	trivial	DP-c	NP-c	NP-c	NP-c	DP-c
adm	NP-c	trivial	NP-c	in P	in P	trivial	in P	coNP-c	trivial	trivial	trivial	trivial
grd	in P	in P	in P	in P	in P	in P	in P	in P	coNP-c	coNP-c	coNP-c	DP-c
com	NP-c	in P	NP-c	in P	in P	in P	in P	in P	coNP-c	coNP-c	coNP-c	DP-c
prf	NP-c	Π_2^{P}-c	NP-c	coNP-c	in P	in P	in P	in P	Σ_2^{P}-c*	Σ_2^{P}-c*	Σ_2^{P}-c*	Π_2^{P}-c*
	k-bipolar				k-acyclic				k-σ-concise			
cf	in P	trivial	in P	in P	NP-c	trivial	in P	NP-c	NP-c	trivial	NP-c	NP-c
nai	in P	coNP-c	in P	in P	NP-c	coNP-h	in P	DP-c	NP-c	in Δ_2^{P}	NP-c	DP-c
adm	NP-c	trivial	NP-c	in P	in Δ_2^{P}	trivial	in Δ_2^{P}	coNP-c	Σ_2^{P}-c*	trivial	Σ_2^{P}-c*	coNP-c
grd	in P	in P	in P	in P	coNP-c	coNP-c	coNP-c	coNP-h	coNP-c	coNP-c	coNP-c	DP-c
com	NP-c	in P	NP-c	in P	in Δ_2^{P}	coNP-c	in Δ_2^{P}	coNP-h	Σ_2^{P}-c*	coNP-c	Σ_2^{P}-c*	DP-c
prf	NP-c	Π_2^{P}-c	NP-c	coNP-c	in Δ_2^{P}	in Δ_2^{P}	in Δ_2^{P}	in Δ_2^{P}	Σ_2^{P}-c*	in Δ_3^{P}	Σ_2^{P}-c*	Π_2^{P}-c*

Table 2: Complexity of subclasses of ADFs. $\mathcal{C}^*$ means that hardness for $\mathcal{C}$ is under randomized reductions.

Acyclicity does not help in verifying whether an interpretation is admissible (or conflict-free): e.g., asking whether it is admissible to assign a non-initial argument to true and its parents to undefined still requires to consider all completions.

Proposition 3.4. *For acyclic ADFs, Ver$_{adm}$ is* coNP*-complete and Ver$_{cf}$ is* NP*-complete.*

However, existence of non-trivial conflict-free or naive interpretations is trivial: an initial argument has a constant acceptance condition (can always be true or false).

In contrast, credulous acceptance under *cf* and *nai* remains as hard as in the general case. An argument may be false in the unique preferred interpretation, but true in a naive one.

Example 3.5. *Let $D = (X \cup \{y\}, L, C)$ be an ADF with $\varphi_x = \top$ for $x \in X$ and $\varphi_y = \phi$. Then, $\{x \mapsto \mathbf{u} \mid x \in X\} \cup \{y \mapsto \mathbf{t}\} \in cf(D)$ iff ϕ is satisfiable ($J(y) = \mathbf{f}$ for $J \in prf(D)$ if ϕ evaluates to $\mathbf{f}$ when all X are assigned true).*

Proposition 3.6. *For acyclic ADFs, Cred$_\sigma$ is* NP*-complete ($\sigma \in \{cf, nai\}$); Ver$_{nai}$ is* DP*-complete; Skept$_{nai}$ is* coNP*-hard.*

Concise ADFs. The subclass of concise ADFs is characterized by the existence of exactly one σ-interpretation.

Definition 3.7. *An ADF D is concise for semantics σ (or σ-concise) if $|\sigma(D)| = 1$.*

For *cf*-concise (resp. *adm*-concise) ADFs there is only one interpretation that is conflict-free (resp. admissible), namely $I_{\mathbf{u}}$. Hence, problems *Cred*$_\sigma$, *Skept*$_\sigma$, *Exists*$_\sigma$ ($\sigma \in \{cf, adm\}$) are always false, while *Ver*$_\sigma$ is positive only for $I_{\mathbf{u}}$.

Proposition 3.8. *For $\sigma \in \{cf, adm\}$, it holds that Cred$_\sigma$, Skept$_\sigma$, Exists$_\sigma$, and Ver$_\sigma$ are trivial for σ-concise ADFs.*

In σ-concise ADFs *Cred*$_\sigma$ and *Skept*$_\sigma$ coincide. Complexity under *nai*, except *Skept*$_{nai}$, does not deviate from ADFs.

Proposition 3.9. *For nai-concise ADFs, Cred$_{nai}$, Skept$_{nai}$, and Exists$_{nai}$ are* NP*-complete; Ver$_{nai}$ is* DP*-complete.*

Proof (Sketch). Given a formula ϕ over X, let $D_\phi = (X \cup \{z\}, L, C)$ with $\varphi_x = \neg x$ for $x \in X$, and $\varphi_z = \phi \vee \neg z$. For any $I \in cf(D_\phi)$ we have $I(x) = \mathbf{u}$. We have $I(z) = \mathbf{t}$ for some $I \in cf(D_\phi)$ iff $\varphi_z[I]$ is satisfiable iff ϕ is satisfiable. Finally, $I(z) \neq \mathbf{f}$ in every $I \in cf(D_\phi)$, as $\varphi_z[I]$ is then a tautology. Hence, $cf(D_\phi) = \{I_{\mathbf{u}}, \{z \mapsto \mathbf{t}\} \cup \{x \mapsto \mathbf{u} \mid x \in X\}\}$ if ϕ is satisfiable and $cf(D_\phi) = \{I_{\mathbf{u}}\}$ otherwise. ☐

Since grounded semantics is a unique-status semantics it holds that any ADF is *grd*-concise. Further, the grounded interpretation is the least complete interpretation, and, thus, reasoning under *grd* and *com* coincides for *com*-concise ADFs.

Proposition 3.10. *For com-concise ADFs, Cred$_{com}$, Skept$_{com}$, and Exists$_{com}$ are* coNP*-complete; Ver$_{com}$ is* DP*-complete.*

Due to Valiant and Vazirani [1986] the SAT problem is NP-hard under randomized reductions if the formula has at most one model. Briefly, these reductions produce a set of instances s.t. "yes" instances are translated to a set with a certain lower bound on the number of "yes" instances. The result was generalized to quantified Boolean formulas by Dunne [2009]. Using [Strass and Wallner, 2015, Reduction 3.2], we can show the following result.

Proposition 3.11. *For prf-concise ADFs, Cred$_{prf}$, Skept$_{prf}$, and Exists$_{prf}$ are Σ_2^{P}-complete under randomized reductions; Ver$_{prf}$ is Π_2^{P}-complete under randomized reductions.*

4 Distance to Subclasses

In this section we extend our subclasses to ones containing ADFs that have a constant distance to bipolar ADFs, acyclic ADFs, or concise ADFs (for results see Table 2 (bottom)).

Distance to Bipolar ADFs. We analyze the complexity of ADFs which have a limited number of non-bipolar links.

Definition 4.1. *An ADF D is k-bipolar for some integer $k \geq 0$ if for each $a \in A$ it holds that $|\{(b, a) \in L_D^?\}| \leq k$.*

For a k-bipolar ADF D, the (non-)polarity of links is known and L_D^-, L_D^+, and $L_D^?$ denote the links of D which are attacking, supporting and neither of the two, respectively.

The main ingredient for the complexity of k-bipolar ADFs is a generalization of why BADFs exhibit milder complexity [Strass and Wallner, 2015]: to test validity of $\varphi[I]$, fix the values of bipolar parents and check the remaining completions of I, the number of which is bounded by 2^k.

Lemma 4.2. *Given a k-bipolar ADF D, an interpretation I, and an argument $a \in A$, deciding whether $\varphi_a[I]$ is a tautology and deciding whether $\varphi_a[I]$ is unsatisfiable is in* P.

It follows that reasoning on k-bipolar and bipolar ADFs has the same complexity; the proof uses the generic results of [Strass and Wallner, 2015, Theorem 3.18] and Lemma 4.2.

Theorem 4.3. *Let $k \geq 0$ be a constant and $\sigma \in \{cf, nai, adm, com, grd, prf\}$. It holds that the complexity of Ver_σ, $Exists_\sigma$, $Skept_\sigma$, and $Cred_\sigma$ coincides for k-bipolar and bipolar ADFs.*

Distance to Acyclic ADFs. An ADF is k-acyclic if removing links from parents of k arguments results in acyclicity.

Definition 4.4. *An ADF $D = (A, L, C)$ is k-acyclic if there is a set $B \subseteq A$ with $|B| \leq k$ s.t. $(A, L \setminus (A \times B))$ is acyclic.*

By fixing a truth value assignment on B and then computing the grounded interpretation on the resulting acyclic ADF (via polynomially many NP oracle calls) we get a potential preferred interpretation of the original ADF. By trying out all such assignments (bounded by 3^k), we enumerate $prf(D)$.

Proposition 4.5. *For $k \geq 0$, $Cred_\sigma$, $Exists_\sigma$, $Skept_{prf}$, Ver_{prf} are in Δ_2^{P} for k-acyclic ADFs and $\sigma \in \{adm, com, prf\}$.*

Reasoning under *grd* is not easier on k-acyclic ADFs, since, intuitively, the undefined value may be propagated.

Proposition 4.6. *For 1-acyclic ADFs, it holds that $Cred_{grd}$, $Skept_{grd}$, and $Exists_{grd}$ are* coNP*-complete.*

The proof of this result relies on a small odd-attack-cycle so that an argument can be assigned to true iff a formula is unsatisfiable. This can be generalized to show the next results.

Corollary 4.7. *For 1-acyclic ADFs, $Cred_\sigma$, $Skept_\sigma$, $Exists_\sigma$, Ver_σ, and Ver_{grd} are* coNP*-hard for $\sigma \in \{adm, com, prf\}$.*

Existence of non-trivial conflict-free (naive) interpretations in k-acyclic ADFs is trivial if $k < |A|$ and decidable in polynomial time if $k \geq |A|$. In the former case, there is an argument not depending on itself (which can be assigned true or false without conflict). In the latter case, one can enumerate all truth value assignments and check for conflict-freeness.

Distance to Concise ADFs. An ADF has distance k to a σ-concise ADF if there are at most k σ-interpretations.

Definition 4.8. *An ADF D is k-σ-concise for a semantics σ and some integer $k \geq 0$ if $|\sigma(D)| \leq k$.*

In terms of complexity, membership carries over from general ADFs and hardness follows from concise ADFs. We can recycle the reduction from the proof of Proposition 3.9 to obtain the hardness part of the following result.

Proposition 4.9. *For k-cf-concise ADFs it holds that $Cred_{cf}$, $Exists_{cf}$, and Ver_{cf} are* NP*-complete for every $k \geq 2$.*

An upper bound on the number of interpretations is beneficial for skeptical acceptance. Under *prf*, it allows to decide the problem with a polynomial number of Σ_2^{P}-oracle calls. Key is that Σ_2^{P}-membership of $Exists_{adm}$ can be adapted to the problem $Exists_{adm}^{>}$, asking whether there is an *adm* interpretation strictly more informative than a given interpretation.

Theorem 4.10. *For k-prf-concise ADFs, $Skept_{prf}$ is in Δ_3^{P}.*

Proof (Sketch). Let $\mathcal{I} = \emptyset$. Construct a preferred interpretation I of $D = (A, L, C)$ by starting with $I = I_{\mathbf{u}}$ and ask a Σ_2^{P} oracle whether there is a $J \in adm(D)$ with $J >_i I$ and no $J' \in \mathcal{I}$ with $J \leq_i J'$. If yes, set $I = J$ and repeat the oracle call. Otherwise, if $I \neq I_{\mathbf{u}}$, add I to $\mathcal{I}$ and proceed to find the next preferred interpretation. $|prf(D)|$ is constant and a linear number of calls suffices to find a preferred interpretation. □

Corollary 4.11. *For k-nai-concise ADFs, $Skept_{nai}$ is in Δ_2^{P}.*

For admissible and complete semantics, we consider it rather unlikely that $Exists_\sigma$ or $Cred_\sigma$ get easier for k-σ-concise ADFs with $k \geq 2$. While we are not able to show Σ_2^{P}-hardness under standard reductions, we can show it under randomized reductions giving rise to a slightly weaker result.

Proposition 4.12. *Let $\sigma \in \{adm, com\}$. For k-σ-concise ADFs it holds that $Exists_\sigma$ and $Cred_\sigma$ are Σ_2^{P}-complete under randomized reductions for every $k \geq 2$.*

Finally, verifying admissibility of an interpretation does also not benefit from a bounded number of interpretations.

Proposition 4.13. *For k-adm-concise ADFs it holds that Ver_{adm} is* coNP*-complete for every $k \geq 2$.*

Inspecting Theorem 4.3, the only problem for (k-)bipolar which is hard for the second level of the polynomial hierarchy is $Skept_{prf}$. If, in addition to bipolarity, also the number of preferred interpretations is bounded, we can again show that a polynomial number of NP-oracle calls is sufficient.

Theorem 4.14. *For k_1-bipolar and k_2-prf-concise ADFs it holds that $Skept_{prf}$ is in Δ_2^{P}.*

5 Algorithms

We now present novel algorithms for ADF reasoning tasks based on results from Sections 3 and 4. We focus on $Skept_{prf}$ and $Cred_{adm}$ (= $Cred_{prf}$), for which we present different approaches. The backbone of our algorithms are SAT solvers, to which we delegate (sub)problems that are in NP. A central subproblem is $Exists_\sigma^{>}$, which asks for a σ-interpretation strictly more informative than a given interpretation.

Proposition 5.1. *Given $\sigma \in \{cf, nai, grd, adm, com, prf\}$, complexity of $Exists_\sigma^{>}$ and $Exists_\sigma$ coincide for general, (k)-bipolar, and (k)-concise ADFs.*

In what follows, we first define and discuss the encodings utilized in SAT calls, then present the base algorithms for the considered problems, and finally outline different concrete solving procedures resulting from the algorithms.

Encodings. Figure 1 shows the building blocks for the encodings, given an ADF $D = (A, L, C)$ and an interpretation I. The formulas range over variables $U = \{s^{\mathbf{t}}, s^{\mathbf{f}} \mid s \in A\} \cup \{p_s^r \mid (r, s) \in L\}$. Moreover, $\varphi_s^{\downarrow}$ denotes the acceptance condition φ_s where each occurrence of $r \in par_D(s)$ is replaced by p_s^r. The encoding for $Exists_{cf}^{>}$ is then given by $\Phi_{cf}^{D,I} = \phi_{<_i}^{I} \wedge \bigwedge_{s \in A}(\phi_s^{\Gamma} \wedge \phi_s^{pr})$. It is satisfiable iff there is a $J \in cf(D)$ with $I <_i J$. Intuitively, atoms $s^{\mathbf{t}}$ and $s^{\mathbf{f}}$ represent the guess of an interpretation J, indicating that $J(s) = \mathbf{t}$

$$
\begin{aligned}
\phi^I_{<_i} =& \bigwedge_{s\in A}(\neg s^{\mathbf{t}} \vee \neg s^{\mathbf{f}}) \wedge \bigwedge_{I(s)=\mathbf{t}} s^{\mathbf{t}} \wedge \bigwedge_{I(s)=\mathbf{f}} s^{\mathbf{f}} \wedge \bigvee_{I(s)=\mathbf{u}} (s^{\mathbf{t}} \vee s^{\mathbf{f}}) \\
\phi^{\Gamma}_s =& (s^{\mathbf{t}} \to \varphi^{\downarrow}_s) \wedge (s^{\mathbf{f}} \to \neg\varphi^{\downarrow}_s) \\
\phi^{pr}_s =& \Big(s^{\mathbf{t}} \to \bigwedge_{(s,r)\in L} p^s_r\Big) \wedge \Big(s^{\mathbf{f}} \to \bigwedge_{(s,r)\in L} \neg p^s_r\Big) \\
\phi_{bip} =& \bigwedge_{r\in A} r^{\mathbf{t}} \to \Big(\bigwedge_{(s,r)\in L^{\ominus}_D} (\neg s^{\mathbf{f}} \to p^s_r) \wedge \bigwedge_{(s,r)\in L^{\oplus}_D} (\neg s^{\mathbf{t}} \to \neg p^s_r)\Big) \wedge \\
& r^{\mathbf{f}} \to \Big(\bigwedge_{(s,r)\in L^{\ominus}_D} (\neg s^{\mathbf{t}} \to \neg p^s_r) \wedge \bigwedge_{(s,r)\in L^{\oplus}_D} (\neg s^{\mathbf{f}} \to p^s_r)\Big) \\
\phi^{\Gamma_?}_s =& \Big(s^{\mathbf{t}} \to \Big(\bigwedge_{I_X\in\mathcal{V}(X_s)} \Big(\varphi^{\downarrow,I_X}_s \vee \bigvee_{I_X(r)=\mathbf{t}} r^{\mathbf{f}} \vee \bigvee_{I_X(r)=\mathbf{f}} r^{\mathbf{t}}\Big)\Big)\Big) \wedge \\
& \Big(s^{\mathbf{f}} \to \Big(\bigwedge_{I_X\in\mathcal{V}(X_s)} \Big(\neg\varphi^{\downarrow,I_X}_s \vee \bigvee_{I_X(r)=\mathbf{t}} r^{\mathbf{f}} \vee \bigvee_{I_X(r)=\mathbf{f}} r^{\mathbf{t}}\Big)\Big)\Big)
\end{aligned}
$$

Figure 1: Building blocks for encodings, given ADF D and interpretation I. Thereby, $L^{\ominus}_D = L^-_D \setminus L^+_D$ and $L^{\oplus}_D = L^+_D \setminus L^-_D$.

(resp. $J(s) = \mathbf{f}$). Subformula $\phi^I_{<_i}$ ensures $I <_i J$. In order to have $J \in \mathit{cf}(D)$, it must hold that, if $J(s) = \mathbf{t}$ (resp. $J(s) = \mathbf{f}$) then $\varphi_s[J]$ is satisfiable (resp. refutable). To this end, atoms p^r_s represent the guess of a two-valued interpretation J_s with $J_s \geq_i J$ (cf. ϕ^{pr}_s) aiming to satisfy (resp. refute) φ_s. By that, $\varphi^{\downarrow}_s$ being true (resp. false) is equivalent with $\varphi_s[J]$ being satisfiable (resp. refutable), as is encoded by ϕ^{Γ}_s.

For bipolar ADFs, $\mathit{Exists}^{>}_{adm}$ is in NP. We use the encoding $\Phi^{D,I}_{adm,bip} = \phi^I_{<_i} \wedge \phi_{bip} \wedge \bigwedge_{s\in A}(\phi^{\Gamma}_s \wedge \phi^{pr}_s)$ which is satisfiable iff there is a $J \in \mathit{adm}(D)$ with $I <_i J$. For $J \in \mathit{adm}(D)$, $\varphi_r[J]$ now has to evaluate to true (resp. false) under *every* completion of J if $J(r) = \mathbf{t}$ (resp. $J(r) = \mathbf{f}$). Due to bipolarity, it suffices to check this for one particular completion of J, represented by p^s_r for each $s \in A$. That is, if for some $s \in \mathit{par}_D(r)$ neither $s^{\mathbf{t}}$ nor $s^{\mathbf{f}}$ is true, we can set p^s_r according to the polarity of the link (s, r) and the value of $I(r)$, as done by ϕ_{bip}. If all $\varphi^{\downarrow}_s$ evaluate according to the guess J under this particular valuation, J is admissible (cf. ϕ^{Γ}_s).

By Thm. 4.3, $\mathit{Exists}^{>}_{adm} \in$ NP for k-bipolar ADFs (formula size exponential in k). For $s \in A$ let $X_s = \{r \mid \exists(r, s) \in L^?_D, I(r) = \mathbf{u}\}$ and $\mathcal{V}(X_s)$ be the set of two-valued interpretations over X_s. For $I_X \in \mathcal{V}(X_s)$, let $\varphi^{\downarrow,I_X}_s = \varphi_s[r \mapsto p^r_s \mid (r, s) \in L, r \notin X_s][r \mapsto I_X(r) \mid (r, s) \in L, r \in X_s]$. Let $\Phi^{D,I}_{adm,kbip} = \phi^I_{<_i} \wedge \phi_{bip} \wedge \bigwedge_{s\in A}(\phi^{\Gamma_k}_s \wedge \phi^{pr}_s)$, which differs from $\Phi^{D,I}_{adm,bip}$ by $\phi^{\Gamma_k}_s$, which is equal to ϕ^{Γ}_s if s has only bipolar incoming links and $\phi^{\Gamma_?}_s$ else. The values of the parents r connected through non-bipolar links (up to k; identified by p^r_s) are not determined by ϕ_{bip}. Instead, all completions I_X of those r with $J(r) = \mathbf{u}$ (approximated by X_s) must be checked to satisfy (resp. refute) $\varphi^{\downarrow,I_X}_s$. This check can be disregarded if the values of $I_X(r)$ and $J(x)$ differ for some non-bipolar parent r (done by the disjunctions in $\phi^{\Gamma_?}_s$).

Recall that the encodings not only decide the respective $\mathit{Exists}^{>}_{\sigma}$-problems, but also yield concrete interpretations $J \in \sigma(D)$ with $I <_i J$ via assignments to variables $s^{\mathbf{t}}$ and $s^{\mathbf{f}}$.

Checking admissibility of an interpretation is also solved via SAT. We encode the complement, $\neg \mathit{Ver}_{adm}$, by $\Phi^{D,I}_{\neg Ver_{adm}} = \bigwedge_{I(s)=\mathbf{t}} p^s \wedge \bigwedge_{I(s)=\mathbf{f}} \neg p^s \wedge \bigvee_{I(s)=\mathbf{t}} \neg\varphi^{\downarrow}_s \vee \bigvee_{I(s)=\mathbf{f}} \varphi^{\downarrow}_s$ with $\varphi^{\downarrow}_s = \varphi_s[r \mapsto p^r \mid (r, s) \in L]$. Variables p^s represent a guess of an interpretation $J \geq_i I$ that acts as a witness for $I \notin \mathit{adm}(D)$ if either $I(s) = \mathbf{t}$ but $\varphi_s[J] \equiv \mathbf{f}$ (then $\neg\varphi^{\downarrow}_s$ is true), or $I(s) = \mathbf{f}$ but $\varphi_s[J] \equiv \mathbf{t}$ (then $\varphi^{\downarrow}_s$ is true) for some $s \in A$. Then, $\Phi^{D,I}_{\neg Ver_{adm}}$ is satisfiable iff $I \notin \mathit{adm}(D)$.

Algorithm 1 $\mathit{Skept}_{prf}(D, \alpha)$

1: $\rho \leftarrow \emptyset$
2: **while** $I \leftarrow \mathit{Exists}^{>}_{adm}(D, I_{\mathbf{u}}, \rho)$ **do**
3:     **while** $I' \leftarrow \mathit{Exists}^{>}_{adm}(D, I, \rho)$ **do** $I \leftarrow I'$
4:     **if** $I(\alpha) \neq \mathbf{t}$ **then** *reject* **else** $\rho \leftarrow \rho \wedge \mathit{Refine}^{>}(I)$
5: **if** $\rho = \emptyset$ **then** *reject* **else** *accept*

Algorithms. Algorithm 1 solves Skept_{prf} (which is Π^{P}_3-c) by implementing the idea of the membership proof of Theorem 4.10. It iteratively calls $\mathit{Exists}^{>}_{adm}$ until a preferred interpretation is reached. Thereby, preferred interpretations already visited are ruled out by maintaining the formula ρ. This formula is refined in line 4 by $\mathit{Refine}^{>}(I) = \bigvee_{I(s)=\mathbf{t}} s^{\mathbf{f}} \vee \bigvee_{I(s)=\mathbf{f}} s^{\mathbf{t}} \vee \bigvee_{I(s)=\mathbf{u}}(s^{\mathbf{t}} \vee s^{\mathbf{f}})$, i.e., all $J \leq_i I$ are excluded from now on. This is done internally, enabling incremental calls to the SAT solver. The algorithm rejects an instance if it finds a preferred interpretation such that the query argument is not mapped to $\mathbf{t}$ (or if it finds $\mathit{prf}(D) = \{I_{\mathbf{u}}\}$) and accepts otherwise. By Theorem 4.10 the number of iterations in Algorithm 1 is polynomial in k for k-prf-concise ADFs, i.e., it is polynomially bounded by the number of interpretations.

Algorithm 2 implements the Σ^{P}_2-hard problem $\mathit{Exists}^{>}_{adm}$ for the general case. As an additional parameter, it takes the formula ρ encoding the search space of interpretations. Using the encoding $\Phi^{D,I}_{cf}$, a SAT solver is employed to get a conflict-free interpretation J. If J is then found to be admissible by another SAT call using $\Phi^{D,J}_{\neg Ver_{adm}}$, we can return J, otherwise we exclude J from the search space via refining ρ by $\mathit{Refine}^{\neq}(J) = \bigvee_{J(s)=\mathbf{t}} \neg s^{\mathbf{t}} \vee \bigvee_{J(s)=\mathbf{f}} \neg s^{\mathbf{f}} \vee \bigvee_{J(s)=\mathbf{u}}(s^{\mathbf{t}} \vee s^{\mathbf{f}})$.

Solvers. We denote Algorithm 1 by PRF-3, which uses Algorithm 2 for the $\mathit{Exists}^{>}_{adm}$ subprocedure, and employs three levels of SAT which is in line with worst-case complexity. If the given ADF is bipolar, a SAT call with $\Phi^{D,I}_{adm,bip}$ decides $\mathit{Exists}^{>}_{adm}$, giving rise to PRF-BIP. Otherwise, the polarity of each link can be determined by two SAT calls as a preprocessing step, enabling $\mathit{Exists}^{>}_{adm}$ to be decidable via a SAT

Algorithm 2 $\mathit{Exists}^{>}_{adm}(D, I, \rho)$

$\rho \leftarrow \rho \wedge \Phi^{D,I}_{cf}$
while true **do**
    $(\tau, sat) \leftarrow \mathrm{SAT}(\rho)$
    if sat **then**
        $J \leftarrow \mathit{Extract}(\tau)$ # get interpretation by values of $s^{\mathbf{t}}$, $s^{\mathbf{f}}$
        if $\mathrm{SAT}(\Phi^{D,J}_{\neg Ver_{adm}})$ **then** $\rho \leftarrow \rho \wedge \mathit{Refine}^{\neq}(J)$ **else return** J
    else return *fail*

	Cred$_{adm}$					*Skept*$_{prf}$					
Domain	ADM-K-BIP	ADM-2	goDiamond	QADF	YADF	PRF-K-BIP-OPT	PRF-K-BIP	PRF-3	goDiamond	QADF	YADF
ABA2AF	**11.20** (15)	16.12 (**12**)	19.84 (54)	40.58 (55)	78.92 (75)	25.90 (**16**)	37.38 (**16**)	34.84 (42)	68.06 (54)	**18.13** (86)	165.74 (75)
Planning	**0.14** (**0**)	13.95 (14)	6.72 (**0**)	4.6 (87)	193.25 (47)	**11.14** (3)	30.23 (5)	135.18 (73)	17.52 (**0**)	NA (100)	281.32 (84)
Traffic	**0.05** (**0**)	10.62 (9)	6.42 (**0**)	10.56 (37)	35.47 (21)	17.18 (1)	25.61 (1)	**12.33** (35)	28.42 (**0**)	172.88 (82)	110.77 (42)

Table 3: Mean running times for ADF reasoning tasks (timeouts excluded). Number of timeouts (out of 100 instances) in brackets.

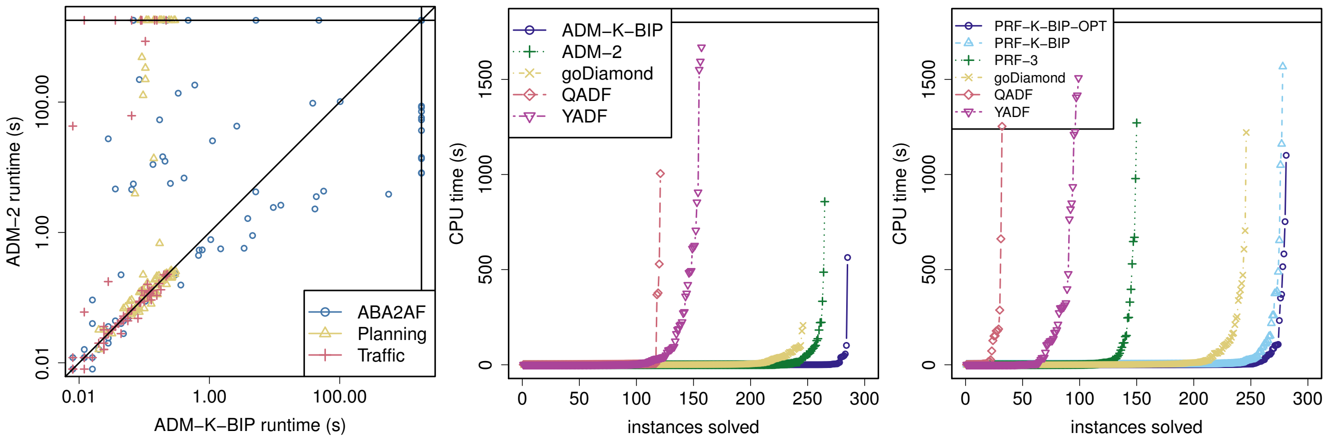

Figure 2: Running times for ADM-K-BIP vs. ADM-2 on *Cred*$_{adm}$ (left); solver comparison for *Cred*$_{adm}$ (middle) and *Skept*$_{prf}$ (right).

call using $\Phi^{D,I}_{adm,kbip}$. We call this variant PRF-K-BIP. Adding shortcuts by (i) computing *Cred*$_{adm}$ (rejecting if not accepted) and (ii) pre-computing $grd(D)$ gives rise to PRF-K-BIP-OPT.

For *Cred*$_{adm}$ with query argument s we get the procedure ADM-2 by adding $s^{\mathbf{t}}$ to $\Phi^{D,I}_{cf}$ in Algorithm 2 with parameters $I = I_{\mathbf{u}}$ and $\rho = \emptyset$. In the k-bipolar case ADM-K-BIP decides the problem by a single SAT call using $\Phi^{D,I_{\mathbf{u}}}_{adm,kbip} \wedge s^{\mathbf{t}}$.

6 Experiments

We implemented above algorithms resulting in the system k++ADF, available in open source at `www.cs.helsinki.fi/group/coreo/k++adf`. The implementation currently supports credulous and skeptical reasoning, and enumeration, under the semantics *cf*, *nai*, *adm*, *com*, *prf*, and *grd*, and includes MiniSAT 2.2.0 [Eén and Sörensson, 2003] as the SAT solver. Here, we present an overview of the scalability of the approach, and a comparison to existing ADF systems.

We selected the most challenging instances from the YADF system page (`www.dbai.tuwien.ac.at/proj/adf/yadf/`). These ADFs were generated from the AFs of domains ABA2AF, Planning, and Traffic from ICCMA 2017 (`argumentationcompetition.org/2017/`) as follows [Brewka *et al.*, 2017a]. Given AF graphs, acceptance conditions are constructed by splitting parents into 5 groups, each constituting a subformula representing attack, group-attack, support, group-support, and exclusive or. The subfomulas are connected via $\wedge$ or $\vee$ with equal probability, resulting in 300 ADFs (10 to 150 arguments for ABA2AF; 10 to 300 arguments else). Query arguments were selected by sampling uniformly at random from the set of all arguments.

For the experiments we used 2.83GHz Intel Xeon E5440 quad-core machines with 32GB RAM running Debian GNU/Linux, enforcing a 1800-sec. timeout per instance. We compared the performance of our system with goDiamond 0.6.6 [Strass and Ellmauthaler, 2017] with clingo 5.2.1 [Gebser *et al.*, 2016], QADF 2.9.3 [Diller *et al.*, 2015] with bloqqer 037 [Biere *et al.*, 2011] and DepQBF 6.03 [Lonsing and Egly, 2017], and YADF 0.1.0 [Brewka *et al.*, 2017a] with lpopt 2.2 [Bichler *et al.*, 2016] and clingo 5.2.1.

We considered *Cred*$_{adm}$ and *Skept*$_{prf}$. Table 3 presents an overview of the mean running times and the number of timeouts, for each domain and solver considered. In the case of *Cred*$_{adm}$, ADM-K-BIP (a single SAT solver call) has the lowest mean running times (see also Figure 2 (center)). However, for the ABA2AF instances the algorithm ADM-2 produces less timeouts, albeit being slower on average. For *Skept*$_{prf}$, the procedure PRF-K-BIP-OPT performs exceptionally well with the ABA2AF instances, also having the lowest mean running time for Planning. As can be seen also in Figure 2 (right), the experiments clearly demonstrate the viability and efficiency of our SAT-based approach, in particular when making use of k-bipolarity in PRF-K-BIP. The shortcuts used in PRF-K-BIP-OPT then give another minor performance improvement.

The case of the ADM-K-BIP encoding vs the ADM-2 procedure for credulous reasoning under *adm* is shown in Figure 2 (left). The second level procedure performs better for some ABA2AF instances, even solving some of those for which the k-bipolar encoding produces a memory-out. We noticed that this happens when the parameter k exceeds a value of 15, and hypothesize this to be due to the size of the encoding.

7 Conclusion

In this work, we studied the complexity of subclasses of ADFs and incorporated these insights into a novel and competitive system based on incremental SAT solving. For future work, we plan to include further subclasses (see, e.g. [Polberg, 2016]) and tune the system by exploiting acyclicity.

Acknowledgments

We thank Matti Järvisalo for discussions on the CNF translation. This work has been supported by the Austrian Science Fund (FWF): P30168-N31, I2854, and Y698, the Academy of Finland grant 312662, and the Doctoral Programme in Computer Science (DoCS) at the University of Helsinki.

References

[Al-Abdulkarim *et al.*, 2016] L. Al-Abdulkarim, K. Atkinson, and T. J. M. Bench-Capon. A methodology for designing systems to reason with legal cases using abstract dialectical frameworks. *Artif. Intell. Law*, 24(1):1–49, 2016.

[Atkinson *et al.*, 2017] K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, H. Prakken, C. Reed, G. R. Simari, M. Thimm, and S. Villata. Towards artificial argumentation. *AI Magazine*, 38(3):25–36, 2017.

[Bench-Capon and Dunne, 2007] T. J. M. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.

[Bichler *et al.*, 2016] M. Bichler, M. Morak, and S. Woltran. lpopt: A rule optimization tool for answer set programming. In *Proc. LOPSTR*, pages 114–130, 2016.

[Biere *et al.*, 2011] A. Biere, F. Lonsing, and M. Seidl. Blocked clause elimination for QBF. In *Proc. CADE*, pages 101–115, 2011.

[Brewka and Woltran, 2010] G. Brewka and S. Woltran. Abstract dialectical frameworks. In *Proc. KR*, pages 102–111, 2010.

[Brewka *et al.*, 2017a] G. Brewka, M. Diller, G. Heissenberger, T. Linsbichler, and S. Woltran. Solving advanced argumentation problems with answer-set programming. In *Proc. AAAI*, pages 1077–1083. 2017.

[Brewka *et al.*, 2017b] G. Brewka, S. Ellmauthaler, H. Strass, J. P. Wallner, and S. Woltran. Abstract dialectical frameworks. An overview. *IfCoLog Journal of Logics and their Applications*, 4(8):2263–2318, 2017.

[Cabrio and Villata, 2016] E. Cabrio and S. Villata. Abstract dialectical frameworks for text exploration. In *Proc. ICAART*, pages 85–95, 2016.

[Cayrol and Lagasquie-Schiex, 2013] C. Cayrol and M.-C. Lagasquie-Schiex. Bipolarity in argumentation graphs: Towards a better understanding. *Int. J. Approx. Reasoning*, 54(7):876–899, 2013.

[Cerutti *et al.*, 2014] F. Cerutti, M. Giacomin, and M. Vallati. ArgSemSAT: Solving argumentation problems using SAT. In *Proc. COMMA*, pages 455–456, 2014.

[Charwat *et al.*, 2015] G. Charwat, W. Dvořák, S. A. Gaggl, J. P. Wallner, and S. Woltran. Methods for solving reasoning problems in abstract argumentation – A survey. *Artif. Intell.*, 220:28–63, 2015.

[Diller *et al.*, 2015] M. Diller, J. P. Wallner, and S. Woltran. Reasoning in abstract dialectical frameworks using quantified Boolean formulas. *Argument & Computation*, 6(2):149–177, 2015.

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

[Dunne, 2009] P. E. Dunne. The computational complexity of ideal semantics. *Artif. Intell.*, 173(18):1559–1591, 2009.

[Dvořák *et al.*, 2012] W. Dvořák, S. Ordyniak, and S. Szeider. Augmenting tractable fragments of abstract argumentation. *Artif.Intell.*, 186:157–173, 2012.

[Dvořák *et al.*, 2014] W. Dvořák, M. Järvisalo, J. P. Wallner, and S. Woltran. Complexity-sensitive decision procedures for abstract argumentation. *Artif. Intell.*, 206:53–78, 2014.

[Eén and Sörensson, 2003] N. Eén and N. Sörensson. An extensible SAT-solver. In *Proc. SAT*, pages 502–518, 2003.

[Gebser *et al.*, 2016] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and Philipp Wanko. Theory solving made easy with clingo 5. In *Techn. Commun. of ICLP*, pages 2:1–2:15, 2016.

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

[Lonsing and Egly, 2017] F. Lonsing and U. Egly. DepQBF 6.0: A search-based QBF solver beyond traditional QCDCL. In *Proc. CADE*, pages 371–384, 2017.

[Neugebauer, 2017] D. Neugebauer. Generating defeasible knowledge bases from real-world argumentations using D-BAS. In *Proc. AI^3@AI*IA*, pages 105–110, 2017.

[Nielsen and Parsons, 2007] S. Nielsen and S. Parsons. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In *Proc. ArgMAS*, pages 54–73. 2007.

[Polberg, 2016] S. Polberg. Understanding the abstract dialectical framework. In *Proc. JELIA*, pages 430–446, 2016.

[Strass and Ellmauthaler, 2017] H. Strass and S. Ellmauthaler. goDiamond 0.6.6. `argumentationcompetition.org/2017/goDIAMOND.pdf`, 2017.

[Strass and Wallner, 2015] H. Strass and J. P. Wallner. Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. *Artif. Intell.*, 226:34–74, 2015.

[Strass, 2014] H. Strass. Implementing instantiation of knowledge bases in argumentation frameworks. In *Proc. COMMA*, pages 475–476, 2014.

[Thimm and Villata, 2017] M. Thimm and S. Villata. The first international competition on computational models of argumentation: Results and analysis. *Artif. Intell.*, 252:267–294, 2017.

[Valiant and Vazirani, 1986] L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theor. Comput. Sci.*, 47(3):85–93, 1986.