Complexity-Sensitive Decision Procedures for Abstract Argumentation

Wolfgang Dvořák[a], Matti Järvisalo[b], Johannes Peter Wallner[c], Stefan Woltran[c]

[a] *University of Vienna, Faculty of Computer Science, Austria*
[b] *University of Helsinki, Helsinki Institute for Information Technology HIIT & Department of Computer Science, Finland*
[c] *Vienna University of Technology, Institute of Information Systems, Austria*

Abstract

Abstract argumentation frameworks (AFs) provide the basis for various reasoning problems in the area of Artificial Intelligence. Efficient evaluation of AFs has thus been identified as an important research challenge. So far, implemented systems for evaluating AFs have either followed a straight-forward reduction-based approach or been limited to certain tractable classes of AFs. In this work, we present a generic approach for reasoning over AFs, based on the novel concept of complexity-sensitivity. Establishing the theoretical foundations of this approach, we derive several new complexity results for preferred, semi-stable and stage semantics which complement the current complexity landscape for abstract argumentation, providing further understanding on the sources of intractability of AF reasoning problems. The introduced generic framework exploits decision procedures for problems of lower complexity whenever possible. This allows, in particular, instantiations of the generic framework via harnessing in an iterative way current sophisticated Boolean satisfiability (SAT) solver technology for solving the considered AF reasoning problems. First experimental results show that the SAT-based instantiation of our novel approach outperforms existing systems.

Keywords:
abstract argumentation, computational complexity, argumentation procedures

1. Introduction

Formal argumentation has evolved into an important field in Artificial Intelligence. Abstract argumentation frameworks (AFs for short), as introduced by Dung [1], are central in formal argumentation, providing a simple yet powerful formalism to reason about conflicts between arguments. The power of the formalism, however, comes at a price. In particular, many important reasoning problems for AFs are located on the second level of the polynomial hierarchy, including skeptical reasoning in the preferred semantics [2], and both skeptical and credulous reasoning in the semi-stable and the stage semantics [3]. This naturally raises the question about the origin of this high complexity and, in particular, calls for research on lower complexity fragments of the reasoning tasks. The focus of this article is both on the identification of such lower-complexity fragments of second-level reasoning problems arising from abstract argumentation, and on exploiting this knowledge in developing efficient *complexity-sensitive* decision procedures for the generic second-level problems.

Tractable (i.e., polynomial-time decidable) fragments have been quite thoroughly studied in the literature (see [4, 5, 6, 7, 8] for instance). However, there is only little work on identifying fragments which are located on the first level (NP/coNP layer), that is, *in-between* tractability and full second-level complexity.

Identification of first-level fragments of second-level reasoning tasks is important due to several reasons. Firstly, from a theoretical point of view, such fragments show particular (but not all) sources of complexity of the considered problems, and pave the way towards "trichotomy"-style results (see [9] for an example in the context of answer-set programming). Secondly, NP fragments can be efficiently reduced to the problem of satisfiability in classical propositional logic (SAT). This allows for realizations of argumentation procedures by employing sophisticated SAT-solver technology [10, 11] for reasoning over argumentation frameworks.

Going even further, in this work we aim at designing decision procedures for second-level argumentation problems by exploiting fragments extending such first-level fragments. To this end, we use the NP decision procedures as NP oracles in an iterative fashion. Such approaches fall under the general *counter-example guided abstraction refinement* (CEGAR) approach originating from the field of model checking [12, 13]. For problems complete for the second level of the polynomial hierarchy, this leads to general procedures which, in the worst case, requires an exponential number of calls to the NP oracle, which is indeed unavoidable under the assumption that the polynomial hierarchy does not collapse. Nevertheless, we show in this work that such procedures can be designed to behave adequately on input instances that fall into the considered NP fragment and on instances for which a relatively low number of oracle calls is sufficient; as a generic notion, we say that such a procedure is *complexity-sensitive* w.r.t. the NP fragment at hand. For instance, for the second-level problem of answer-set existence for disjunctive logic programs [14], the successful loop-formula approach (see, e.g. [15]) yields a polynomial reduction to SAT for the fragment of tight programs, although in general the resulting SAT instance is of exponential size. This approach gives thus a practical decision procedure for (the second-level problems of) answer-set programming that is at the same time complexity-sensitive w.r.t. the NP fragment of tight disjunctive programs.

In this work we identify various lower-complexity fragments of second-level reasoning problems arising from abstract argumentation, and show how some of the fragments can be exploited in complexity-sensitive CEGAR-style decision procedures for the generic second-level problems. The fragments identified and exploited are based on notions of "*distance*" to particular NP fragments. This leads to the intuition that, the higher the distance, the more iterative calls to the NP oracle are needed. We also employ the concept of distance to *generalize* known classes of NP fragments.

In more detail, we focus on three important semantics for abstract argumentation: the preferred, semi-stable and stage semantics. Our complexity analysis is based on six different classes of argumentation frameworks which are known to yield milder complexity results for at least one of these semantics. Firstly, we present complexity results for these classes in cases where the exact complexity for a particular semantics has not been established yet. Moreover, we categorize the classes into *syntactical* and *semantical* families. For the former family, we consider the known concepts of acyclic and odd-cycle free AFs, as well as a new class (so-called weak cyclic AFs).

As semantical subclasses we consider the prominent class of coherent AFs [2]; the class of AFs which possess at least one stable extension (stable-consistent AFs); and the class of AFs which possess a unique preferred extension.

Secondly, we consider alternative notions of distance in order to capture AFs which are "close" to one of the aforementioned classes. We consider the following realizations of distance:

(i) *graph-based distance measures*, where the parameter is given by the number of arguments to be deleted from a given AF in order to fall into a specified class; and

(ii) *extension-based distance measures*, which apply to the semantical subclasses.

For instance (among others), starting from the class of coherent AFs (where the preferred and stable extensions coincide), we consider as parameter the number of additional preferred extensions.

The main contributions of this article are the following.

- We show new complexity results for acceptance problems in argumentation on certain fragments. In particular, for the class of frameworks which possess a unique preferred extension, semi-stable semantics yields milder complexity than stage semantics. To the best of our knowledge, this is the first result that indicates a difference between the complexities of these two semantics.

- We show that graph-based distance measures are in most cases tight: already a small distance from the subclass at hand leads to the full second-level complexity. This reveals that syntactic fragments based on such distance measures do not hint towards complexity-sensitive decision procedures.

- Towards the design of complexity-sensitive decision procedures, we also identify extension-based distance measures and show that certain problems can be solved by a bounded number (in terms of the distance) of NP-oracle calls.

- Exploiting the suitable extension-based distance measures, we develop a generic framework of complexity-sensitive decision procedures for the different second-level reasoning problems within abstract argumentation. We present our procedures in terms of (first-level) argumentation problems, i.e., we give novel characterizations of preferred, semi-stable, and stage semantics in terms of simpler semantics (such as stable and complete). The actual computation of the simpler semantics can be instantiated in various ways.

- We show in detail how the generic framework can be instantiated using a SAT-based CEGAR-style approach. For this, we develop novel SAT-encodings for the oracle calls, differing from previously suggested SAT-encodings of first-level AF reasoning problems [16]. Notably, we exploit possibilities of learning from counter-examples both on the level of the original argumentation framework as well as the SAT oracle during computation. Importantly, while monolithic SAT-encodings of second-level argumentation problems are deemed to be of exponential size, our procedures are truly complexity-sensitive in that the exponential space requirements may be circumvented in cases where it suffices to consider a small part of a monolithic encoding to decide the actual query.

- We have implemented a prototype of the SAT-based instantiation of our approach, exploiting a state-of-the-art *conflict-driven clause learning* (CDCL) SAT-solver as the underlying NP-oracle. Experiments show the high potential of the proposed approach compared to other state-of-the-art implementations for abstract argumentation, in particular the logic-programming approach based on monolithic encodings of second-level problems [17, 18] arising in argumentation.

The rest of this article is organized as follows. After preliminaries on argumentation frameworks, their semantics, and related decision problems (Section 2), we review and prove new complexity results for reasoning with preferred, stage, and semi-stable semantics over several classes of AFs (Section 3). We then introduce extensions of these classes into syntactic classes of AFs based on graph-based distance measures, and show that these syntactic classes preserve the full second-level complexity under preferred, stage, and semi-stable semantics (Section 4). We then proceed by considering so-called extension-based distance measures which take the number of extensions into account, giving rise to so-called semantical subclasses of AFs, and provide positive upper bounds on the complexity of these subclasses (Section 5). These results give rise to novel complexity-sensitive decision procedures for skeptical and credulous acceptance (Section 6). After introducing the generic procedures, we show in detail how the procedures can be naturally instantiated using a SAT-solver as the NP-oracle within the procedures (Section 7). Before conclusions, we provide an experimental evaluation of the SAT-based instantiations of the procedures (Section 8).

2. Preliminaries

In this section we review (abstract) argumentation frameworks [1], the semantics studied in this paper (see also [19, 20]), and known complexity results for decision problems under the different semantics.

Definition 1. *An* argumentation framework (AF) *is a pair $F = (A, R)$ where A is a finite set of arguments and $R \subseteq A \times A$ is the* attack relation. *For a given AF $F = (A, R)$ we use A_F to denote the set A of its arguments and R_F to denote its attack relation R. We sometimes use the notation $a \rightarrowtail^R b$ instead of $(a, b) \in R$. For $S \subseteq A$ and $a \in A$, we write $S \rightarrowtail^R a$ (resp. $a \rightarrowtail^R S$) in case there is a $b \in S$*

such that $b \rightarrowtail^R a$ (resp. $a \rightarrowtail^R b$). In case no ambiguity arises, we use $\rightarrowtail$ instead of $\rightarrowtail^R$.

Semantics for argumentation frameworks assign to each AF $F = (A, R)$ a set $\sigma(F) \subseteq 2^A$ of extensions. We consider here for σ the functions stb, adm, prf, com, stg, and sem which stand for stable, admissible, preferred, complete, stage, and respectively, semi-stable semantics. Before giving the actual definitions for these semantics, we need to define a few more formal concepts.

Definition 2. *Given an AF $F = (A, R)$, an argument $a \in A$ is* defended *(in F) by a set $S \subseteq A$ if for each $b \in A$ such that $b \rightarrowtail a$, also $S \rightarrowtail b$ holds. Moreover, for $S \subseteq A$, we define the* range *of S, denoted by S^+_R, as the set $S \cup \{b \mid S \rightarrowtail b\}$.*

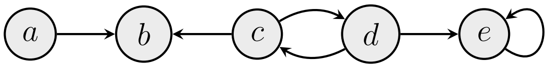
We are now ready to define the concrete semantics for AFs.

Definition 3. *Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is* conflict-free (in F), *denoted $S \in cf(F)$, iff there are no $a, b \in S$ such that $(a, b) \in R$. For $S \in cf(F)$, it holds that*

- *$S \in stb(F)$ if for each $a \in A \setminus S$, $S \rightarrowtail a$, i.e. $S^+_R = A$;*
- *$S \in adm(F)$ if each $a \in S$ is defended by S;*
- *$S \in prf(F)$ if $S \in adm(F)$ and there is no $T \in adm(F)$ with $T \supset S$;*
- *$S \in com(F)$ if $S \in adm(F)$ and for each $a \in A$ defended by S, $a \in S$ holds;*
- *$S \in stg(F)$ if there is no $T \in cf(F)$ with $T^+_R \supset S^+_R$;*
- *$S \in sem(F)$, if $S \in adm(F)$ and there is no $T \in adm(F)$ with $T^+_R \supset S^+_R$.*

We recall that for each AF F, $stb(F) \subseteq sem(F) \subseteq prf(F) \subseteq com(F) \subseteq adm(F)$ holds, and that for each of the considered semantics σ (except stable) $\sigma(F) \neq \emptyset$ holds. Moreover, in case an AF has at least one stable extension, its stable, semi-stable, and respectively, stage extensions coincide.

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



Here $stb(F) = stg(F) = sem(F) = \{\{a, d\}\}$. The admissible sets of F are $\emptyset$, $\{a\}$, $\{c\}$, $\{d\}$, $\{a, c\}$ and $\{a, d\}$. The set of preferred extensions is $prf(F) = \{\{a, c\}$, $\{a, d\}\}$. The complete extensions are $\{a\}$, $\{a, c\}$ and $\{a, d\}$. ◇

We assume the reader has knowledge about standard complexity classes, i.e. P, NP and L (logarithmic space). Nevertheless we briefly recapitulate the concept of oracle machines and some related complexity classes. Thus let $\mathcal{C}$ denote some complexity class. By a $\mathcal{C}$-oracle machine we mean a (polynomial time) Turing machine which can access an oracle that decides a given (sub)-problem in $\mathcal{C}$ within one step. We denote such machines as $\mathrm{P}^{\mathcal{C}}$ if the underlying Turing machine is deterministic; and $\mathrm{NP}^{\mathcal{C}}$ if the underlying Turing machine is nondeterministic. Moreover we consider oracle machines where the number of queries to the oracle is bounded by $f(k)$. We denote the corresponding complexity classes as $\mathrm{P}^{\mathcal{C}[f(k)]}$ for deterministic machines.

In the following we consider complexity classes using NP-oracles. First, the class $\Sigma_2^P = \mathrm{NP}^{\mathrm{NP}}$ denotes the set of problems which can be decided by a nondeterministic polynomial time algorithm that has (unrestricted) access to an NP-oracle. The class $\Pi_2^P = \text{coNP}^{\mathrm{NP}}$ is defined as the complementary class of Σ_2^P, i.e. $\Pi_2^P = \text{co}\Sigma_2^P$.

We now recall the complexity of reasoning in AFs for the following decision problems under different semantics σ.

- *Credulous Acceptance* (Cred_σ):
  Given an AF F and argument $a \in A_F$, is a contained in some $S \in \sigma(F)$?
- *Skeptical Acceptance* (Skept_σ):
  Given an AF F and argument $a \in A_F$, is a contained in each $S \in \sigma(F)$?
- *Verification* (Ver_σ): Given an AF F and a set $S \subseteq A_F$, is $S \in \sigma(F)$?
- *Existence* (Exists_σ): Given an AF F, is $\sigma(F) \neq \emptyset$?
- *Non-emptiness* ($\mathsf{Exists}_\sigma^{\neg\emptyset}$):
  Given an AF F, is there a set $S \subseteq A_F, S \neq \emptyset$ such that $S \in \sigma(F)$?

In accordance with the above problems we say that an argument is credulously (resp. skeptically) accepted iff it is contained in at least one extension (resp. in

Table 1: Complexity of decision problems for AFs.

σ	Cred_σ	Skept_σ	Ver_σ	Exists_σ	$\mathsf{Exists}_\sigma^{\neg\emptyset}$
stb	NP-c	coNP-c	in L	NP-c	NP-c
adm	NP-c	trivial	in L	trivial	NP-c
com	NP-c	P-c	in L	trivial	NP-c
prf	NP-c	Π_2^P-c	coNP-c	trivial	NP-c
sem	Σ_2^P-c	Π_2^P-c	coNP-c	trivial	NP-c
stg	Σ_2^P-c	Π_2^P-c	coNP-c	trivial	in L

all extensions). Table 1 summarizes the computational complexity of these problems [4, 21, 1, 2, 22, 3, 23], where an entry $\mathcal{C}$-c denotes that the corresponding problem is complete for class $\mathcal{C}$.

We will focus on Skept_{prf}, Cred_{sem}, Skept_{sem}, Cred_{stg} and Skept_{stg}. As Table 1 indicates, these problems are the ones on the second level of the polynomial hierarchy.

3. Subclasses of Argumentation Frameworks

In this section we review several classes of AFs where reasoning with preferred, stage or semi-stable semantics becomes easier compared to the results for the general case. Both earlier and new results are discussed. First, we consider the classes of acyclic and weakly cyclic AFs.

Definition 4. *An AF F is* acyclic *if there is no directed cycle of attacks in F; F is* weakly cyclic *if F can be made acyclic by deleting one argument (and its incident attacks) from each strongly connected component* [1] *(SCC) of F. We denote these classes of AFs by* acyc *and* wcyc.

One can easily show that deciding whether a given AF falls into one of these classes can be done efficiently. First, it is well known that deciding whether a

[1] A set of arguments is called *strongly connected* in an AF if there is a path from each argument to every other argument in the set. A *strongly connected component* of an AF is a $\subseteq$-maximal strongly connected set.

directed graph is cyclic or not can be done in polynomial time. Moreover, the SCCs of a graph can be computed in polynomial time. Thus, for testing whether a graph is in wcyc one first computes the SCCs and then looks in each SCC for an argument which, if removed, turns the SCC into an acyclic graph.

It is well known that the problems we are interested here become tractable when restricted to acyclic AFs. For the class wcyc (these are the AFs where the graph parameter cycle-rank is at most 1 [7]), we can make direct use of the following complexity result.

Proposition 1 ([7])**.** *For weakly cyclic AFs,* Skept_{prf} *is* coNP*-complete.*

The reasoning problems for stage and semi-stable semantics still maintain their full complexity when restricted to weak cyclic AFs [7].

We now turn to semantical subclasses.

Definition 5. *An AF F for which $stb(F) \neq \emptyset$ is* stable-consistent*. We denote the class of such AFs by* stablecons.

We recall that testing an AF for the existence of a stable extension is NP-complete [21] (cf. Table 1). The following result is immediate from the fact that, in case an AF has at least one stable extension, its stable, semi-stable, and respectively, stage extensions coincide.

Proposition 2. *For stable-consistent AFs,* Cred_σ *is* NP*-complete and* Skept_σ *is* coNP*-complete, $\sigma \in \{sem, stg\}$.*

However, in case of the preferred semantics, stable consistency is of no help for deciding skeptical acceptance.

Proposition 3. *For stable-consistent AFs, the problem* Skept_{prf} *is Π_2^P-complete.*

Proof. Hardness follows from a reduction in [2] that maps the Π_2^P-hard problem of deciding whether given a QBF $\Phi = \forall Y \exists Z \varphi(Y, Z)$, where φ is a CNF formula $\bigwedge_{c\in C} c$ with each clause $c \in C$ a disjunction of literals from $X = Y \cup Z$, is true to Skept_{prf}. We build $F_\Phi = (A_\Phi, R_\Phi)$ as follows (see also Figure 1):

$$\begin{aligned}
A_\Phi &= \{\varphi, b\} \cup C \cup X \cup \bar{X} \\
R_\Phi &= \{(c, \varphi) \mid c \in C\} \cup \{(x, \bar{x}), (\bar{x}, x) \mid x \in X\} \cup \\
&\quad \{(x, c) \mid x \text{ occurs in } c\} \cup \{(\bar{x}, c) \mid \neg x \text{ occurs in } c\} \cup \\
&\quad \{(\varphi, b), (b, b)\} \cup \{(b, z), (b, \bar{z}) \mid z \in Z\}
\end{aligned}$$

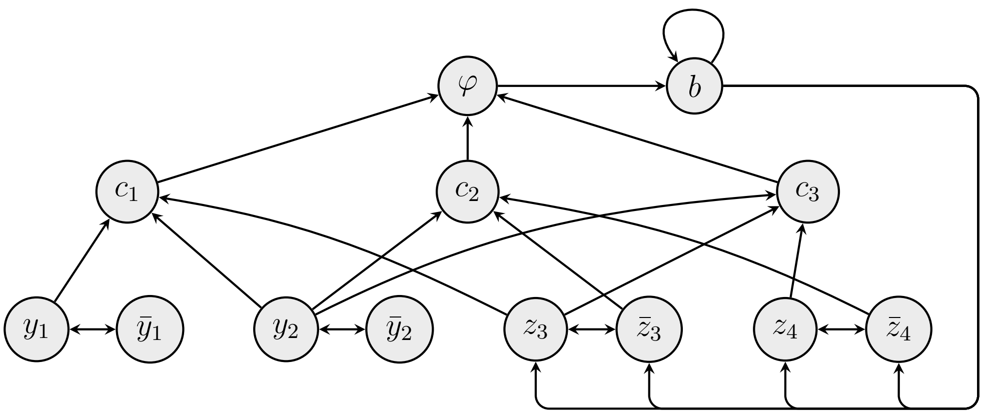


Figure 1: F_Φ for the QBF $\Phi = \forall y_1 y_2 \exists z_3 z_4 (y_1 \vee y_2 \vee z_3) \wedge (y_2 \vee \neg z_3 \vee \neg z_4) \wedge (y_2 \vee z_3 \vee z_4)$.

with $\bar{X} = \{\bar{x} \mid x \in X\}$ and $\bar{x}$ being a fresh argument.

We have that Φ is valid iff the argument φ is skeptically accepted in F_Φ [2]. Moreover, each model M of φ corresponds to the stable extension $M \cup \overline{X \setminus M} \cup \{\varphi\}$ of F_Φ. Since the QBF-problem remains hard for instances where the propositional formula φ has at least one model, the result follows. □

The next class relates preferred and stable extensions and was already introduced in Dungs seminal paper [1] and has been thoroughly discussed in [2].

Definition 6. *An AF F is* coherent *if $prf(F) = stb(F)$. We denote the class of such AFs by* coherent.

Proposition 4. *For coherent AFs and $\sigma \in \{prf, sem, stg\}$,* Cred_σ *is* NP*-complete and* Skept_σ *is* coNP*-complete.*

Proof. The result for $\sigma = prf$ is clear by definition when taking the complexity of stable semantics (recall Table 1) into account. Since each AF possesses at least one preferred extension, each coherent AF is also stable-consistent. The result for the remaining cases $\sigma \in \{sem, stg\}$ thus follows immediately as well. □

Testing coherence is in general even worse (Π_2^P-complete [2]) than testing stable-consistency. At first glance this restricts the practical value of this fragment, but there is a class of easy detectable coherent AFs, namely the AFs without odd-length cycles. Thus, we introduce one further syntactical subclass.

Definition 7. *An AF F is* odd-cycle free *if there is no directed cycle consisting of an odd number of attacks in F. We denote the class of odd-cycle free AFs by* ocf.

In fact, testing for odd-length cycles in digraphs can be done in polynomial time (see e.g. [24]).

The following proposition generalizes the observation of [5] that for $F \in \mathsf{ocf}$ reasoning with preferred semantics is complete for the first level of the polynomial hierarchy.

Proposition 5. *For AFs $F \in \mathsf{ocf}$, and $\sigma \in \{prf, sem, stg\}$, Cred_σ is NP-complete and Skept_σ is coNP-complete.*

Proof. Membership is immediate from the well-known result that every AF without odd-length cycles is coherent [1].

The hardness for the case of preferred semantics was already shown in [5]. For the sake of completeness we restate the proof which also applies to semi-stable and stage semantics. To this end we use what [25] calls the standard translation of a CNF formula to an AF given below, which originates from the work of [21]. Starting from a propositional formula $\varphi(X) = \bigwedge_{c \in C} c$ with each clause $c \in C$ a disjunction of literals from X, we build $F_\varphi = (A_\varphi, R_\varphi)$ as follows (see also Figure 2):

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

We have that φ is satisfiable iff the argument φ is credulously accepted iff the argument $\bar{\varphi}$ is not skeptically accepted [21]. As F_φ is free of odd-length cycles, the theorem follows. □

The final fragment we introduce is another semantical one. It makes use of the complexity gap between credulous and skeptical acceptance for preferred semantics.

Definition 8. *We denote the class of AFs F satisfying $|prf(F)| = 1$ by* $\mathsf{uniqpref}$.

We next show that testing whether there is at most one preferred extensions is coNP-complete, using a result from [26].

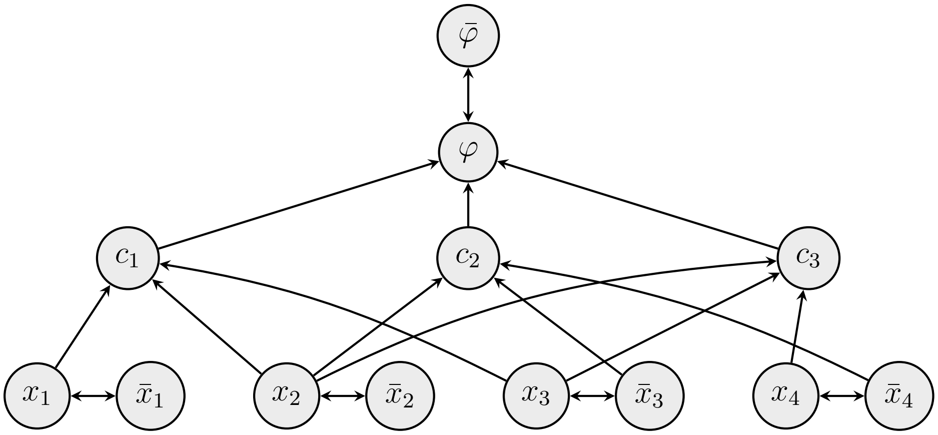


Figure 2: F_φ for the CNF-formula $\varphi = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee \neg x_4) \wedge (x_2 \vee x_3 \vee x_4)$.

Proposition 6. *Deciding whether a given AF F falls into the class* uniqpref *is* coNP-*complete.*

Proof. Membership is by a guess and check algorithm for falsifying that $F \in$ uniqpref. This algorithm guesses two sets $S, S' \subseteq A$ and tests whether $S \in adm(F)$, $S' \in adm(F)$ and $S \cup S' \notin adm(F)$.

The statement $S \in adm(F)$, $S' \in adm(F)$ and $S \cup S' \notin adm(F)$ is equivalent to the fact that $|prf(F)| \geq 2$. To show this let us first assume that we have S, S' being admissible and $S \cup S' \notin adm(F)$. This implies that $S \cup S' \notin cf(F)$. As by definition of preferred extensions, each admissible set is contained in a preferred extensions we can consider E, E' s.t. $S \subseteq E \in prf(F)$ and $S' \subseteq E' \in prf(F)$. As $S \cup S' \notin cf(F)$ also $E \cup E' \notin cf(F)$ and thus $E \neq E'$. Now assuming that we have $E, E' \in prf(F)$ with $E \neq E'$. By definition $E \in adm(F)$, $E' \in adm(F)$ and as they are maximal also $E \cup E' \notin adm(F)$.

We show the hardness part by the following reduction. Starting from a formula $\varphi(X) = \bigwedge_{c \in C} c$ with each clause $c \in C$ a disjunction of literals from X, we build $F_\varphi = (A_\varphi, R_\varphi)$ as follows (see also Figure 3)

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

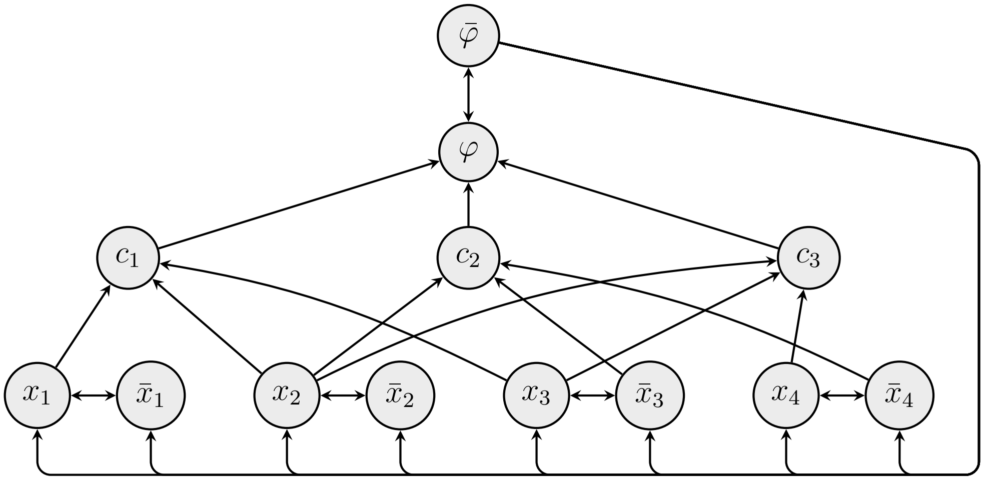


Figure 3: F_φ for the CNF-formula $\varphi = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee \neg x_4) \wedge (x_2 \vee x_3 \vee x_4)$.

We have that φ is satisfiable iff the set $\{\bar{\varphi}\} \cup C$ is not the only preferred extension [26]. □

Proposition 7. *For AFs* $F \in \mathsf{uniqpref}$ *and* $\sigma \in \{prf, sem\}$*, problems* Cred_σ *and* Skept_σ *are in* NP.

Proof. Credulous and skeptical acceptance coincide on AFs with a unique extension. Moreover, for each $F \in \mathsf{uniqpref}$, $sem(F) = prf(F)$ holds since the existence of a semi-stable extension is guaranteed for finite AFs, and each semi-stable extension is also preferred. □

It is open whether these problems are also NP-hard. However, we can show NP-hardness under so-called *randomized reductions* [27].

Proposition 8. *For AFs* $F \in \mathsf{uniqpref}$ *and* $\sigma \in \{prf, sem\}$*, problems* Cred_σ *and* Skept_σ *are* NP*-hard under randomized reductions.*

Proof. To show hardness we consider UNIQUE SAT, a version of the SAT problem where it is guaranteed that there is at most one model. This version of SAT is NP-hard under randomized reductions [27]. We use a variation of the reduction from the proof of Proposition 6 to reduce UNIQUE SAT to Cred_{prf}. More precisely, we extend the AF $F_\varphi = (A_\varphi, R_\varphi)$ by one self-attack $(\bar{\varphi}, \bar{\varphi})$ (see also Figure 4).

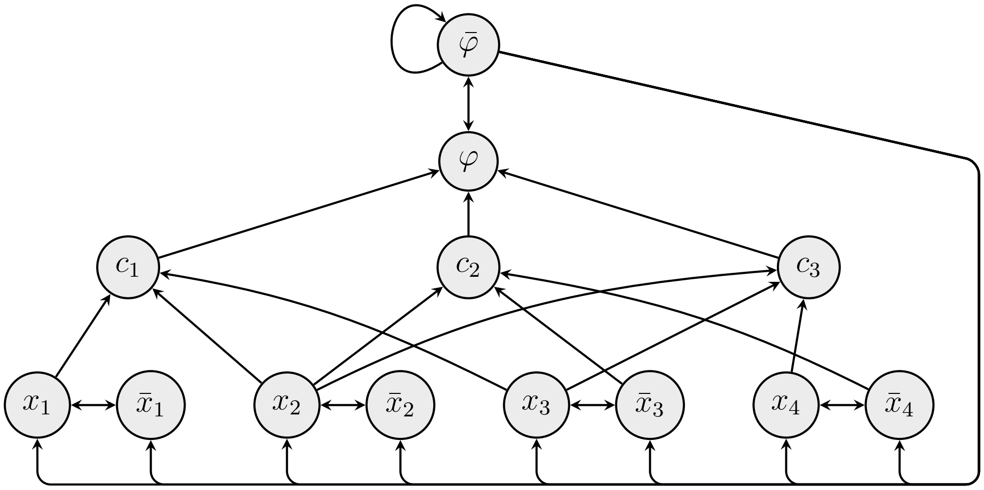


Figure 4: $(A_\varphi, R_\varphi \cup \{(\bar{\varphi}, \bar{\varphi})\})$ for the CNF-formula $\varphi = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee \neg x_4) \wedge (x_2 \vee x_3 \vee x_4)$.

Now the set $\{\bar{\varphi}\} \cup C$ is no longer admissible and $\bar{\varphi}$ cannot be contained in any preferred extension. One can show that each non-empty admissible set contains φ as follows. Indeed, consider a set E admissible for such a modified framework with $\varphi \notin E$. Then all $x, \bar{x}$ are not defended against $\bar{\varphi}$ and thus not in E. Moreover, also the arguments $c \in C$ are not defended by E as each of them is attacked by some $x, \bar{x}$, but none of the $x, \bar{x}$ are attacked by E. Hence also the $c \in C$ are not in E and consequently $E = \emptyset$ follows. Now, as each non-empty admissible set contains φ, such a set—by the standard argument—corresponds to a model of φ. Hence, taking into account that φ is guaranteed to have at most one model, we have at most one non-empty admissible set, and thus the AF has exactly one preferred extension. □

For the stage semantics, however, the complexity of acceptance problems remains as high as in the general case.

Proposition 9. *For AFs $F \in$ uniqpref, Cred$_{stg}$ is Σ_2^P-complete and Skept$_{stg}$ is Π_2^P-complete.*

Proof. Consider an arbitrary AF $F = (A, R)$ and let $t \notin A$ be a fresh argument. We construct $F' = (A \cup \{t\}, R \cup \{t, t\} \cup \{(t, a) \mid a \in A\})$. As the new argument t is in conflict with itself it cannot be contained in a conflict-free set. Furthermore,

Table 2: Complexity when the AF belongs to a sub-class $\mathcal{G}$.

$\mathcal{G}$	Skept_{prf}	Cred_{sem}	Skept_{sem}	Cred_{stg}	Skept_{stg}
acyc	P-c	P-c	P-c	P-c	P-c
wcyc	coNP-c	Σ_2^P-c	Π_2^P-c	Σ_2^P-c	Π_2^P-c
ocf	coNP-c	NP-c	coNP-c	NP-c	coNP-c
stablecons	Π_2^P-c	NP-c	coNP-c	NP-c	coNP-c
coherent	coNP-c	NP-c	coNP-c	NP-c	coNP-c
uniqpref	in NP	in NP	in NP	Σ_2^P-c	Π_2^P-c

as we do not add new conflicts between the original arguments, we have that $cf(F) = cf(F')$. Moreover as all new attacks origin from t also the ranges of the conflict-free sets are unchanged and thus also $stg(F) = stg(F')$. Finally, we have that $prf(F') = \{\emptyset\}$ as none of the arguments can be defended against t. Hence any decision problem for stg can be directly expressed in AFs from $\mathsf{uniqpref}$. □

To summarize, we have introduced several kinds of AF-subclasses. They can be grouped into syntactical (acyc, wcyc, ocf), and semantical classes (stablecons, coherent, uniqpref). The complexity results are summarized in Table 2. In the next two sections, we study possibilities of extending the "good" complexity behavior of these classes. To this end, we will introduce certain distance measures with the aim of maintaining lower complexity as long as the distance to such a class is bound.

4. Graph-Based Distance Measures

A natural way to generalize the introduced subclasses is to consider the minimal number of arguments one has to delete from an AF so that the modified AF falls into the respective class (see also [8]). This gives rise to the following distance measure.

Definition 9. *Let $\mathcal{G}$ be a graph class and $F = (A, R)$ an AF. We define $dist_{\mathcal{G}}(F)$ as the minimal number k such that there is a set $S \subseteq A$ with $|S| = k$ and $(A \setminus S, R \cap (A \setminus S \times A \setminus S)) \in \mathcal{G}$. If there is no such set S we define $dist_{\mathcal{G}}(F) = \infty$.*

We start with the class wcyc of weakly cyclic AFs. Recall that this class decreases complexity only in the case of skeptical acceptance with respect to preferred semantics.

Proposition 10. Skept_{prf} *is* Π_2^P*-hard for AFs* F *with* $dist_{\mathsf{wcyc}}(F) = 1$.

Proof. Π_2^P-hardness is established by re-using the reduction from the proof of Proposition 3 and observing that deletion of the argument b in F_Φ results in a framework with SCCs of size at most 2 which is easily seen to be weakly cyclic. □

In words, the subclass wcyc is tight w.r.t. the introduced distance in the sense that a single argument violating membership in wcyc is sufficient for the general Π_2^P-hardness. An analogous result can be shown for the class ocf.

Proposition 11. Skept_{prf} *is* Π_2^P*-hard, for AFs* F *with* $dist_{\mathsf{ocf}}(F) = 1$.

Proof. Similar as for Proposition 10, deleting argument b from the AF constructed in the proof of Proposition 3 yields an AF that is free of odd-length cycles. □

The same effect can be shown for semi-stable acceptance.

Proposition 12. Cred_{sem} *is* Σ_2^P*-hard and* Skept_{sem} *is* Π_2^P*-hard, for AFs* F *with* $dist_{\mathsf{ocf}}(F) = 1$.

Proof. For a QBF $\Phi = \forall Y \exists Z \varphi(Y,Z)$ with $\varphi = \bigwedge_{c \in C} c$ in CNF build the AF $F_\Phi^{sem} = (A_\Phi, R_\Phi)$ with $X = Y \cup Z$ as follows (see also Figure 5):

$$\begin{aligned}
A_\Phi &= \{\varphi, \bar{\varphi}, b, g\} \cup C \cup X \cup \bar{X} \cup Y' \cup \bar{Y}' \\
R_\Phi &= \{(c,\varphi) \mid c \in C\} \cup \{(x,\bar{x}), (\bar{x},x) \mid x \in X\} \cup \\
&\quad \{(x,c) \mid x \text{ occurs in } c\} \cup \{(\bar{x},c) \mid \neg x \text{ occurs in } c\} \cup \\
&\quad \{(y,y'), (\bar{y},\bar{y}') \mid y \in Y\} \cup \{(\varphi,b), (g,g), (g,b)\} \cup \\
&\quad \{(g,y'), (g,\bar{y}') \mid y \in Y\} \cup \{(\varphi,\bar{\varphi}), (\bar{\varphi},\varphi)\}.
\end{aligned}$$

This reduction is a variation of the reduction presented in [3] and can be easily shown to be equivalent.[2] Hence φ is skeptically accepted iff $\bar{\varphi}$ is not credulously

[2] The difference being that the original reduction does not contain the argument g, but self-attacks for each of the arguments $\{b\} \cup \{y', \bar{y}' \mid y \in Y\}$. In our reduction the argument g is not acceptable and therefore none of the arguments $\{b\} \cup \{y', \bar{y}' \mid y \in Y\}$ can be defended against g and thus are unacceptable. Hence the non-empty admissible sets and semi-stable extensions of the AFs constructed by the two different reductions coincide.

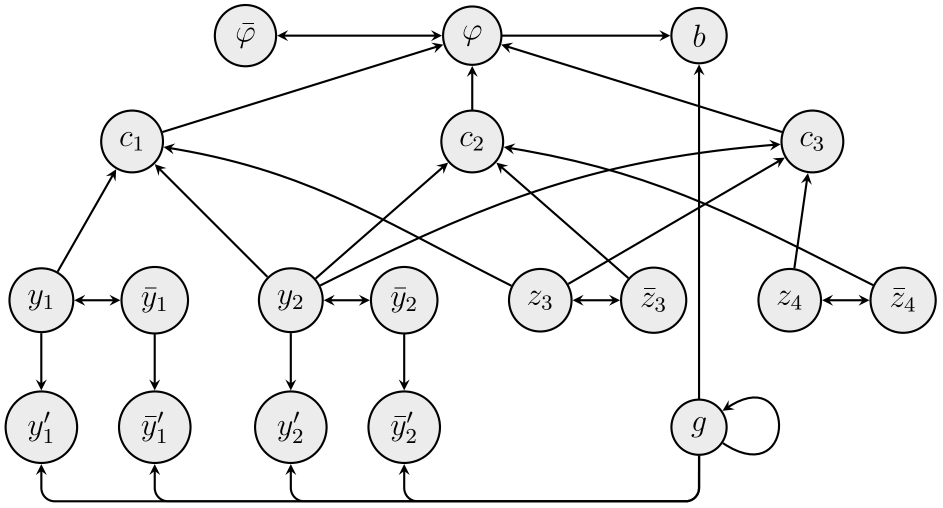


Figure 5: F_{Φ}^{sem} for the QBF $\Phi = \forall y_1 y_2 \exists z_3 z_4 (y_1 \vee y_2 \vee z_3) \wedge (y_2 \vee \neg z_3 \vee \neg z_4) \wedge (y_2 \vee z_3 \vee z_4)$.

accepted iff Φ is true. Finally, deleting the argument g results in an AF that is free of odd-length cycles. □

For the stage semantics, we can give an even stronger result, namely in terms of acyclic frameworks.

Proposition 13. *Cred_{stg} is Σ_2^P-hard and Skept_{stg} is Π_2^P-hard, for AFs F with $dist_{\mathsf{acyc}}(F) = 1$.*

Proof. Hardness results from a reduction from the MINSAT problem, i.e. deciding whether a variable z is in a $\subseteq$-minimal model of a propositional formula $\varphi = \bigwedge_{c \in C} c$ in CNF over atoms X. For the reduction we additionally assume an arbitrary order $<$ on the clauses of φ. We build the AF $F_{\varphi,z} = (A, R)$ where $z \in X$ as follows (see Figure 6 for illustration):

$$\begin{aligned}
A &= \{\varphi, b, q\} \cup C \cup C' \cup X \cup \bar{X} \\
R &= \{(c, \varphi) \mid c \in C\} \cup \{(t, b), (b, b), (q, z)\} \cup \{(\bar{x}, x) \mid x \in X\} \cup \\
&\quad \{(x, c) \mid x \text{ occurs in } c\} \cup \{(\bar{x}, c) \mid \neg x \text{ occurs in } c\} \cup \\
&\quad \{(c', a) \mid c \in C, a \in X \cup \bar{X} \cup (C \setminus \{c\}) \cup \{c' : c' < c\}\}.
\end{aligned}$$

It is easy to see that the only cycle in the constructed AF is by the self-attacking argument b. Thus, $dist_{\mathsf{acyc}}(F_{\varphi,z}) = 1$ as deleting the argument b would result in an acyclic AF. To prove the assertions we show that the following statements are equivalent:

1. z is in a minimal model of φ;
2. z is credulously accepted (w.r.t. stg) in $F_{\varphi,z}$;
3. q is not skeptically accepted (w.r.t. stg) in $F_{\varphi,z}$.

$(1) \Leftrightarrow (2)$ We recall that each stage extension is also a $\subseteq$-maximal conflict free set (mcf). Hence we consider only mcf sets as candidates for stage extensions. In the following we will use the relation $\leq^+_R$ defined as $S \leq^+_R T$ iff $S^+_R \subseteq T^+_R$.

First let us consider mcf sets of $F_{\varphi,z} = (A, R)$ containing an argument c'. For simplicity we enumerate the clauses $c_1, \dots, c_m$ and the arguments $c'_1, \dots, c'_m$, according to the order $<$ on the clauses. Now one can easily check that these mcf sets are given by $\{\{c'_i, \varphi, q\}, \{c'_i, c_i, q\} \mid 1 \leq i \leq m\}$. Further, we have that the arguments $c'_1, \dots, c'_m$ are in conflict with each other but not attacked from any other argument. Thus, when concerning the $\leq^+_R$-maximality (the maximality of the range) of the above mcf sets, they only compete with each other but not with any other mcf set. Comparing the range of these extensions we get that $\{c'_i, c_i, q\}^+ = A \setminus \{b, c'_1, \dots, c'_{i-1}\}$ and $\{c'_i, \varphi, q\}^+ = A \setminus \{c_i, c'_1, \dots, c'_{i-1}\}$. Hence the stage extensions among them are the following $\{\{c'_i, \varphi, q\} \mid 1 \leq i \leq n\} \cup \{\{c'_1, c_1, q\}\}$.

Now let us consider mcf sets E such that for each $1 \leq i \leq m$, $c'_i \notin E$. We already have stage extensions with ranges $A \setminus \{b\}$ and resp. $A \setminus \{c_i, c'_1, \dots, c'_{i-1}\}$. Hence, as $\{c'_1, \dots, c'_m\} \cap E^+ = \emptyset$, the only way for E being $\leq^+_R$-maximal is that $\{b, c_1, \dots, c_m\} \subseteq E^+$. When $b \in E^+$, we have that $\varphi \in E$ and hence for $1 \leq i \leq m$ that $c_i \notin E$. Hence, by the construction of $F_{\varphi,z}$, $\{b, c_1, \dots, c_m\} \subseteq E^+$ iff $\varphi \in E$ and $X \cap E$ is a model of φ. So there is a one-to-one correspondence between models M of φ and candidates for stage extensions $M^* := M \cup \overline{(X \setminus M)} \cup \{\varphi\} \cup \{q \mid \text{if } z \notin M\}$. By the construction the range of each candidate is clearly incomparable with the ranges of the already determined stage extensions $\{\{c_i, \varphi, q\} \mid 1 \leq i \leq n\} \cup \{\{E_1, c_1, q\}\}$; and thus the $\leq^+_R$-maximality of such a candidate only depends on the other candidates with $c'_i \notin E$ for $1 \leq i \leq m$. It remains to show that for two models M, N, we have that $M \subseteq N$ iff $M^* \geq^+_R N^*$.

For the "only if" direction consider $M \subseteq N$. Using that M, N are models of φ we obtain that $(M^*)^+ = A \setminus (\bar{M} \cup \{c'_1, \dots, c'_m\}) \cup \{q \mid \text{if } x \notin M\}$ and $(N^*)^+ = A \setminus (\bar{N} \cup \{c'_1, \dots, c'_m\}) \cup \{q \mid \text{if } x \notin N\}$. As by assumption $M \subseteq N$ we have that $A \setminus (\bar{M} \cup \{c'_1, \dots, c'_m\}) \supseteq A \setminus (\bar{N} \cup \{c'_1, \dots, c'_m\})$ and $\{q \mid \text{if } x \notin M\} \subseteq \{q \mid \text{if } x \notin N\}$. Hence $M^* \geq^+_R N^*$.

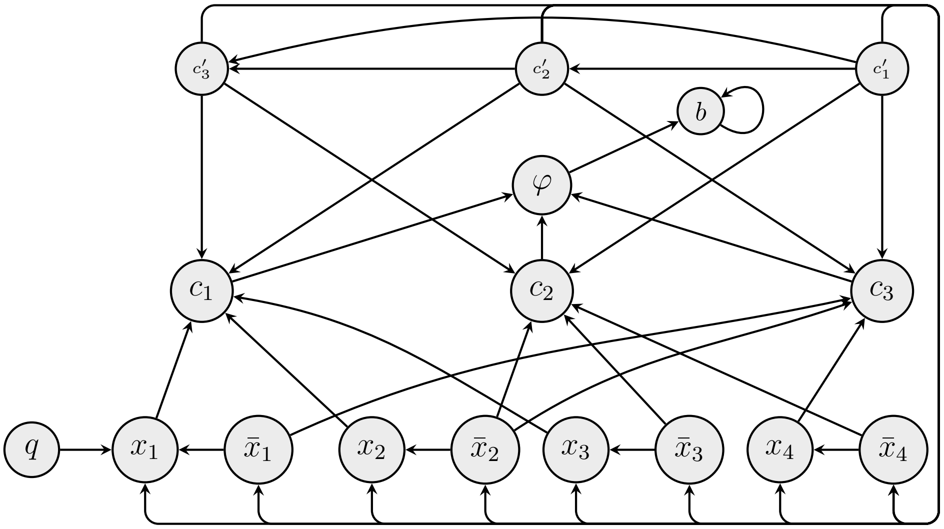


Figure 6: Illustration of the AF F_{φ,x_1} for $\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_2 \vee x_4)$.

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

Now if z is in a minimal model M then the corresponding M^* has maximal range and by the definition of M^* also $z \in M^*$, and thus z is credulously accepted. If z is credulously accepted the corresponding stage extension is of the form M^* (as the other candidates do not contain arguments $x \in X$) and thus there is a minimal model M also containing z.

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

For each $F \in \mathsf{acyc}$ we have $F \in \mathcal{G}$, with $\mathcal{G} \in \{\mathsf{wcyc}, \mathsf{ocf}, \mathsf{coherent}, \mathsf{uniqpref}\}$, and thus the previous result generalizes to all these classes when $dist_{\mathcal{G}}(F) = 1$.

Recall that Propositions 11 and 12 directly yield corresponding hardness results for the classes coherent and stablecons for preferred and semi-stable semantics. It thus remains to consider the class uniqpref.

Proposition 14. *Cred_{sem} is Σ_2^P-hard and Skept_{prf}, Skept_{sem} are Π_2^P-hard for AFs F with $dist_{\mathsf{uniqpref}}(F) = 1$.*

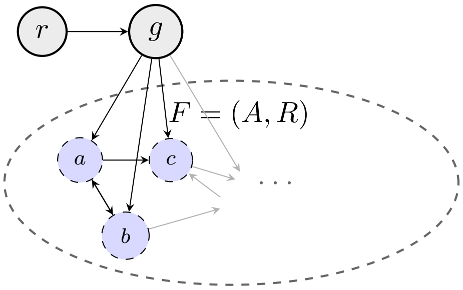


Figure 7: An illustration of the construction in the proof of Proposition 14

Proof. Consider an arbitrary AF $F = (A, R)$, and consider the modified AF $F' = (A \cup \{r, g\}, R \cup \{(r, g)\} \cup \{(g, a) \mid a \in A\})$ (see also Figure 7). Then, $adm(F') = \{E \cup \{r\} \mid E \in adm(F)\} \cup \{\emptyset\}$.

First, we have that each non-empty admissible set must contain r, because all arguments except a, g are attacked by g, g is not acceptable and r is the only argument attacking g. Further g is the only argument attacked by r and hence for each $S \subseteq A$ we have that $S \cup \{r\} \in adm(F')$ iff $S \in adm(F)$.

As an immediate consequence of this $prf(F') = \{E \cup \{r\} \mid E \in prf(F)\}$ and by the fact that the range of admissible sets is only increased by $\{r, g\}$ we also have that $sem(F') = \{E \cup \{r\} \mid E \in sem(F)\}$.

Moreover, $dist_{\mathsf{uniqpref}}(F') = 1$ since deleting argument r would result in an AF where $\{g\}$ is the only non-empty admissible set. This is by the fact that g is unattacked in the resulting AF and itself attacks all the remaining arguments. Hence credulous and skeptical acceptance (under the considered semantics) on arbitrary AFs reduces to AFs F with $dist_{\mathsf{uniqpref}}(F) = 1$. □

Table 3 summarizes our results which are all negative in the sense that full second-level complexity is reached when fragments are parameterized in a "syntactic" way; only acyc yields some positive results (due to [8]). Here FPT (fixed-parameter tractability) means that for a fixed distance, a problem can be solved in polynomial time and the order of the polynomial time bound does not depend on the distance. However, we showed here as a new result that under the stage semantics, full complexity is obtained even for AFs F with $dist_{\mathsf{acyc}}(F) \geq 1$ (this strengthens the NP/coNP-hardness results in [8]).

Table 3: Complexity when parameterized by the distance to a sub-class $\mathcal{G}$ (hardness holds for AFs F of $dist_{\mathcal{G}}(F) = 1$).

$\mathcal{G}$	Skept_{prf}	Cred_{sem}	Skept_{sem}	Cred_{stg}	Skept_{stg}
acyc	FPT	FPT	FPT	Σ_2^P-c	Π_2^P-c
wcyc	Π_2^P-c	Σ_2^P-c	Π_2^P-c	Σ_2^P-c	Π_2^P-c
stablecons	Π_2^P-c	Σ_2^P-c	Π_2^P-c	Σ_2^P-c	Π_2^P-c
ocf	Π_2^P-c	Σ_2^P-c	Π_2^P-c	Σ_2^P-c	Π_2^P-c
coherent	Π_2^P-c	Σ_2^P-c	Π_2^P-c	Σ_2^P-c	Π_2^P-c
uniqpref	Π_2^P-c	Σ_2^P-c	Π_2^P-c	Σ_2^P-c	Π_2^P-c

5. Extension-Based Distance Measures

Here we consider different distance measures which take the number of extensions into account and thus naturally apply only to the semantical subclasses of AFs, i.e. stablecons, coherent, and uniqpref.

We start by generalizing the class stablecons for semantics sem and stg. In fact, for stable-consistent AFs we have that each semi-stable (resp. stage) extension has a range that covers the whole set of arguments. Hence a natural approach to relax this definition is to bound the number of arguments which are not in the range of the extensions.

Definition 10. *For a semantics σ and $k \geq 0$, we call an AF $F = (A, R)$ k-*stable-consistent *under σ if for each $E \in \sigma(F)$, $|A \setminus E_R^+| \leq k$ holds. We use* stablecons$_\sigma^k$ *to denote the respective classes of AFs for given k and σ.*

Theorem 1. *For AFs in* stablecons$_\sigma^k$ *($\sigma \in \{sem, stg\}$),* Cred_σ *and* Skept_σ *are in* P^{NP}.

Proof. Consider an AF $F = (A, R)$. We can decide credulous (resp. skeptical) acceptance by considering all candidates for the range, i.e. sets S with $|A \setminus S_R^+| \leq k$, and test whether there is an extension with range S that contains (resp. does not contain) the queried argument. But while (for fixed k) there are just polynomial many sets S the testing part is still Σ_2^P / Π_2^P-hard.

To overcome this we introduce two shortcuts. To this end let $\mathcal{S}$ be the set of ranges we have to consider, which is initialized by $\mathcal{S} = \{S \subseteq A \mid |A \setminus S| \leq$

$k\}$. First, if we know that there is an extension with range S, we do not have to consider subsets of S for the range of extensions, and thus delete them from $\mathcal{S}$. Second, we only consider maximal sets $S \in \mathcal{S}$. Then we know that there is no extension having a superset of S as range and we can simplify the check for extension by testing whether there is an admissible (resp. conflict-free) set having range S; this is clearly in NP (simply guess a set and check whether it is admissible / conflict-free and whether the range fits).

In what follows we summarize these considerations and give a P^{NP} procedure deciding Cred_{sem} (resp. $\mathsf{coSkept}_{sem}$) for the class $\mathsf{stablecons}^k_{sem}$ of AFs, which answers the query by calling either *accept* or *reject*:

```
Require: AF $F = (A, R)$, argument $\alpha \in A$
Ensure: accepts the input iff $\alpha$ is credulously accepted (resp. not skeptically
        accepted) in $F$ w.r.t. $sem$, rejects otherwise
Initialize $\mathcal{S} \leftarrow \{S \subseteq A \mid |A \setminus S| \leq k\}$.
while $\mathcal{S} \neq \emptyset$ do
    pick $S \in \mathcal{S}$ with maximum cardinality
    if there is an $E \in adm(F)$ with $E^+_R = S$ and $\alpha \in E$ (resp. $\alpha \notin E$ ) then
        accept
    else if there is an $E \in \sigma(F)$ with $E^+_R = S$ then
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S' \mid S' \subseteq S\}$
    else
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S\}$
    end if
end while
reject
```

If we replace adm by cf, we obtain the corresponding decision procedure for stage semantics, i.e., for the class $\mathsf{stablecons}^k_{stg}$. □

Next, we parameterize coherence.

Definition 11. *An AF F is k-*coherent*, where $k \geq 0$, if $|prf(F) \setminus stb(F)| \leq k$. We use* $\mathsf{coherent}^k$ *to denote the respective class of AFs for given k.*

We do not consider the corresponding definitions for sem and stg as in Definition 10: The reason is that either a stable extension exists (hence, the AF is stable-consistent), or the parameter k, as used in Definition 11, would simply mention the number of semi-stable, resp. stage, extensions. In fact, we will consider these classes of AFs of bounded solution cardinality at the end of this section.

In the case of k-coherent AFs, the potential exponential number of stable extensions appears to cause additional difficulties. While we are unable to provide a hardness result using standard reductions at the moment, we provide a slightly weaker result (using randomized reductions) which still suggests that parameterized coherence does not allow for more efficient algorithms.

Theorem 2. Skept_{prf} *for AFs in* $\mathsf{coherent}^k$ *is* Π_2^P*-hard under randomized reductions; hardness holds even for* $k = 1$*.*

Proof. Consider the following promise problem: Given a QBF $\Phi = \forall Y \exists Z \varphi(Y, Z)$ together with the fact that $\exists^{\leq 1} Y \forall Z \neg\varphi(Y, Z)$ is true, decide whether Φ is true. By results in [27] and [26] (extension to QBFs), this problem is Π_2^P-hard under randomized reductions. Now one can apply the reduction presented in the proof of Proposition 3 from QBFs to Skept_{prf} (see also Figure 1). The sets Y with $\forall Z \neg\varphi(Y, Z)$ are in one-to-one correspondence to the preferred extensions which are not stable. Hence using the promise problem results in an AF from the class $\mathsf{coherent}^1$. □

The final class we discuss is $\mathsf{uniqpref}$. The natural distance here is to consider frameworks which possess at most k preferred extensions. We will also apply the same idea for semi-stable and stage semantics.

Definition 12. *Let* $\sigma \in \{prf, sem, stg\}$*. We denote by* sol_σ^k *the class of all AFs* F *such that* $|\sigma(F)| \leq k$*.*

Theorem 3. *For AFs* $F \in \mathsf{sol}_{prf}^k$*,* Skept_{prf} *is in* P^{NP}*.*

Proof. We provide an algorithm which iteratively constructs the set $prf(F)$. The set $\mathcal{E}$ serves this purpose and is initialized by $\mathcal{E} := \emptyset$. At each stage of the algorithm we construct a new preferred extension E as follows: Start with $E = \emptyset$, and iterate over all arguments a asking an NP-oracle whether there is a complete extension C of the given framework such that $E \cup \{a\} \subseteq C$ and there is no $E' \in \mathcal{E}$ such that $C \subseteq E'$. If the oracle returns *yes*, add a to E and proceed with the next arguments. In the end, E is either a preferred extension (i.e. we cannot add further arguments) or the empty set. If $E \neq \emptyset$, simply add E to the set $\mathcal{E}$ and proceed with constructing the next preferred extension. If $E = \emptyset$, there is no non-empty complete extension that is not already contained in one of the extensions in $\mathcal{E}$, and hence the algorithm terminates. If $\mathcal{E} = \emptyset$, add $\emptyset$ to $\mathcal{E}$.

By the assumption $F \in \mathsf{sol}_{prf}^k$, $prf(F)$ is of polynomial size and for each extension only a linear number of steps (iterating over all arguments) is needed. Hence the overall run-time is polynomial (using an NP oracle). □

Table 4: Complexity when the AFs belong to a sub-class $\mathcal{G}$.

$\mathcal{G}$	Skept_{prf}	Cred_{sem}	Skept_{sem}	Cred_{stg}	Skept_{stg}
$\mathsf{stablecons}_{\sigma}^{k}$	–	in P^{NP}	in P^{NP}	in P^{NP}	in P^{NP}
$\mathsf{coherent}^{k}$	Π_2^P-c	in P^{NP}	in P^{NP}	in P^{NP}	in P^{NP}
$\mathsf{sol}_{\sigma}^{k}$	in P^{NP}	in P^{NP}	in P^{NP}	in P^{NP}	in P^{NP}

Theorem 4. *For AFs in* $\mathsf{sol}_{\sigma}^{k}$ *(*$\sigma \in \{sem, stg\}$*),* Cred_{σ} *and* Skept_{σ} *are in* P^{NP}.

Proof. Let $F = (A, R)$. The algorithm we present below is quite similar to the algorithm in the proof of Theorem 3, but instead of computing $\sigma(F)$, we construct here $\sigma^{+}(F) := \{E_R^+ \mid E \in \sigma(F)\}$. Again, below we use $\mathcal{E}$ as the set of the currently computed elements from $\sigma^{+}(F)$, initialized as $\mathcal{E} := \emptyset$ and iteratively extend it. Moreover, let $b_{sem} = com$ and $b_{stg} = cf$.

Within each iteration, start with $S = \emptyset$ and iterate over all arguments $a \in A$ asking an NP-oracle whether there is a set $C \in b_{\sigma}(F)$ such that $S \cup \{a\} \subseteq C_R^+$ and there is no $T \in \mathcal{E}$ such that $C_R^+ \subseteq T$. If the oracle returns *yes*, add a to S and proceed with the next arguments. In the end, S is either the range of a σ-extension or the empty set. If $S \neq \emptyset$, simply add S to the set $\mathcal{E}$ and continue with constructing the next set. If $S = \emptyset$, there is no non-empty set in $b_{\sigma}(F)$ such that S_R^+ is not already contained in a set of $\mathcal{E}$, and hence the algorithm terminates. If $\mathcal{E} = \emptyset$, add $\emptyset$ to $\mathcal{E}$.

By assumption, the number of σ-extensions is bounded and so is the size of $\sigma^{+}(F)$. To decide Cred_{σ} (resp. co-Skept_{σ}) for an argument a, iterate over all sets $S \in \sigma^{+}(F)$ and ask an NP oracle whether there is an $E \in b_{\sigma}(F)$ such that $E_R^+ = S$ and $a \in E$ (resp. $a \notin E$). □

Our results are summarized in Table 4. Recall that the Π_2^P-hardness of Skept_{prf} for class $\mathsf{stablecons}_{\sigma}^{k}$ was shown for randomized reductions. Furthermore, recall that $\mathsf{coherent}^{k}$ for $\sigma \in \{sem, stg\}$ reduces either to $\mathsf{stablecons}$ or to $\mathsf{sol}_{\sigma}^{k}$. We did not consider k-stable-consistent AFs under prf since the full complexity was already reached for the class of stable-consistent AFs (recall Table 2).

6. Complexity-Sensitive Procedures

In this section we describe complexity-sensitive decision procedures for skeptical and credulous acceptance problems, which are on the second level of the polynomial hierarchy. Our procedures follow the lines of Theorems 1, 3, and 4. As a result, the procedures are complexity-sensitive w.r.t. the classes exhibiting lowered complexity, as shown in the previous section (recall Table 4). At the same time, the procedures are applicable to the general second-level skeptical and credulous acceptance problems.

The general framework implemented by our procedures exploits NP-oracles, and uses oracle calls to decide NP-decidable relaxations of the input instance by over- and/or under-approximating the acceptance conditions of the problem at hand. The relaxation is iteratively strengthened based on the answers provided by the oracle calls. More specifically, at the beginning of the procedures, the candidate extensions are the NP-decidable admissible sets (or, alternatively, complete extensions) in case of preferred and semi-stable semantics, or conflict-free sets in case of stage semantics. We refer to the semantics that characterizes the initial candidate extensions as the chosen *base semantics*. Starting from the initial candidate extensions, the remaining set of candidate extensions is then non-trivially reduced in an iterative fashion based on the results returned by the previous oracle calls.

We will now explain in detail our procedures, starting with skeptical acceptance for preferred semantics.

6.1. Procedure for Preferred Semantics

First let us consider preferred semantics and the problem of skeptical acceptance (recall that credulous acceptance is already in NP). Our procedure builds on Theorem 3 and, as an additional shortcut, uses the following observation: In coherent AFs an argument a is skeptically accepted iff a is not attacked by a preferred extension. In general, one has to drop the “if” direction. However, we still have that an argument is skeptically accepted only if a is not attacked by a preferred extension. As the latter is equivalent to a not being attacked by an admissible set or a complete extension we can test it with one call to an NP-oracle.

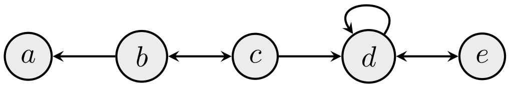
Our complexity-sensitive procedure for deciding skeptical acceptance of an argument w.r.t. preferred semantics is presented as Algorithm 1, $Skept_{prf}(F, \alpha)$. First, the procedure applies the just explained shortcut. If this does not result in a decision for the input instance, the algorithm presented in the proof of Theorem 3 is applied, yielding a complexity-sensitive procedure for deciding skeptical acceptance in the class sol^{k}_{prf}. In the algorithm, we are free to choose the base semantics σ, i.e., the semantics used in the NP search procedures, to be either admissible or complete semantics. In practice, this choice might influence the actual running time of the procedure; experimental results on the impact of the choice are provided in Section 8. Finally, notice that in the following algorithms we use the function NP-ORACLE not only to get yes/no answers for NP-queries, but also to obtain certain extensions. This does not fully match to the formal notion of a NP-oracle, which only return yes or no, but to a functional variant of it. However, NP-ORACLE can be easily implemented by a linear number of calls to a classical NP-oracle (cf. proof of Theorem 3).

Algorithm 1 $Skept_{prf}(F, \alpha)$

```
Require: AF $F = (A, R)$, argument $\alpha \in A$, $\sigma \in \{adm, com\}$
Ensure: accepts the input iff $\alpha$ is skeptically accepted in $F$ w.r.t. $prf$, rejects
        otherwise
 1: $\mathcal{E} \leftarrow \emptyset$
 2: if NP-ORACLE $(\exists E \in \sigma(F) : E$ attacking $\alpha)$ then
 3:     reject
 4: end if
 5: while $E \leftarrow$ NP-ORACLE $(\exists E \in \sigma(F) : \alpha \notin E, \nexists E' \in \mathcal{E} : E \subseteq E')$ do
 6:     while $E' \leftarrow$ NP-ORACLE $(\exists E' \in \sigma(F) : E \subset E')$ do
 7:         $E \leftarrow E'$
 8:     end while
 9:     if $\alpha \notin E$ then
10:         reject
11:     else
12:         $\mathcal{E} \leftarrow \mathcal{E} \cup \{E\}$
13:     end if
14: end while
15: accept
```

Algorithm 1 answers the skeptical acceptance of an argument α for an AF F under preferred semantics. As before the algorithm answers the query by calling either *accept* or *reject*. It first tries to construct an admissible set or complete extension, depending on the choice of the base semantics σ, which attacks α. In this case we can immediately reject the query. The outer while loop in line 5 now computes an admissible set or complete extension E, excluding α and ensuring that E is not a subset of an admissible set or complete extension E', which was checked before, i.e. is in $\mathcal{E}$. If such an E is found it is iteratively extended in the inner while loop, as long as this is possible. Afterwards E must be a subset maximal admissible set or complete extension, hence a preferred extension. Then we check if $\alpha \notin E$. In this case we can reject the skeptical acceptance of α in F.

Example 2. *To illustrate the behavior of Algorithm 1 consider the following AF.*



The AF has two preferred extensions $\{b, e\}$ and $\{a, c, e\}$ with e being the only skeptically accepted argument.

Let us first test b for skeptical acceptance with Algorithm 1 and base semantics adm. Then, already in line 2 of the algorithm we obtain an admissible set attacking b, e.g. $\{a, c\}$, and thus reject b.

Now consider the argument e and base semantics com. As e is only attacked by d and d is in no complete extension the shortcut in line 2 does not apply. In line 5 we enter the while loop as for instance the emptyset $\emptyset$ is a complete extension that does not contain e. We pick $\emptyset$ as our candidate extension and in line 6 we enter the second while loop as $\{b\}$ is complete and a superset of $\emptyset$. In line 7 we choose $\{b\}$ for being our new candidate solution. Then we go back to line 6 and as $\{b, e\}$ is complete and a superset of $\{b\}$ we again update the candidate extension to be $\{b, e\}$. Now as $\{b, e\}$ is maximal we skip the inner loop and in line 10 we check whether e is contained in our candidate. As $e \in \{b, e\}$ we enter the else part and add $\{b, e\}$ to the set of already considered candidates. Now back in line 5 we ask for a complete extension not contained in $\{b, e\}$ and not containing e. As the former is only satisfied by the set $\{a, c, e\}$, the query fails and we skip the outer loop. Finally, as we could not construct a counter example for the skeptical acceptance of e we accept it.

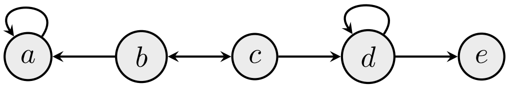
6.2. Procedures for Semi-Stable and Stage Semantics

We now turn to the problems of skeptical and credulous acceptance under semi-stable and stage semantics, and describe our complexity-sensitive procedures for these problems. Our procedures combine the results of Theorems 1 and 4. For each of the cases of skeptical and credulous acceptance, the procedures for semi-stable and stage semantics only differ in the choice of the base semantics, and can hence be presented as a unified procedure.

Our procedure for deciding credulous acceptance for semi-stable semantics (using base semantics $\sigma \in \{adm, com\}$) and credulous acceptance for stage semantics (using base semantics $\sigma = cf$) is presented as Algorithm 2, $Cred_\theta(F, \alpha)$. The procedure first applies the algorithm from the proof of Theorem 1, but bounds the search depth to a fixed constant d as follows: We initialize $\mathcal{S}$ to $\{S \subseteq A \mid |A \setminus S| \leq d+1\}$ and only consider sets S with $|A \setminus S| \leq d$ in the first while loop. Hence, if $\mathcal{S} = \emptyset$ in line 13 then all the sets S with $|A \setminus S| = d+1$ are subsets of ranges of already considered extensions and thus no further extensions exists (as indeed all sets with $|A \setminus S| > d+1$ are subsets of sets with $|A \setminus S| = d+1$). Hence the algorithm has to reject the instance.

If the algorithm from the proof of Theorem 1 does not result in deciding the input instance, the algorithm from the proof of Theorem 4 is applied. Hence Algorithm 2 is complexity-sensitive for $\mathsf{stablecons}^{\leq d}$ and sol^k, the first one expressing that is complexity-sensitive w.r.t. k'-stable-consistency as long k' is below some threshold d.

Example 3. *To illustrate the behavior of Algorithm 2, consider the following AF F.*



The semi-stable extensions are $\{b\}$ and $\{c,e\}$ with ranges $\{b\}^+ = \{a,b,c\}$ and $\{c,e\}^+ = \{b,c,d,e\}$.

Let us first check whether c is credulously accepted with base-semantics adm and $d = 1$. In the first while loop we have to consider the sets $\{a,b,c,d,e\}$, $\{b,c,d,e\}$, $\{a,c,d,e\}$, $\{a,b,d,e\}$, $\{a,b,c,e\}$ and $\{a,b,c,d\}$. In the first iteration we have to pick $S = \{a,b,c,d,e\}$. As there is no admissible set with range $\{a,b,c,d,e\}$, i.e. no stable extension, we simply delete $\{a,b,c,d,e\}$ from $\mathcal{S}$. As $\mathcal{S}$ still contains sets S with $|A \setminus S| \leq 1$ we pick another set S for instance $\{b,c,d,e\}$. As $\{c,e\}$ is an admissible set with range $\{b,c,d,e\}$ we end up in line 6 and accept c.

Now we consider the same argument c, but we set the parameter d to 0. In line 2 we initialize the set $\mathcal{S}$ with $\{\{a,b,c,d,e\},\{b,c,d,e\},\{a,c,d,e\}, \{a,b,d,e\}, \{a,b,c,e\},\{a,b,c,d\}\}$. Again we enter the first loop and pick $S = \{a,b,c,d,e\}$ and then delete it from $\mathcal{S}$. Then there are not further sets in $\mathcal{S}$ with $|A \setminus S| \leq 0$ and thus we skip the first loop, As $\mathcal{S} \neq \emptyset$ we next enter the second loop. Now we pick an admissible set containing c, for instance $\{c\}$. Next we enter the inner loop to maximize the range of the candidate solution. As $\{c,e\}$ has a larger range we update our E to $\{c,e\}$ and as this is range maximal we skip the inner loop. Now as $c \in \{c,e\}$ the algorithms accepts c.

Our procedure for skeptical acceptance under semi-stable and stage semantics, presented as Algorithm 3, works in a similar fashion as Algorithm 2 exploiting

Algorithm 2 $Cred_\theta(F, \alpha)$ (where $\theta \in \{sem, stg\}$)

```
Require: AF $F = (A, R)$, argument $\alpha \in A$, integer $d$, base-semantics $\sigma \in \{adm, com\}$ for $\theta = sem$; $\sigma = cf$ for $\theta = stg$;
Ensure: *accept*s the input iff $\alpha$ is credulously accepted in $F$ w.r.t. $\theta$, *reject*s otherwise
$\mathcal{E} \leftarrow \emptyset$;
$\mathcal{S} \leftarrow \{S \subseteq A \mid |A \setminus S| \leq d + 1\}$
while $\exists S \in \mathcal{S} : |A \setminus S| \leq d$ do
    pick $S \in \mathcal{S}$ with maximum cardinality
    if NP-ORACLE$\big(\exists E \in \sigma(F) : E_R^+ = S, \alpha \in E\big)$ then
        *accept*
    else if NP-ORACLE$\big(\exists E \in \sigma(F) : E_R^+ = S\big)$ then
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S' \mid S' \subset S\}$
    else
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S\}$
    end if
end while
if $\mathcal{S} = \emptyset$ then
    *reject*
end if
while $E \leftarrow$ NP-ORACLE $\big(\exists E \in \sigma(F) : \alpha \in E, \nexists E' \in \mathcal{E} : E_R^+ \subseteq E'\big)$ do
    while $E' \leftarrow$ NP-ORACLE $\big(\exists E' \in \sigma(F) : E_R^+ \subset E'^+_R\big)$ do
        $E \leftarrow E'$
    end while
    if NP-ORACLE$\big(\exists E' \in \sigma(F) : E'^+_R = E_R^+ : \alpha \in E\big)$ then
        *accept*
    else
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{E_R^+\}$
    end if
end while
*reject*
```

Theorems 1 and 4 once more. The key difference to Algorithm 2 is that, instead of constructing a witness for an argument to be credulously accepted, it attempts to construct a counter-example for an argument to be skeptically accepted, and only accepts if no such counter-example can be constructed.

6.3. Materializing the NP-Oracles

Our procedures just described make heavy use of NP-oracle queries for the base semantics applied in each case. Hence, in practical implementations of our procedures, one wants to make use of (existing) NP-solvers for the base semantics for deciding the oracle queries. However, some of our queries are rather non-standard, and thus are unlikely to be directly implemented in a solver for the base semantics.

We see two ways to deal with this problem. One alternative is to translate the oracle queries to standard AF reasoning tasks, like credulous/skeptical acceptance of an argument, existence of a stable extension, or existence of a non-empty extension, by modifying the frameworks for the queries.

As a second alternative, one can characterize argumentation semantics within other suitable formalisms, and then apply a decision procedure for the chosen formalism as the underlying NP-oracle for deciding the queries. In this approach, the key is to find a formalism that is well-suited for encoding the NP-queries, most preferably in an incremental manner, and for which at the same time efficient off-the-shelves implementations are readily available.

In the next section, we will present an instantiation of our procedures based on the second alternative, using Boolean satisfiability (SAT) as a well-suited formalism. In other words, the SAT-based instantiation is based on representing the NP-queries in a natural way in propositional logic, and applying readily available and highly optimized state-of-the-art SAT-solvers for decision of the queries. As will be later shown in Section 8, it turns out that an implementation of the SAT-based instantiation of our procedures provide a competitive approach for the second-level argumentation problems considered.

Before describing details on the SAT-based instantiation of our procedures, let us shortly outline the other possibility, i.e., the translation-based approach based on translating the oracle queries to standard AF reasoning tasks. We exemplify this alternative by providing translations for two of our NP-queries. We start with the NP query for an admissible/conflict-free set of a given range used in lines 5,7 and 20, respectively, of Algorithms 2 and 3, respectively, which can be translated to computing a stable extension of a modified AF.

Algorithm 3 $Skept_\theta(F, \alpha)$ (where $\theta \in \{sem, stg\}$)

```
Require: AF $F = (A, R)$, argument $\alpha \in A$, integer $d$, base-semantics $\sigma \in$
         $\{adm, com\}$ for $\theta = sem$; $\sigma = cf$ for $\theta = stg$;
Ensure: accepts the input iff $\alpha$ is skeptically accepted in $F$ w.r.t. $\theta$, rejects
        otherwise
$\mathcal{E} \leftarrow \emptyset$;
$\mathcal{S} \leftarrow \{S \subseteq A \mid |A \setminus S| \leq d + 1\}$
while $\exists S \in \mathcal{S} : |A \setminus S| \leq d$ do
    pick $S \in \mathcal{S}$ with maximum cardinality
    if NP-Oracle$\big(\exists E \in \sigma(F) : E_R^+ = S, \alpha \in E\big)$ then
        reject
    else if NP-Oracle$\big(\exists E \in \sigma(F) : E_R^+ = S\big)$ then
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S' \mid S' \subset S\}$
    else
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S\}$
    end if
end while
if $\mathcal{S} = \emptyset$ then
    accept
end if
while $E \leftarrow$ NP-Oracle $\big(\exists E \in \sigma(F) : \alpha \notin E, \nexists E' \in \mathcal{E} : E_R^+ \subseteq E'\big)$ do
    while $E' \leftarrow$ NP-Oracle $\big(\exists E' \in \sigma(F) : E_R^+ \subset E'^+_R\big)$ do
        $E \leftarrow E'$
    end while
    if NP-Oracle$\big(\exists E' \in \sigma(F) : E'^+_R = E_R^+ : \alpha \notin E\big)$ then
        reject
    else
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{E_R^+\}$
    end if
end while
accept
```

Proposition 15. *Let $F = (A, R)$ be an AF and $S \subseteq A$, and define F_S^{stg}, F_S^{sem} as follows:*

$$F_S^{stg} = (A \cap S, (R \cap (S \times S)) \cup \{(b, b) \mid b \rightarrowtail (A \setminus S)\})$$
$$F_S^{sem} = (A \cap S, (R \cap (S \times S)) \cup \{(b, b) \mid b \rightarrowtail (A \setminus S) \vee (A \setminus S) \rightarrowtail b\})$$

This construction ensures the following correspondence:

i. $E \in stb(F_S^{stg})$ iff $E \in cf(F)$ and $E_R^+ = S$; and
ii. $E \in stb(F_S^{sem})$ iff $E \in adm(F)$ and $E_R^+ = S$

Proof. To prove (i) let us first consider a $E \in stb(F_S^{stg})$. As the construction of F_S^{stg} does not drop conflicts in S we have that $E \in cf(F)$ and further as $E \in stb(F_S^{stg})$ and the attacks origin from F we have that $E_R^+ \supset S$. Moreover there is no argument $a \in E$ that attacks an argument in $A \setminus S$ as such an a is self-attacking in F_S^{stg}. Hence also $E_R^+ = S$. Now let us consider $E \in cf(F)$ and $E_R^+ = S$. As $E_R^+ = S$ also $E \subset S$ and there is no argument $a \in E$ attacking an element from $A \setminus S$. Hence $E \in cf(F_S^{stg})$ and as all attacks within S are preserved in F_S^{stg} also $E \in stb(F_S^{stg})$.

Now to prove (ii) consider $E \in stb(F_S^{sem})$. We show that $E \in adm(F)$. As before we have that $E \in cf(F)$ and $E_R^+ = S$. Next consider an argument a attacking an argument b in E. By construction $a \in S$ (otherwise b would be self-attacking and thus not in E), and as $E_R^+ = S$ there is a $c \in E$ attacking a. Now consider an $E \in adm(F)$ and $E_R^+ = S$. Again as $E_R^+ = S$ also $E \subset S$ and there is no argument $a \in E$ attacking an $A \setminus S$. Furthermore as $E \in adm(F)$ and $E_R^+ = S$ there is no argument $b \in A \setminus S$ attacking E. Hence $E \in cf(F_S^{stg})$ and as all attacks within S are preserved in F_S^{sem} also $E \in stb(F_S^{sem})$. □

Now consider the NP-query in line 6 of Algorithm 1, which asks whether there is a superset of the current candidate solution that is still admissible/complete.

Proposition 16. *For an AF $F = (A, R)$, $S \in adm(F)$, and F_S defined as the AF $(A \setminus S_R^+, R \cap ((A \setminus S_R^+) \times (A \setminus S_R^+)))$ it holds that $S \in prf(F)$ iff $adm(F_S) = \{\emptyset\}$.*

Proof. First let us consider the case where $S \notin prf(F)$, then there exists a non-empty $S' \subseteq A \setminus S$ such that $S \cup S' \in adm(F)$. Now it is easy to see that also $S' \in adm(F_S)$. Next let us consider the case where there exists a non-empty $S \in adm(F_S)$. Then one can easily show that $S \cup S' \in adm(F)$ and as S' is disjoint from S, also $S \notin prf(A, R)$. □

In the rest of the paper, however, our main focus will be on the second alternative, developing SAT-based instantiations of our generic complexity-sensitive procedures.

7. SAT-Based Complexity-Sensitive Procedures

In this section we describe instantiations based on SAT for the complexity-sensitive procedures (see Algorithms 1, 2, and 3) described in the previous section. In the SAT-based instantiations, the queries, i.e. the inputs to the NP-oracle, are propositional formulas, encoding the currently remaining candidate extensions under the chosen base semantics.

Harnessing SAT for instantiating the procedures has several benefits. First, the NP-oracle calls can be encoded in a natural way using propositional logic. During the main iterative loop of the procedures, restricting the set of candidate extensions further, based on already found counter-examples, can be achieved by iteratively adding new clauses to the propositional formula, which encodes the initial candidate extensions. Hence the input to the SAT-solver, acting as the NP-oracle, needs not to be built from scratch before each oracle call. This can be seen as a form of *no-good learning* which further prunes the possible IN/OUT-labellings [28] of the arguments on the level of the propositional encoding. Furthermore, any off-the-shelf complete SAT-solver can hence be used as the NP-oracle. This relieves the implementer from the non-straight-forward task of implementing the actual NP search procedure. By applying SAT as the NP-oracle, the highly optimized and efficient conflict-driven clause learning (CDCL) SAT-solvers available today can be directly exploited. This especially includes CDCL SAT-solvers that can be applied *incrementally*, which basically means that (after a satisfiable oracle call) the SAT-solver can maintain its state from one oracle call to the next without needing to start search from scratch when adding further restrictions on the set of candidate extensions on the propositional level. Furthermore, the clauses learned within the CDCL SAT search, imposing further restrictions on the candidate extensions, can be kept from one iteration to another.

7.1. A SAT-Based Complexity-Sensitive Framework

The main structure of our SAT-based instantiation, based on the class sol^k_σ, which applies well to all semantics under our consideration, can be represented as a unifying framework for all the semantics. In what follows we consider semantics $\sigma \in \{prf, sem, stg\}$, a reasoning mode $M \in \{\mathsf{Skept}, \mathsf{Cred}\}$, an AF $F = (A, R)$, and an argument $\alpha \in A$. The generic structure of our SAT-based framework is shown in Algorithm 4.

Notice that in the algorithms we use the function SAT not only to get a yes answer if the provided formula is satisfiable and a no answer if the formula is unsatisfiable, but also to obtain a model in the former case. The dual function UNSAT returns yes iff the formula is unsatisfiable.

Algorithm 4 $Decide_\theta(F, \alpha, M)$ (where $\theta \in \{prf, sem, stg\}$)

```
Require: AF $F = (A, R)$, argument $\alpha \in A$, base-semantics $\sigma \in \{adm, com\}$
         for $\theta \in \{prf, sem\}$; $\sigma = cf$ for $\theta = stg$; and reasoning mode $M \in$
         $\{\mathsf{Skept}, \mathsf{Cred}\}$;
Ensure: *accept*s the input iff $M = \mathsf{Skept}$ and $\alpha$ is skeptically rejected or $M =$
         $\mathsf{Cred}$ and $\alpha$ is credulously accepted in $F$ w.r.t. $\sigma$, *reject*s otherwise
 1: if $M = \mathsf{Skept}$ then
 2:     $q \leftarrow \neg x_\alpha$
 3: else if $M = \mathsf{Cred}$ then
 4:     $q \leftarrow x_\alpha$
 5: end if
 6: $\varphi \leftarrow \varphi_\sigma(F) \wedge q \wedge \textsc{Shortcuts}_\theta(F, \alpha, M)$
 7: while $I \leftarrow$SAT $(\varphi)$ do
 8:     while $I' \leftarrow$ SAT $(\psi^I_\theta(F) \wedge q)$ do
 9:         $I \leftarrow I'$
10:     end while
11:     if UNSAT $\big(\psi^I_\theta(F)\big)$ then
12:         *accept*
13:     else
14:         $\varphi \leftarrow \varphi \wedge \gamma^I_\theta$
15:     end if
16: end while
17: *reject*
```

We will provide details for the generic concepts $\psi^I_\sigma(F)$ and γ^I_σ as well as for the function $\textsc{Shortcuts}_\sigma(F, \alpha, M)$ for the different semantics separately below. Let us note at this point already that the procedure might terminate in $\textsc{Shortcuts}_\sigma(F, \alpha, M)$ in certain cases. Then, we also terminate the procedure with the status returned by $\textsc{Shortcuts}_\sigma(F, \alpha, M)$, i.e. *accept* or *reject*.

Overall, the procedure works as follows. Depending on the reasoning mode, we test whether there is an extension containing α, or whether there is an extension not containing α, hence accepting α credulously or rejecting it for skeptical reasoning. This is encoded via the query-literal q. In line 6, a formula is built to encode extensions of the base semantics, i.e. not taking maximality into account, together with the query q as well as semantics-specific *shortcuts* that can be applied for pruning the search space via learning inferred information; this will be discussed later in more detail. The SHORTCUTS function allows for refining the base encoding using the inferred information it outputs. The loop in line 7 follows the ideas of the main loops in the Algorithms 1, 2 and 3: starting with a model that corresponds to a set of arguments satisfying the base semantics and our query q, each iteration then extends this set to a larger one satisfying q until we have a maximal set satisfying q. In line 11, the condition q is dropped for testing whether the set is maximal among all sets, i.e. whether it is an extension. If this is the case, the algorithm accepts. Otherwise, we learn that none of the smaller sets can be an extension (line 14). Finally, after excluding all sets satisfying the base semantics and q from being a valid extension, the algorithm rejects the query (line 17).

7.2. Instantiating the Framework

A key aspect of instantiating the SAT-based framework is how the AF reasoning tasks are encoded as propositional formulas over the variables $X = \{x_a \mid a \in A\}$ and $Y = \{y_a \mid a \in A\}$ such that the models of the formulas are in correspondence with certain sets of arguments. The intuition behind the variables is that x_a is true iff a is in the set, and y_a is true iff a is in the range of the set.

7.2.1. Encoding the Base Semantics

To ensure the relation between the set and its range, we apply the formula

$$y_a \leftrightarrow x_a \vee \bigvee_{(b,a)\in R} x_b$$

together with a propositional encoding that restricts the models to correspond to particular types of AF extensions. To this end, we use the following propositional formulas $\varphi_{cf}(F)$, $\varphi_{adm}(F)$, and $\varphi_{com}(F)$, respectively, to encode the conflict-free sets, admissible sets, and complete extensions, respectively for a given AF $F = (A, R)$:

$$\begin{aligned}
\varphi_{cf}(F) &= \bigwedge_{(a,b)\in R} (\neg x_a \vee \neg x_b) \\
\varphi_{adm}(F) &= \varphi_{cf}(F) \wedge \bigwedge_{(b,c)\in R} (\neg x_c \vee \bigvee_{(a,b)\in R} x_a) \\
\varphi_{com}(F) &= \varphi_{adm}(F) \wedge \bigwedge_{a\in A} (y_a \vee \bigvee_{(b,a)\in R} \neg y_b)
\end{aligned}$$

7.2.2. Preferred Semantics

For preferred semantics we are only interested in skeptical reasoning, since credulous reasoning for preferred semantics has a lower complexity.

Given an AF $F = (A, R)$ and an argument $\alpha \in A$ we can choose the base semantics from $\{adm, com\}$. The SHORTCUTS function for preferred semantics is shown in pseudocode in Algorithm 5. It implements the shortcut from Algorithm 1 (lines 2-4) and additionally exploits learning in case the shortcut does not decide the instance.

Algorithm 5 SHORTCUTS$_{prf}(F, \alpha, \mathsf{Skept})$

```
Require: AF $F = (A, R)$, argument $\alpha \in A$ and base-semantics $\sigma \in \{adm, com\}$
Ensure: rejects the input only if $\alpha$ is not skeptically accepted in $F$ w.r.t. $prf$,
        returns a formula otherwise
1: if SAT ($\varphi_\sigma(F) \wedge \big(\bigvee_{(\beta,\alpha)\in R} x_\beta\big)$) then
2:     reject
3: else
4:     return $\bigwedge_{(\beta,\alpha)\in R} \neg x_\beta$
5: end if
```

Here we simply check in line 1 whether there is a counter-example for skeptical acceptance of α under the chosen base-semantics witnessed by a set attacking α, and if not, learn in line 4 that this is the case. Furthermore, in the case of preferred semantics with a base semantics $\sigma \in \{adm, com\}$ the formulas $\psi^I_\theta(F)$ and γ^I_θ are:

$$\begin{aligned}
\psi^I_{prf}(F) &= \varphi_\sigma(F) \wedge \bigwedge_{x \in I \cap X} x \wedge (\bigvee_{x \in X \setminus I} x) \\
\gamma^I_{prf} &= \bigvee_{x \in X \setminus I} x.
\end{aligned}$$

Here, given a set I satisfying base semantics of the main procedure, ψ^I_{prf} encodes the supersets $I' \supset I$ satisfying σ. [3] By adding the formula γ^I_{prf} we learn that we are no longer interested in subsets of I, i.e. we need at least one argument from the complement, thereby reducing the search space.

[3] Note that in principle one can choose different base-semantics for the main procedure, the shortcut function and the formula $\psi^I_{prf}(F)$.

7.2.3. Semi-Stable and Stage Semantics

Let us next consider semantics $\theta \in \{sem, stg\}$. For semi-stable, we have the choice of using either adm or com as base-semantics. For stage semantics, we use base semantics cf. The SHORTCUTS function for semi-stable and stage semantics, based on $\mathsf{stablecons}^k_\sigma$, is presented in pseudocode in Algorithm 6.

The procedure follows the idea of the first loop in the Algorithms 2 and 3, and thus in principle can decide the reasoning problems without any further investigations, but would be expensive for high values of k. To profit in cases where k is small from the complexity-sensitive perspective, and avoid unnecessary computational cost otherwise, we include again the parameter d to bound the search-depth to sets of arguments to those whose *range* S satisfies $|A \setminus S| \leq d$.

The loop in line 7 iterates over sets $S \subseteq A$. In line 8, MAXIMAL returns a maximal-cardinality set S from U, (starting with $S = A$). Line 9 builds the formula f_S encoding that sets of interest have range S. Line 10 tests whether there is a set with range S satisfying q under the base semantics. If so, we have found an extension satisfying q and accept. Line 12 tests whether there is an extension under base semantics which has range S. If so, we learn that we are no longer interested in sets $S' \subseteq S$. Otherwise (line 16) we learn that we are not interested in extensions with range S. Finally, if all sets have been excluded from being the range of an extension satisfying q, the procedures rejects. In case the bound d is exceeded before this, we return the formula ψ as learnt information.

The helper formulas ψ^I_σ and γ^I_σ ($\theta \in \{sem, stg\}$) are:

$$\begin{aligned}
\psi^I_\theta(F) &= \varphi_\sigma(F) \wedge \bigwedge_{y \in I \cap Y} y \wedge \big(\bigvee_{y \in Y \setminus I} y \big) \\
\gamma^I_\theta &= \bigvee_{y \in Y \setminus I} y.
\end{aligned}$$

where $F = (A, R)$ is an AF, base-semantics $\sigma \in \{adm, com\}$ if $\theta = sem$ and $\sigma = cf$ in case $\theta = stg$. For a set I satisfying base-semantics σ the formula ψ^I_σ encodes sets I' satisfying σ and $I^+_R \subset (I')^+_R$, while formula γ^I_σ encodes that we are no longer interested in sets I' with $(I')^+_R \subseteq I^+_R$.

8. Experiments

We have implemented the SAT-based instantiations from Section 7 of our complexity-sensitive decision procedures as the SAT-based argumentation system

Algorithm 6 SHORTCUTS$_\theta(F, \alpha, M)$ (where $\theta \in \{sem, stg\}$)

```
Require: AF $F = (A, R)$, argument $\alpha \in A$,
         base-semantics $\sigma \in \{adm, com\}$ for $\theta = sem$; $\sigma = cf$ for $\theta = stg$;
         and reasoning mode $M \in \{\mathsf{Skept}, \mathsf{Cred}\}$;
Ensure: accepts (rejects) the input only if $M = \mathsf{Skept}$ and $\alpha$ is skeptically rejected (accepted) or $M = \mathsf{Cred}$ and $\alpha$ is credulously accepted (rejected) in $F$ w.r.t. $\sigma$, returns a formula otherwise
if $M = \mathsf{Skept}$ then
    $q \leftarrow \neg x_\alpha$
else if $M = \mathsf{Cred}$ then
    $q \leftarrow x_\alpha$
end if
$\psi \leftarrow \top; \mathcal{S} \leftarrow \{S \subseteq A \mid |A \setminus S| \leq d + 1\}$
while $(\exists S \in \mathcal{S} : |A \setminus S| \leq d)$ do
    $S \leftarrow \text{MAXIMAL}(U)$
    $f_S \leftarrow \bigwedge_{s \in S} y_s \wedge \bigwedge_{s \in A \setminus S} \neg y_s$
    if SAT $(\varphi_\sigma(F) \wedge q \wedge f_S(X))$ then
        accept
    else if SAT $(\varphi_\sigma(F) \wedge f_S(X))$ then
        $\psi \leftarrow \psi \wedge \left(\bigvee_{s \in A \setminus S} y_s\right)$
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S' \mid S' \subset S\}$
    else
        $\psi \leftarrow \psi \wedge \left(\bigvee_{s \in S} \neg y_S \vee \bigvee_{s \in A \setminus S} y_s\right)$
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S\}$
    end if
end while
if $(\mathcal{S} = \emptyset)$ then
    reject
else
    return $\psi$
end if
```

CEGARTIX[4]. In this section we present experimental results on the efficiency of CEGARTIX (v0.1a).

In the evaluation, we compare the performance of CEGARTIX to that of a recently proposed, state-of-the-art argumentation reasoning system [18] that exploits advances in answer-set programming (ASP). It turns out that CEGARTIX improves significantly over the state-of-the-art ASP-based system. Furthermore, we evaluate within CEGARTIX the impact of (i) applying the alternative base semantics and (ii) the impact of using different values of the parameter d within the shortcut functions for the different semantics.

CEGARTIX employs the CDCL SAT-solver Minisat [11] (v2.2.0) as the SAT oracle in an incremental mode, which allows us to retain from starting search from scratch when adding newly learnt information to a current satisfiable working formula.

In the experiments, we considered all the five reasoning tasks focused on in this article. The experiments were executed under OpenSUSE with 2.33-GHz Intel Xeon processors and 49-GB main memory. At most one query was run at a time to reduce performance bottlenecks. A timeout of five minutes was enforced on each individual run. We note that this high amount of memory is not actually used by the considered algorithms. We executed the algorithms with the hardest queries identified in our performance tests, see Section 8.2 for more details, with an additional hard limit of 4 GB, which was never reached.

8.1. Benchmarks

As benchmarks we generated AFs ranging from 60 to 1000 arguments using two parameterized methods for randomly generating the attack relation, following the line of [18] for benchmarking. We note that, as identified in [29], there is still need for diverse benchmark libraries for AFs incorporating e.g. AFs from applications.

Random AFs generated by inserting for any pair of arguments (a, b) with $a \neq b$ the attack from a to b with a given probability p.

Grid AFs that are sub-graphs of an $n \times m$ grid-structure. We consider two different neighborhoods, one connecting arguments vertically and horizontally

[4] Available at `www.dbai.tuwien.ac.at/research/project/argumentation/cegartix/`.

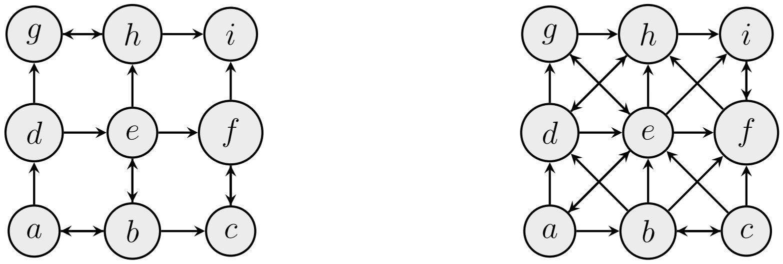


Figure 8: Examples of grid-structured AFs

and one that additionally connects the arguments diagonally. Such a connection is a mutual attack with a given probability p and an attack in only one direction otherwise. In the last case the direction is chosen with 0.5 probability. See Figure 8 for examples.

For both methods, we used the values $p \in \{0.1, 0.2, 0.3, 0.4\}$ and in addition for the grid AFs $n \in \{5, 10, 15\}$. The parameter m can then be calculated from the total number of arguments in the framework. For the number of arguments, we distinguished between medium-sized AFs with 60, 70, 80, 90, 100, 110, 130, 160 and 200 arguments as well as larger AFs with 300, 400, 500, 600, 700, 800, 900 and 1000 arguments. For the former we generated random and grid attack relations, for the latter only grid AFs. Overall this resulted in 360 random AFs, 1080 medium-sized grid AFs as well as 960 large grid AFs for each neighborhood. The total number is 4440 generated AFs.

We let the solvers compute queries for all semantics in this paper, that is, for the semantics prf, sem and stg, using skeptical reasoning with all three, and additionally credulous reasoning with the latter two. For the random instances three different arguments were queried, while for the grid instances we used five arguments. This resulted in a total of 107400 benchmark instances. We would like to stress that the generated AFs are by no means tailored to the fragments our approach is based on. Lastly note also that the number of attacks scales linearly with the number of arguments for grid AFs, while it scales quadratically with the number of arguments for random AFs.

8.2. Comparison with State-of-the-Art

We compare the performance of CEGARTIX to that of a recently proposed, state-of-the-art argumentation reasoning system [18] that exploits advances in answer-set programming (ASP) via the so-called `metasp` approach[5]. This system is a further improvement of the ASPARTIX system [17]. For comparing CEGARTIX with the ASP-based approach, we used the state-of-the-art no-good learning disjunctive ASP solver claspD [31] (v1.1.2) combined with the grounder gringo [32] (v3.0.4).

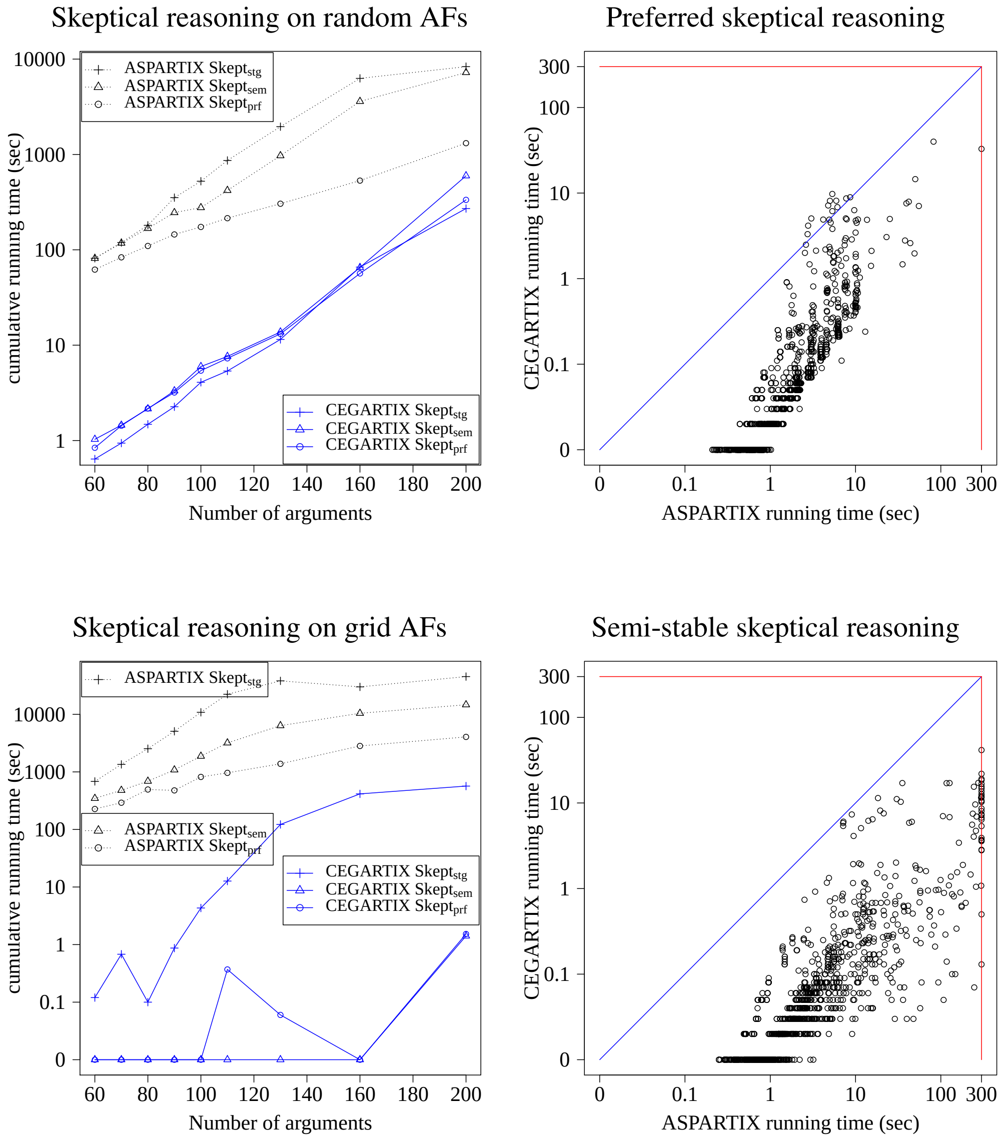
For this comparison, we employed the base semantics $\textsc{base-sem}(\sigma) = com$ for $\sigma \in \{prf, sem\}$, and $\textsc{base-sem}(stg) = cf$ within CEGARTIX. The parameter for the $\textsc{Shortcuts}_\sigma(F, \alpha, M)$ function for $\sigma \in \{sem, stg\}$ was set to $d = 1$. The reason for setting the shortcut parameter d to 1 was to allow for semi-stable and stage semantics a fast computation if a stable extension exists ($d = 0$ is sufficient for this) or if a semi-stable or stage extension with a range covering almost all arguments exists. On the other hand by setting d to 1 we keep the number of checks in the shortcut low.

We considered two metrics for comparison, namely, (cumulative) running time in seconds for all queries of a particular reasoning mode without timed out instances, and separately the number of timeouts. This has the effect that the figures depicting cumulative running times exclude timed out runs, and therefore show a faster performance if the number of timed out queries was high. For performance comparison of CEGARTIX with ASPARTIX, using the above mentioned `metasp` approach, we considered only medium-sized AFs.

Figures 9 and 10 present results on comparing ASPARTIX and CEGARTIX. On the left, the line plots comparing the cumulative running times (using logarithmic scale) are shown. On the right, the scatter plots present running time differences of individual queries, *including* timed out instances. The dotted lines denote ASPARTIX and the solid lines show the performance for CEGARTIX. For the plots on the left side, the running times for queries made on AFs with a particular number of arguments are grouped together. The number of timeouts are shown in Table 5 for ASPARTIX. Using CEGARTIX with the parameters noted above, no timeouts were encountered.

First consider Figure 9 for skeptical reasoning with random AFs (upper left plot). CEGARTIX behaves very similarly for all three semantics with respect to the considered metric, and outperforms ASPARTIX. The difference of ASPARTIX and CEGARTIX is more distinct with semi-stable and stage semantics and less so for preferred semantics. Note that for AFs with 200 arguments, 29 and 44

[5] Skeptical and credulous reasoning in `metasp` is done by introducing constraints in the so-called meta answer-set programs; for details, see [30].



Skeptical reasoning on random AFs

Preferred skeptical reasoning

Skeptical reasoning on grid AFs

Semi-stable skeptical reasoning

Figure 9: Comparison of ASPARTIX and CEGARTIX: Cumulative running times using logarithmic scale (left), scatter plots (right).

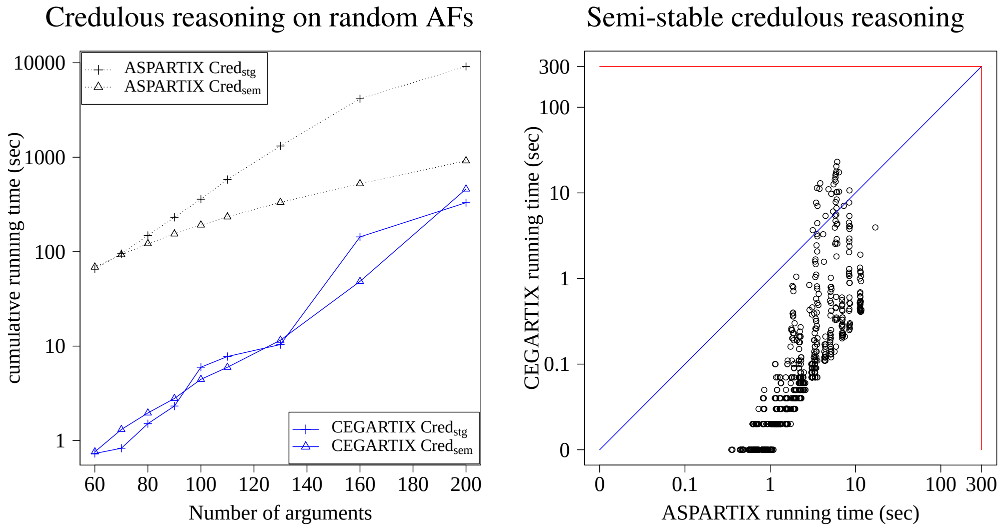


Figure 10: Comparison of ASPARTIX and CEGARTIX: Cumulative running times using logarithmic scale (left), a scatter plot (right).

timeouts were encountered for ASPARTIX under skeptical semi-stable and stage reasoning, respectively.

The grid structured AFs typically performed quite differently than random AFs, which is why we investigate them separately in Figure 9 (lower left plot). In many cases the running times were in fact very close to 0. One can see that grid AFs are mostly trivial for CEGARTIX. CEGARTIX could solve skeptical reasoning tasks with preferred and semi-stable semantics combined within 10 seconds, while ASPARTIX took significantly more time. The stage semantics is the only one here with higher running times for CEGARTIX. We will consider the larger grid AFs below for a more detailed study of CEGARTIX.

Credulous reasoning shows in general a similar picture to skeptical, with the exception that for semi-stable reasoning on random AFs, ASPARTIX and CEGARTIX appear to be closer with respect to performance, which can be seen in Figure 10 (on the left). However, a closer look at the encountered timed out instances reveals that CEGARTIX significantly outperforms ASPARTIX also in this case.

For deeper understanding, we also looked into scatter plots comparing ASPARTIX and CEGARTIX. These depict running time comparisons for individual instances for both solvers, including timed out instances. The scatter plots in Figures 9 and 10 are for skeptical preferred, semi-stable reasoning, and credulous

Table 5: Timeouts encountered with ASPARTIX on medium-sized random/grid AFs.

Arguments	$Skept_{prf}$	$Cred_{sem}$	$Skept_{sem}$	$Cred_{stg}$	$Skept_{stg}$
100	$0/1$	$0/0$	$0/0$	$0/0$	$0/3$
110	$0/2$	$0/1$	$0/6$	$0/7$	$0/11$
130	$0/2$	$0/1$	$0/4$	$0/77$	$0/120$
160	$0/4$	$0/25$	$0/36$	$0/349$	$1/411$
200	$1/18$	$0/85$	29/89	$22/497$	$44/562$

semi-stable acceptance. The x-axis shows the running time of individual queries with ASPARTIX and the y-axis with CEGARTIX on the same instance. Due to the low running times of CEGARTIX on grid AFs, we only considered random AFs for scatter plots.

For preferred semantics, many queries are in favor of CEGARTIX. Except for a few instances, which are drastically faster for CEGARTIX, the difference is usually within a few seconds, however. Semi-stable semantics yields a different picture, depending on the reasoning mode. For skeptical queries CEGARTIX clearly outperforms ASPARTIX on basically all instances. Credulous reasoning overall also is solved faster by CEGARTIX. A number of queries, however, were computed more efficiently by ASPARTIX. We omit the scatter plots for stage reasoning as they show a behavior similar to skeptical semi-stable semantics.

In general stage and semi-stable semantics show a similar behavior, which reflects their similar nature as well as the similar procedures used for the evaluation of the semantics. It seems that, when considering range-maximality, the choice of the base-semantics (cf or com) has only minor effects on the qualitative behavior of the semantics. We will see a similar behavior in the next section, when comparing adm and com as base-semantics for semi-stable.

Finally we made some experiments regarding the memory usage of the algorithms. We let both CEGARTIX and ASPARTIX compute the skeptical acceptance under stage semantics on AFs with 200 arguments (these are the hardest instances used in the above comparison) and additionally enforced a hard limit of 4 GB of memory, which was never reached.

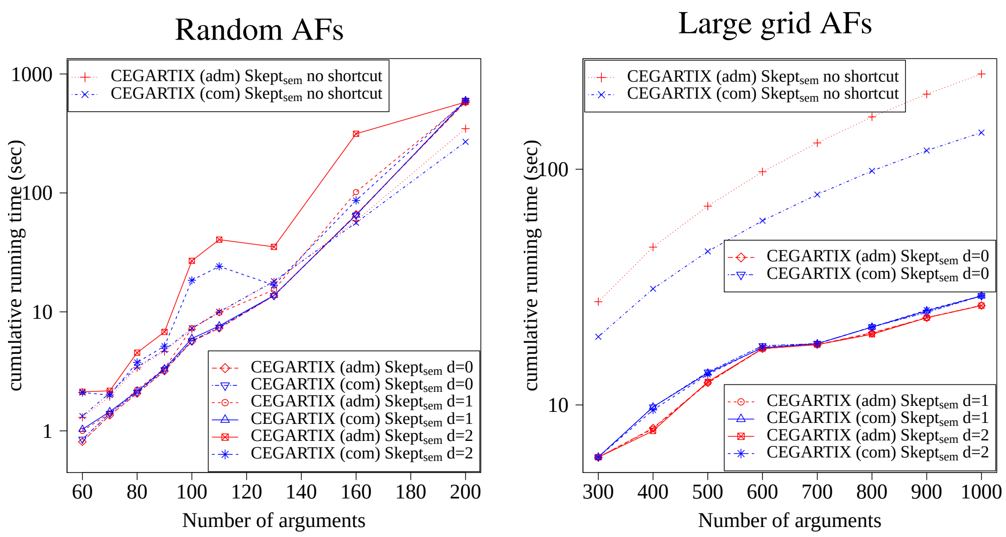


Figure 11: The effect of different parameter settings for the shortcut function and the choice of base semantics for skeptical semi-stable reasoning.

8.3. Impact of Base Semantics and Shortcuts within CEGARTIX

In this section we investigate the choice of the base semantics for solving the reasoning problems, as well as the effect of the shortcuts for semi-stable semantics, within CEGARTIX. We will again distinguish between random AFs and grid AFs, since CEGARTIX behaves quite differently on these two classes of AFs.

First we consider results regarding the impact of the base semantics. On random AFs there appears to be only a minimal effect for most CEGARTIX parameters and reasoning tasks. The cumulative running times were comparable between admissible and complete base semantics. Figure 11 shows the performance resulting from applying different combinations of (i) parameter settings for the shortcut $d = 0, 1, 2$ and (ii) the alternative base semantics for semi-stable semantics and skeptical reasoning. The complete base semantics is slightly outperforming the admissible base for $d = 2$, for example. For credulous reasoning the difference between the choice of the base semantics was even smaller. The results for preferred semantics were similar.

The choice of the base semantics, however, has a stronger influence for reasoning under the preferred semantics on grid AFs. For certain queries the number of oracle calls rises tremendously using the admissible base semantics. This in turn can be seen in the cumulative running times in Figure 12 for large grid AFs. In

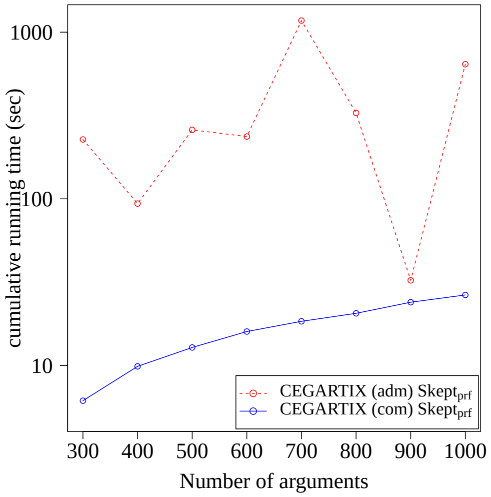


Figure 12: Performance comparison for the alternative choices of the base semantics for preferred skeptical reasoning on large grid AFs.

contrast, only a few SAT-solver calls are made when using the complete base semantics. We encountered a higher number of SAT-calls in 130 out of 9600 queries for preferred semantics on large grid AFs with admissibility base semantics. Algorithm 4 entered in these queries the outer while loop (line 7) more than 40 times, i.e. had to consider a new admissible set many times, while on the remaining 9470 queries this number was lower. In fact the average number the loop was entered in these 130 queries was 1179.01. This is also reflected in the running time. From these 130 queries we have that 75 had a running time of five up to 277.08 seconds. These make up the largest proportion of the cumulative running time in Figure 12. Additionally we encountered 305 timeouts using admissibility base semantics for preferred reasoning and large grid AFs. For complete base semantics either using the shortcut was enough, or otherwise the query was decided without entering the outer loop in all large grid AFs, i.e. indicating that there did not exist a complete extension without the queried argument.

Regarding the number of SAT-calls for semi-stable semantics and skeptical reasoning, out of the 1080 queries on random AFs, using the shortcut with $d = 0$ the algorithm could solve 996 within the shortcut and the remaining with one application of the outer while loop (line 7) of Algorithm 4, for both base semantics. Since the number of SAT calls within the outer while loop is linearly bounded by the number of arguments in the AF, this means that the overall number of SAT-calls was low in these cases. Without the shortcuts we have that also the outer while loop was entered at most once for semi-stable skeptical reasoning.

For the performance of the shortcuts, we experienced a slight decrease of performance for random AFs using the shortcuts for semi-stable skeptical reasoning, see Figure 11 on the left. For the grid AFs we again have a different picture. Here, as can be seen in Figure 11 on the right, the shortcuts decrease the overall running time, due to the fact that for grid AFs with a neighborhood of at most four arguments we have no odd-cycles in the AFs, which in turn means that computing the stable extensions is sufficient. This is reflected in the running time using the shortcuts for semi-stable semantics. Here it is even sufficient to consider depth $d = 0$.

9. Related Work

To the best of our knowledge, our complexity-sensitive approach to developing decision procedures for second-level abstract argumentation problems is novel. The CEGAR approach has been harnessed for solving various other intrinsically hard reasoning problems [33, 34, 35, 36, 37, 38, 39, 40, 41]. However, we are not aware of earlier work on developing complexity-sensitive CEGAR-based procedures.

Systems developed for solving second-level AF reasoning problems typically rely on monolithic encodings in other reasoning problems of similar complexity, see e.g. [42, 43] for encodings in terms of quantified Boolean formulas or [17] for an answer-set programming based approach. Approaches to solving ”easier”, i.e., first-level, AF problems are mostly monolithic; these include SAT-encodings [16] as well as CSP-encodings [44, 45].

A noteworthy exception is the family of labelling-based algorithms [46, 47, 48, 49, 50]. Typically such approaches deal with the problem of enumerating all extensions and thus, by the exponential number of extensions, systems are restricted to rather small AFs. The main idea of labelling-based algorithms is to distinguish different statuses of arguments, e.g. an argument can be in the extension, attacked by an argument in the extension, or neither. These statuses are encoded via different labels. Now if one argument is labeled with a certain label, this has immediate effects on the possible labels for its neighbors. On a high-level there are two different approaches to exploiting these effects in computing extensions. First, starting with all arguments being in the extension and resolving the conflicts by removing arguments from the extension and update the labels correspondingly (see e.g. [47]). A second family of algorithms which have similarities with our approach consist of algorithms which start from a small set, test whether it is preferred, and, if not, extend the set by further arguments to finally reach a preferred extension (see e.g. [49, 50]). In [49] also optimizations for credulous and skeptical reasoning with preferred semantics are considered. A recent empirical comparison of different labelling-based approaches for enumerating preferred extensions can be found in [50]. The main difference between existing work on labelling-based algorithms and our procedures are as follows: (i) we delegate the combinatorial hard problems to a SAT-solver (an NP-oracle respectively), (ii) we do not have to deal with sets which are not admissible at all (but may become admissible by adding further arguments), and (iii) our procedures come with a (parametrized) complexity classification.

Other current branches in abstract argumentation are certain preprocessing techniques [51, 52, 53] to divide the problem into small pieces as well as implementations of fixed-parameter tractable algorithms [7, 6, 54]. While the concept of complexity-sensitivity as used in our approach bears some resemblance to fixed-parameter tractability and parameterized complexity, there are fundamental differences. Especially, while parameterized complexity usually studies parameterizations with respect to polynomial-time tractability, the CEGAR approach in general is typically based on exploiting "NP-tractability" via iterative calls to efficient NP decision procedures. Hence our work takes further steps towards a full classification for argumentation problems of high complexity. This also distinguishes our complexity results to the ones obtained in [8], where AFs with a small (graph-based) distance to tractable graph classes are studied. Moreover, the above-mentioned works on fixed-parameter tractability and splitting focus on the graph structure of an AF while in this work we also exploit structure in the set of the extensions.

Finally there is recent work on average-case algorithms for value-based argumentation [55], using an elaborate handling of the possible value orderings. However, this is a different setting and [55] focuses on the problem of handling the different possible value-orderings which do not occur in our setting.

This article extends work presented at the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR 2012) [56], with the following improvements. First of all, we now provide full proofs of the results, and additional explanations and examples. In Section 6 we describe our complexity-sensitive procedures in a generic setting, while [56] was restricted to the SAT-based instantiation. The experiments with the CEGARTIX system have been extended in several directions. First we now include more details on the performance tests showing e.g. scatter plots and also incorporating more details for the stage semantics. We additionally conducted experiments with larger grid-structured AFs (up to 1000 arguments) and investigated in detail the choice of the base-semantics and shortcuts. We used newer versions of the systems we compared to, and re-ran all experiments.

10. Conclusion

In this work, we developed a novel method for solving hard problems in the area of argumentation in a "complexity-sensitive" way. From one perspective, the approach can be seen as an argumentation-customized incarnation of the counter-example guided abstraction refinement approach originating from the field of model checking. Our prototype implementation CEGARTIX employs SAT-solvers as underlying inference engines. Experiments show that CEGARTIX significantly outperforms existing systems developed for hard argumentation problems (i.e. problems under the preferred, semi-stable, or stage semantics). The fundamental aspects of our approach are generic, allowing in principle to exploit as the underlying NP-oracle systems developed for other reasoning problems such as CSP or ASP, or even native argumentation systems for "easier" semantics such as the stable or complete semantics. Building necessary ground for the complexity-sensitive approach, we also presented an extensive complexity theoretic analysis, providing new results for certain fragments of argumentation frameworks, as well as distance-based complexity analysis, complementing recent results from [8].

The promising experimental results suggest to apply our approach also to formalisms extending the Dung-style frameworks which we focused on here. In particular, abstract dialectical frameworks [57] would be an appealing target formalism since it generalizes other proposals such as bipolar [58] and extended argumentation frameworks [59]. In the opposite direction, one could consider further fragments of Dung-style frameworks. For preferred semantics, an interesting class are AFs having a bound number of odd cycles; the complexity of evaluating such AFs is currently open. An interesting research direction is to develop complexity-sensitive approaches for this fragment.

Acknowledgments

This work has been funded by Vienna Science and Technology Fund (WWTF) through project ICT08-028, by the Austrian Science Fund (FWF) through projects

I1102 and P25518-N23 and by Academy of Finland through grants 132812 and 251170.

The authors want to thank the anonymous referees of the current paper and of the preceding KR 2012 paper for their comments which helped to improve this article. The authors are also grateful to Paul E. Dunne, for discussing his randomized complexity results in [26].

[1] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence 77 (2) (1995) 321–358.

[2] P. E. Dunne, T. J. M. Bench-Capon, Coherence in finite argument systems, Artificial Intelligence 141 (1/2) (2002) 187–203.

[3] W. Dvořák, S. Woltran, Complexity of semi-stable and stage semantics in argumentation frameworks, Information Processing Letters 110 (11) (2010) 425–430.

[4] S. Coste-Marquis, C. Devred, P. Marquis, Symmetric argumentation frameworks, in: Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005), Vol. 3571 of Lecture Notes in Computer Science, Springer, 2005, pp. 317–328.

[5] P. E. Dunne, Computational properties of argument systems satisfying graph-theoretic constraints, Artificial Intelligence 171 (10–15) (2007) 701–729.

[6] W. Dvořák, S. Szeider, S. Woltran, Reasoning in argumentation frameworks of bounded clique-width, in: Proceedings of the 3rd International Conference on Computational Models of Argument (COMMA 2010), Vol. 216 of Frontiers in Artificial Intelligence and Applications, IOS Press, 2010, pp. 219–230.

[7] W. Dvořák, R. Pichler, S. Woltran, Towards fixed-parameter tractable algorithms for abstract argumentation, Artificial Intelligence 186 (2012) 1–37.

[8] W. Dvořák, S. Ordyniak, S. Szeider, Augmenting tractable fragments of abstract argumentation, Artificial Intelligence 186 (2012) 157–173.

[9] M. Truszczynski, Trichotomy and dichotomy results on the complexity of reasoning with disjunctive logic programs, Theory and Practice of Logic Programming 11 (6) (2011) 881–904.

[10] J. P. Marques-Silva, K. A. Sakallah, GRASP: A search algorithm for propositional satisfiability, IEEE Transactions on Computers 48 (5) (1999) 506–521.

[11] N. Eén, N. Sörensson, An extensible SAT-solver, in: Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT 2003), Vol. 2919 of Lecture Notes in Computer Science, Springer, 2004, pp. 502–518.

[12] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu, H. Veith, Counterexample-guided abstraction refinement for symbolic model checking, Journal of the ACM 50 (5) (2003) 752–794.

[13] E. M. Clarke, A. Gupta, O. Strichman, SAT-based counterexample-guided abstraction refinement, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 23 (7) (2004) 1113–1123.

[14] T. Eiter, G. Gottlob, On the computational cost of disjunctive logic programming: Propositional case, Annals of Mathematics and Artificial Intelligence 15 (3-4) (1995) 289–323.

[15] Y. Lierler, cmodels - SAT-based disjunctive answer set solver, in: Proceedings of the 8th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2005), Vol. 3662 of Lecture Notes in Computer Science, Springer, 2005, pp. 447–451.

[16] P. Besnard, S. Doutre, Checking the acceptability of a set of arguments, in: Proceedings of the 10th International Workshop on Non-Monotonic Reasoning (NMR 2004), 2004, pp. 59–64.

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

[18] W. Dvořák, S. A. Gaggl, J. P. Wallner, S. Woltran, Making use of advances in answer-set programming for abstract argumentation systems, in:

Post-Conference Proceedings of the 19th International Conference on Applications of Declarative Programming and Knowledge Management, INAP 2011, and the 25th Workshop on Logic Programming, WLP 2011, Lecture Notes in Computer Science, vol. 7773, Springer, 2013, in press. Preliminary version: `http://arxiv.org/abs/1108.4942`.

[19] P. Baroni, M. Giacomin, Semantics of abstract argument systems, in: I. Rahwan, G. Simari (Eds.), Argumentation in Artificial Intelligence, Springer, 2009, pp. 25–44.

[20] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, The Knowledge Engineering Review 26 (4) (2011) 365–410.

[21] Y. Dimopoulos, A. Torres, Graph theoretical structures in logic programs and default theories, Theoretical Computer Science 170 (1-2) (1996) 209–244.

[22] P. E. Dunne, M. Caminada, Computational complexity of semi-stable semantics in abstract argumentation frameworks, in: Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA 2008), Vol. 5293 of Lecture Notes in Computer Science, Springer, 2008, pp. 153–165.

[23] W. Dvořák, S. Woltran, On the intertranslatability of argumentation semantics, Journal of Artificial Intelligence Research 41 (2011) 445–475.

[24] J. Bang-Jensen, G. Gutin, Digraphs: Theory, Algorithms and Applications, Springer Monographs in Mathematics, Springer, 2010.

[25] P. E. Dunne, M. Wooldridge, Complexity of abstract argumentation, in: G. Simari, I. Rahwan (Eds.), Argumentation in Artificial Intelligence, Springer, 2009, pp. 85–104.

[26] P. E. Dunne, The computational complexity of ideal semantics, Artificial Intelligence 173 (18) (2009) 1559–1591.

[27] L. G. Valiant, V. V. Vazirani, NP is as easy as detecting unique solutions, Theoretical Computer Science 47 (3) (1986) 85–93.

[28] M. Caminada, D. Gabbay, A logical account of formal argumentation, Studia Logica 93 (2–3) (2009) 109–145.

[29] W. Dvořák, S. A. Gaggl, S. Szeider, S. Woltran, Benchmark libraries for argumentation, in: Agreement Technologies, Vol. 8 of LGTS, Springer, 2013, Ch. The Added Value of Argumentation, pp. 389–393.

[30] M. Gebser, R. Kaminski, T. Schaub, Complex optimization in answer set programming, Theory and Practice of Logic Programming 11 (4-5) (2011) 821–839.

[31] C. Drescher, M. Gebser, T. Grote, B. Kaufmann, A. König, M. Ostrowski, T. Schaub, Conflict-driven disjunctive answer set solving, in: Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008), AAAI Press, 2008, pp. 422–432.

[32] M. Gebser, R. Kaminski, A. König, T. Schaub, Advances in *gringo* series 3, in: Proceedings of the 11th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2011), Vol. 6645 of Lecture Notes in Computer Science, Springer, 2011, pp. 345–351.

[33] R. Sebastiani, Lazy satisfiability modulo theories, Journal of Satisfiability, Boolean Modeling and Computation 3 (3-4) (2007) 141–224.

[34] L. de Moura, H. Ruess, M. Sorea, Lazy theorem proving for bounded model checking over infinite domains, in: Proceedings of the 18th International Conference on Automated Deduction (CADE-18), Vol. 2392 of Lecture Notes in Computer Science, Springer, 2002, pp. 438–455.

[35] C. W. Barrett, D. L. Dill, A. Stump, Checking satisfiability of first-order formulas by incremental translation to SAT, in: Proceedings of the 14th International Conference on Computer Aided Verification (CAV 2002), Vol. 2404 of Lecture Notes in Computer Science, Springer, 2002, pp. 236–249.

[36] C. Flanagan, R. Joshi, X. Ou, J. B. Saxe, Theorem proving using lazy proof explication, in: Proceedings of the 15th International Conference on Computer Aided Verification (CAV 2003), Vol. 2725 of Lecture Notes in Computer Science, Springer, 2003, pp. 355–367.

[37] C. M. Wintersteiger, Y. Hamadi, L. de Moura, Efficiently solving quantified bit-vector formulas, in: Proceedings of the 10th International Conference on Formal Methods in Computer-Aided Design (FMCAD 2010), IEEE, 2010, pp. 239–246.

[38] D. Monniaux, Quantifier elimination by lazy model enumeration, in: Proceedings of the 22nd International Conference on Computer Aided Verification (CAV 2010), Vol. 6174 of Lecture Notes in Computer Science, Springer, 2010, pp. 585–599.

[39] M. Janota, R. Grigore, J. Marques-Silva, Counterexample guided abstraction refinement algorithm for propositional circumscription, in: Proceedings of the 12th European Conference on Logics in Artificial Intelligence (JELIA 2010), Vol. 6341 of Lecture Notes in Computer Science, Springer, 2010, pp. 195–207.

[40] M. Janota, J. P. Marques-Silva, Abstraction-based algorithm for 2QBF, in: Proceedings of the 14th International Conference on Theory and Applications of Satisfiability Testing (SAT 2011), Vol. 6695 of Lecture Notes in Computer Science, Springer, 2011, pp. 230–244.

[41] M. Janota, W. Klieber, J. Marques-Silva, E. M. Clarke, Solving QBF with counterexample guided refinement, in: Proceedings of the 15th International Conference on Theory and Applications of Satisfiability Testing (SAT 2012), Vol. 7317 of Lecture Notes in Computer Science, Springer, 2012, pp. 114–128.

[42] O. Arieli, M. W. A. Caminada, A QBF-based formalization of abstract argumentation semantics, J. Applied Logic 11 (2) (2013) 229–252.

[43] U. Egly, S. Woltran, Reasoning in argumentation frameworks using quantified boolean formulas, in: Proceedings of the 1st International Conference on Computational Models of Argument (COMMA 2006), Vol. 144 of Frontiers in Artificial Intelligence and Applications, IOS Press, 2006, pp. 133–144.

[44] L. Amgoud, C. Devred, Argumentation frameworks as constraint satisfaction problems, in: Proceedings of the 5th International Conference on Scalable Uncertainty Management (SUM 2011), Vol. 6929 of Lecture Notes in Computer Science, Springer, 2011, pp. 110–122.

[45] S. Bistarelli, P. Campli, F. Santini, Finding partitions of arguments with Dung's properties via SCSPs, in: Proceedings of the 2011 ACM Symposium on Applied Computing (SAC 2011), ACM, 2011, pp. 913–919.

[46] B. Verheij, A labeling approach to the computation of credulous acceptance in argumentation, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), 2007, pp. 623–628.

[47] S. Modgil, M. Caminada, Proof theories and algorithms for abstract argumentation frameworks, in: Argumentation in Artificial Intelligence, Springer, 2009, pp. 105–132.

[48] M. Podlaszewski, M. Caminada, G. Pigozzi, An implementation of basic argumentation components, in: Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), IFAAMAS, 2011, pp. 1307–1308.

[49] S. Doutre, J. Mengin, Preferred extensions of argumentation frameworks: Query answering and computation, in: R. Goré, A. Leitsch, T. Nipkow (Eds.), Automated Reasoning, First International Joint Conference, IJCAR 2001, Siena, Italy, June 18-23, 2001, Proceedings, Vol. 2083 of Lecture Notes in Computer Science, Springer, 2001, pp. 272–288.

[50] S. Nofal, P. E. Dunne, K. Atkinson, On preferred extension enumeration in abstract argumentation, in: Proceedings of the 4th International Conference on Computational Models of Argument (COMMA 2012), Vol. 245 of Frontiers in Artificial Intelligence and Applications, IOS Press, 2012, pp. 205–216.

[51] R. Baumann, Splitting an argumentation framework, in: Proceedings of the 11th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2011), Vol. 6645 of Lecture Notes in Computer Science, Springer, 2011, pp. 40–53.

[52] R. Baumann, G. Brewka, R. Wong, Splitting argumentation frameworks: An empirical evaluation, in: Revised Selected Papers of the 1st International Workshop on Theories and Applications of Formal Argumentation (TAFA 2011), Vol. 7132 of Lecture Notes in Computer Science, Springer, 2012, pp. 17–31.

[53] R. Baumann, G. Brewka, W. Dvořák, S. Woltran, Parameterized splitting: A simple modification-based approach, in: E. Erdem, J. Lee, Y. Lierler, D. Pearce (Eds.), Correct Reasoning - Essays on Logic-Based AI in Honour of Vladimir Lifschitz, Vol. 7265 of Lecture Notes in Computer Science, Springer, 2012, pp. 57–71.

[54] W. Dvořák, M. Morak, C. Nopp, S. Woltran, A dynamic programming reasoner for abstract argumentation, in: Post-Conference Proceedings of the 19th International Conference on Applications of Declarative Programming and Knowledge Management, INAP 2011, and the 25th Workshop on Logic Programming, WLP 2011 Management (INAP 2011), Lecture Notes in Computer Science, vol. 7773, Springer, 2013, in press. Preliminary version: `http://arxiv.org/abs/1108.4804`.

[55] S. Nofal, P. E. Dunne, K. Atkinson, Towards average-case algorithms for abstract argumentation, in: Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART 2012), Volume 1 - Artificial Intelligence, SciTePress, 2012, pp. 225–230.

[56] W. Dvořák, M. Järvisalo, J. P. Wallner, S. Woltran, Complexity-sensitive decision procedures for abstract argumentation, in: Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR 2012), AAAI Press, 2012, pp. 54–64.

[57] G. Brewka, S. Woltran, Abstract dialectical frameworks, in: Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR 2010), AAAI Press, 2010, pp. 102–111.

[58] L. Amgoud, C. Cayrol, M.-C. Lagasquie, P. Livet, On bipolarity in argumentation frameworks, International Journal of Intelligent Systems 23 (2008) 1–32.

[59] S. Modgil, Reasoning about preferences in argumentation frameworks, Artificial Intelligence 173 (9-10) (2009) 901–934.