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

Thomas Linsbichler¹, Marco Maratea², Andreas Niskanen³, Johannes P. Wallner¹, Stefan Woltran¹
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 Lagasque-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 \( \Pi^P_3 \)-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 ADFs 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 $I : A \to \{t, f, u\}$. That is, an interpretation maps each argument to either true ($t$), false ($f$), or undefined ($u$). An interpretation $I$ is two-valued if $I(a) \in \{t, f\}$ for all $a \in A$, and trivial (denoted $I_u$) if $I(a) = u$ for all $a \in A$. For a two-valued interpretation $I$, $I(\varphi)$ extends to the evaluation of a formula $\varphi$ under $I$ as usual. $I$ is equally or more informative than $J$, denoted by $J \leq I$, if $J(a) \in \{t, f\}$ implies $I(a) = I(a)$ for all $a \in A$. We denote by $<_i$ the strict version of $\leq_i$, i.e., $J <_i I$ if $J \leq I$ and $\exists a \in A$ s.t. $J(a) = u$ and $I(a) \in \{t, f\}$.

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: $\varphi(D)(a) = \{b \in A \mid (b, a) \in L\}$.

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

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

where $\varphi[I]$ is the formula obtained from $\varphi$ with each argument that $I$ assigns to either true or false being replaced by the corresponding truth constant, i.e., $\varphi[I] = \varphi[x \to \top \mid I(x) = t][x \to \bot \mid I(x) = f]$; 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 $I \leq \Gamma_D(I)$;
- complete in $D$ iff $I = \Gamma_D(I)$;
- grounded in $D$ iff $I$ is the least fixed-point of $\Gamma_D$;
- preferred in $D$ iff $I$ is $<_i$ 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 $a \in A$, $I(a) = t$ implies $\varphi_a[I]$ is satisfiable, and $I(a) = f$ 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 $x \in \{t, f, u\}$ for argument $b$ by $I^x_b$, i.e., $I^x_b(b) = x$ and

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$Cred_{\sigma}$</th>
<th>$Skept_{\sigma}$</th>
<th>$Exists_{\sigma}$</th>
<th>$Ver_{\sigma}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>cf</td>
<td>NP-c</td>
<td>trivial</td>
<td>NP-c</td>
<td>NP-c</td>
</tr>
<tr>
<td>nai</td>
<td>NP-c</td>
<td>$\Sigma^P_2$</td>
<td>NP-c</td>
<td>DP-c</td>
</tr>
<tr>
<td>adm</td>
<td>$\Sigma^P_2$</td>
<td>trivial</td>
<td>$\Sigma^P_2$</td>
<td>coNP-c</td>
</tr>
<tr>
<td>grd</td>
<td>coNP-c</td>
<td>coNP-c</td>
<td>coNP-c</td>
<td>DP-c</td>
</tr>
<tr>
<td>com</td>
<td>$\Sigma^P_2$</td>
<td>coNP-c</td>
<td>$\Sigma^P_2$</td>
<td>DP-c</td>
</tr>
<tr>
<td>prf</td>
<td>$\Sigma^P_2$</td>
<td>$\Pi^P_3$</td>
<td>$\Sigma^P_2$</td>
<td>$\Pi^P_3$</td>
</tr>
</tbody>
</table>

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

$I^x_b(a) = I(a)$ 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 $I$, $I(\varphi_a) = t$ implies $I^x_b(\varphi_a) = t$; (ii) attacking (in $D$) if for every two-valued interpretation $I$, $I(\varphi_a) = f$ implies $I^x_b(\varphi_a) = f$. $D$ is called bipolar if each link $(b, a) \in L$ is attacking or supporting.

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 $\sigma$, given ADF $D = (A, L, C)$:

- $Cred_{\sigma}$: Given $a \in A$, is there an $I \in \sigma(D)$ s.t. $I(a) = t$?
- $Skept_{\sigma}$: Given $a \in A$, is $I(a) = t$ for all $I \in \sigma(D)$?
- $Exists_{\sigma}$: Is there an $I \in \sigma(D)$ with $I \neq I_u$?
- $Ver_{\sigma}$: 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 $I = \{a \to t, b \to t, c \to f\}$ and $I' = \{a \to f, b \to f, c \to t\}$ are preferred in $D$ and $I_u$ 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, $Cred_{\sigma}$, $Skept_{\sigma}$, $Ver_{\sigma}$ are in P for $\sigma \in \{adm, grd, com, prf\}$ and $\tau \in \{grd, com, prf\}$. 
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\(_{prf}\) 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 \( \varphi_x = T \) for \( x \in X \) and \( \varphi_y = \phi \). Then, \( \{x \rightarrow u | x \in X\} \cup \{y \rightarrow t\} \in cf(D) \) iff \( \phi \) is satisfiable (\( J(y) = f \) for \( J \in prf(D) \) if \( \phi \) evaluates to \( f \) when all \( X \) are assigned true).

**Proposition 3.6.** For acyclic ADFs, Cred\(_{nai}\) 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 \( \sigma \)-interpretation.

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

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

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

In \( \sigma \)-concise ADFs Cred\(_{cf}\) and Skept\(_{cf}\) coincide. Complexity under nai, except Skept\(_{nai}\), does not deviate from \( I_a \).

**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 \( \phi \) over \( X \), let \( D_\phi = (X \cup \{z\}, L, C) \) with \( \varphi_x = \neg z \) for \( x \in X \), and \( \varphi_z = \phi \lor \neg z \). For any \( I \in cf(D_\phi) \) we have \( I(x) = u \). We have \( I(z) = t \) for some \( I \in cf(D_\phi) \) iff \( \varphi_z[I] \) is satisfiable iff \( \phi \) is satisfiable. Finally, \( I(z) \neq f \) in every \( I \in cf(D_\phi) \), as \( \varphi_z[I] \) is then a tautology. Hence, \( cf(D_\phi) = \{I_a, \{z \rightarrow t\} \cup \{x \rightarrow u | x \in X\}\} \) if \( \phi \) is satisfiable and \( cf(D_\phi) = \{I_a\} \) 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-conceivable 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-conceivable ADFs, Cred\(_{prf}\), Skept\(_{prf}\), and Exists\(_{prf}\) are \( \Sigma^P_2 \)-complete under randomized reductions; Ver\(_{prf}\) is \( \Pi^P_2 \)-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 \( L^0_D \) 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

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>bipolar [Strass and Wallner, 2015]</th>
<th>acyclic</th>
<th>( \sigma )-concise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cred(<em>{cf}), Skept(</em>{cf}), Exists(<em>{cf}), Ver(</em>{cf})</td>
<td>Cred(<em>{cf}), Skept(</em>{cf}), Exists(<em>{cf}), Ver(</em>{cf})</td>
<td>Cred(<em>{cf}), Skept(</em>{cf}), Exists(<em>{cf}), Ver(</em>{cf})</td>
<td>Cred(<em>{cf}), Skept(</em>{cf}), Exists(<em>{cf}), Ver(</em>{cf})</td>
</tr>
<tr>
<td>cf</td>
<td>in ( P ) trivial in ( P ) in ( P )</td>
<td>NP-c trivial trivial NP-c</td>
<td>trivial trivial trivial trivial</td>
</tr>
<tr>
<td>nai</td>
<td>in ( P ) coNP-c in ( P ) in ( P )</td>
<td>NP-c coNP-h trivial DP-c</td>
<td>NP-c NP-c NP-c DP-c</td>
</tr>
<tr>
<td>adm</td>
<td>NP-c trivial NP-c in ( P ) in ( P )</td>
<td>in ( P ) trivial in ( P ) in ( P ) coNP-c</td>
<td>trivial trivial trivial trivial</td>
</tr>
<tr>
<td>grd</td>
<td>NP-c in ( P ) in ( P ) in ( P )</td>
<td>in ( P ) in ( P ) in ( P ) in ( P )</td>
<td>coNP-c coNP-c coNP-c DP-c</td>
</tr>
<tr>
<td>com</td>
<td>NP-c in ( P ) in ( P ) in ( P )</td>
<td>in ( P ) in ( P ) in ( P ) in ( P )</td>
<td>coNP-c coNP-c coNP-c DP-c</td>
</tr>
<tr>
<td>prf</td>
<td>NP-c ( \Pi^P_2 )-concise coNP-c in ( P ) in ( P ) in ( P )</td>
<td>in ( P ) in ( P ) in ( P ) in ( P )</td>
<td>( \Sigma^P_2 )-concise ( \Sigma^P_2 )-concise ( \Sigma^P_2 )-concise ( \Pi^P_2 )-concise</td>
</tr>
</tbody>
</table>

Table 2: Complexity of subclasses of ADFs. \( C^* \) means that hardness for \( C \) is under randomized reductions.
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 \( \mathbb{P} \).

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.

**Proposition 4.9.** Exists \( \nsigma \) and some integer \( k \) is a set \( \Sigma^P_2 \)-hardness under standard reductions, we can show it under randomized reductions giving rise to a slightly weaker result.

**Theorem 4.10.** For \( k \)-prf-concise ADFs, Skept\(_{prf}\) is in \( \Delta^P_2 \).

**Proof (Sketch).** Let \( \mathcal{I} = \emptyset \). Construct a preferred interpretation \( I \) of \( D = (A, L, C) \) by starting with \( I = I_0 \) and ask a \( \Sigma^P_2 \) oracle whether there is a \( J \in \text{adm}(D) \) with \( J > I \) and no \( J' \in \mathcal{I} \) with \( J' \leq I \). If yes, set \( I = J \) and repeat the oracle call. Otherwise, if \( I = I_0 \), 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. \( \square \)

**Corollary 4.11.** For \( k \)-nai-concise ADFs, Skept\(_{nai}\) is in \( \Delta^P_2 \).

For admissible and complete semantics, we consider it rather unlikely that Exists\(_{nai}\) or Cred\(_{nai}\) get easier for \( k \)-sigma-concise ADFs with \( k \geq 2 \). While we are not able to show \( \Sigma^P_2 \)-hardness under standard reductions, we can show it under randomized reductions giving rise to a slightly weaker result.

**Proposition 4.12.** Let \( \sigma \in \{ \text{adm}, \text{com} \} \). For \( k \)-sigma-concise ADFs it holds that Exists\(_{nai} \) and Cred\(_{nai} \) are \( \Sigma^P_2 \)-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\(_{pbf}\). 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 \( k \)-bipolar and \( k \)-prf-concise ADFs it holds that Skept\(_{prf}\) is in \( \Delta^P_2 \).

## 5 Algorithms

We now present novel algorithms for ADF reasoning tasks based on results from Sections 3 and 4. We focus on Skept\(_{pbf}\) 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\(_{pbf}^\sigma\), which asks for a \( \sigma \)-interpretation strictly more informative than a given interpretation.

**Proposition 5.1.** Given \( \sigma \in \{ \text{cf}, \text{nai}, \text{grd}, \text{adm}, \text{com}, \text{prf} \} \), complexity of Exists\(_{pbf}^\sigma\) and Exists\(_{nai} \) 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 \( U = \{ s^t, s^t \mid s \in A \} \cup \{ p^s_r \mid (r, s) \in L \} \). Moreover, \( \varphi^s_r \) denotes the acceptance condition \( \varphi_s \) where each occurrence of \( r \in \text{par}_D(s) \) is replaced by \( p^s_r \). The encoding for Exists\(_{adm}\) is then given by \( \text{Exists}^D_{adm} = \varphi^s_{\text{com}} \land \bigwedge_{s \in A} (\phi^s_{\text{nai}} \land \phi^s_{\text{prf}}) \). It is satisfiable iff there is a \( J \in \text{eff}(D) \) with \( I <_i J \). Intuitively, atoms \( s^t \) and \( s^t \) represent the guess of an interpretation \( J \), indicating that \( J(s) = t \)
Algorithm 1 SKEW_{\text{prf}}(D, \alpha)

1: \rho \leftarrow \emptyset
2: while I \leftarrow \text{Exists}_{\text{adm}}(D, I_{\alpha}, \rho) do
3: \quad while I' \leftarrow \text{Exists}_{\text{adm}}(D, I, \rho) do I' \leftarrow I'
4: \quad if I(\alpha) \neq t then reject else \rho \leftarrow \rho \land \text{Refine}_{\text{prf}}(I)
5: \quad if \rho = \emptyset then reject else accept

\[ \land \{ I(s) = t \land \bigwedge \{ f \mid I(s) = f \} \} = \bigwedge \{ f \mid I(s) = f \} \]
\[ \land (s_{T} \lor \neg s_{T}) \land (s_{T} \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}

\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \]

Algorithms. Algorithm 1 solves SKEW_{\text{prf}} which is \Pi^p_3-c complexity by making the implementation of the membership proof of Theorem 4.10. It iteratively calls Exists_{\text{adm}} until a preferred interpretation is reached. Thereby, preferred interpretations already visited are ruled out by maintaining the formula \rho. This formula is refined in line 4 by Refine_{\text{prf}}(I) = \bigwedge \{ f \mid I(s) = f \} \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \}
\[ \land (s_{T} \lor \neg s_{T}) \land \bigwedge \{ f \mid I(s) = f \} \]

Solvers. We denote Algorithm 1 by PRF-3, which uses Algorithm 2 for the Exists_{\text{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_{\text{adm,bip}} decides Exists_{\text{adm}}, giving rise to PRF-BIP. Otherwise, the polarity of each link can be determined by two SAT calls as a preprocessing step, enabling Exists_{\text{adm}} to be decidable via a SAT
call using $\Phi^{D,I}_{adm,bip}$. 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 $s^\perp$ to $\Phi^{D,I}_{adm,bip}$ in Algorithm 2 with parameters $I = I_n$ and $\rho = \emptyset$. In the $k$-bipolar case ADM-K-BIP decides the problem by a single SAT call using $\Phi^{D,I}_{adm,bip} \land s^\perp$.

### 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](http://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 [Één 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/](http://www.dbai.tuwien.ac.at/proj/adf/yadf/)). These ADFs were generated from the AFs of domains ABA2AF, Planning, and Traffic from IICMA 2017 ([argumentationcompetition.org/2017/](http://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 subformulas are connected via $\land$ or $\lor$ 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 2.8GHz 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 blocker 037 [Biere et al., 2011] and DepQBF 6.03 [Lonsing and Eigl, 2017], and YADF 0.1.0 [Brewka et al., 2017a] with lpspe 2.2 [Bichler et al., 2016] and clingo 5.2.1.

We considered $Cred_{adm}$ and $Skept_{bip}$. 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 $Skept_{bip}$, 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. [Polver, 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


