This manuscript was published as:
Tuomo Lehtonen, Johannes P. Wallner, and Matti Järvisalo. From Structured to Abstract Argumentation: Assumption-Based Acceptance via AF Reasoning. In Alessandro Antonucci, Laurence Cholvy and Odile Papini, editors, *Proceedings of the 14th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU 2017*, volume 10369 of *Lecture Notes in Artificial Intelligence*, pages 57–68, Lugano, Switzerland, July 2017. Springer. The final publication is available at Springer via `http://dx.doi.org/10.1007/978-3-319-61581-3_6`.

# From Structured to Abstract Argumentation: Assumption-Based Acceptance via AF Reasoning*

Tuomo Lehtonen[1], Johannes P. Wallner[2], and Matti Järvisalo[1]

[1] HIIT, Department of Computer Science, University of Helsinki, Finland
[2] Institute of Information Systems, TU Wien, Austria

**Abstract.** We study the applicability of abstract argumentation (AF) reasoners in efficiently answering acceptability queries over assumption-based argumentation (ABA) frameworks, one of the prevalent forms of structured argumentation. We provide a refined algorithm for translating ABA frameworks to AFs allowing the use of AF reasoning to answer ABA acceptability queries, covering credulous and skeptical acceptance problems over ABAs in a seamless way under several argumentation semantics. We empirically show that the approach is complementary with a state-of-the-art ABA reasoning system.

## 1 Introduction

Argumentation is today a vibrant area of modern AI research, providing formalisms for representing and reasoning about conflicting arguments, aiming at conflict resolution through detecting sets of non-conflicting arguments which together counter—or defend themselves against—all counterarguments.

Several argumentation formalisms have been proposed, with different desirable properties. Perhaps the simplest formalism for argumentation are abstract argumentation frameworks (AFs) [11]. AFs allow for representing conflicts—or attacks—between arguments as directed graphs, where nodes represent abstract arguments, and edges represent attacks. Several reasoning system implementations for AF reasoning exists today [18, 17, 5, 6, 24, 25], especially for central AF reasoning problems such as credulous and skeptical acceptance of arguments under various AF semantics.

Another central formalism is structured argumentation [1, 3, 22, 2, 26] in which, in contrast to abstract argumentation, the internal structure of arguments is made explicit through derivations from more basic structures. One well-known approach to structured argumentation is assumption-based argumentation (ABA) [3, 13, 29]. In ABA arguments are represented compactly as graph-based derivations [7] from a given rule-based deductive system over sentences, starting from assumptions. A central approach to reasoning about acceptability of arguments over ABAs are so-called dispute derivations [7, 12, 14, 20, 21, 28], implemented in various ABA reasoning systems [19–21, 14, 8, 28, 9, 7]. The abagraph system [7] supporting credulous reasoning over ABAs under the admissible and grounded semantics represents the current state of the art.

* Work funded by Academy of Finland, grants 251170 COIN, 276412, and 284591; Research Funds of the University of Helsinki; and the Austrian Science Fund (FWF): I2854 and P30168.

While systems for reasoning over AFs and ABAs have been developed, the applicability of state-of-the-art abstract argumentation reasoners for reasoning about assumption-based argumentation frameworks has received little attention. To bridge this gap, we study the applicability of state-of-the-art abstract argumentation reasoners in efficiently answering acceptability queries over ABA frameworks. While theoretical work on mapping ABAs to AFs exists [14, 4], here we concretely implement an approach to reasoning about acceptance of sentences in assumption-based argumentation via translating ABA frameworks into abstract argumentation frameworks, and thereafter using AF reasoning to decide acceptance of sentences. While it would be desirable to exactly compute a small, yet sufficient, set of AF arguments for a given ABA, we show that restricting argument construction to only those arguments satisfying a minimality condition in their supports, which we call relevant arguments, is computationally very demanding: we prove that counting the number of such relevant arguments is #P-complete. To overcome this obstacle, we propose an algorithm for overapproximating the set of relevant arguments for a given ABA framework. We implement the reasoning part by answer set programming (ASP) encodings specifically suited for the types of AFs the translation gives rise to. We show that a prototype implementation of the approach is complementary in terms of performance with the state-of-the-art abagraph system for credulous acceptance in ABA. Our approach is generic in that it covers both credulous and skeptical acceptance problems under several central argumentation semantics over ABAs in a seamless way. Proofs of the main theorems are available in the paper supplement online at `https://cs.helsinki.fi/group/coreo/ecsqaru17`.

## 2 Preliminaries

**Assumption-Based Argumentation** We recall definitions related to assumption-based argumentation (ABA) [3, 29], following [10]. We assume a deductive system $(\mathcal{L}, \mathcal{R})$ with $\mathcal{L}$ a formal language, i.e., a countable set of sentences, and $\mathcal{R}$ a set of inference rules over $\mathcal{L}$ with a rule $r \in \mathcal{R}$ having the form $a_0 \leftarrow a_1, \ldots, a_n$ with $a_i \in \mathcal{L}$. We denote the head of rule $r$ by $head(r) = \{a_0\}$ and the (possibly empty) body of $r$ by $body(r) = \{a_1, \ldots, a_n\}$. A sentence $a \in \mathcal{L}$ is derivable from a set $X \subseteq \mathcal{L}$ via rules $\mathcal{R}$, denoted by $X \vdash_{\mathcal{R}} a$, if there is a sequence of rules $(r_1, \ldots, r_n)$ s.t. $head(r_n) = a$ and for each rule $r_i$ it holds that $r_i \in \mathcal{R}$ and each sentence in the body of $r_i$ is derived from rules earlier in the sequence or in $X$, i.e., $body(r_i) \subseteq X \cup \bigcup_{j<i} head(r_j)$. The deductive closure for $X$ w.r.t. rules $\mathcal{R}$ is given by $Th_{\mathcal{R}}(X) = \{a \mid X \vdash_{\mathcal{R}} a\}$.

An ABA framework is a tuple $(\mathcal{L}, \mathcal{R}, \mathcal{A}, {}^-)$ with $(\mathcal{L}, \mathcal{R})$ a deductive system, a set of assumptions $\mathcal{A} \subseteq \mathcal{L}$, and a function $^-$ (contrary function) mapping assumptions $\mathcal{A}$ to sentences $\mathcal{L}$. We focus on flat ABA frameworks where assumptions cannot be derived. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, {}^-)$ be an ABA framework. A set of assumptions $\Delta \subseteq \mathcal{A}$ attacks an assumption $b \in \mathcal{A}$ in the ABA framework $D$ if the contrary of $b$ is derivable from $\Delta$ in $D$, i.e., $\overline{b} \in Th_{\mathcal{R}}(\Delta)$. Further, $\Delta$ attacks a set of assumptions $\Delta' \subseteq \mathcal{A}$ in the ABA framework $D$ if an assumption in $\Delta'$ is attacked by $\Delta$, i.e., $Th_{\mathcal{R}}(\Delta) \cap \{\overline{a} \mid a \in \Delta'\} \neq \emptyset$.

**Definition 1.** *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, {}^-)$ be an ABA framework. Further, let $\Delta \subseteq \mathcal{A}$ be a set of assumptions that does not attack itself in $D$. Set $\Delta$ is*

– admissible *in $D$ if each set of assumptions $\Delta'$ that attacks $\Delta$ is attacked by $\Delta$;*
– preferred *in $D$ if $\Delta$ is admissible and there is no admissible set of assumptions $\Delta'$ in $D$ with $\Delta \subset \Delta'$; and*
– stable *in $D$ if each $a \in \mathcal{A} \setminus \Delta$ is attacked by $\Delta$.*

We use the term $\sigma$-assumption-set to refer to an assumption set under a specific semantics $\sigma \in \{adm, stb, prf\}$.[3] Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework and $\sigma$ a semantics. A sentence $s \in \mathcal{L}$ is credulously accepted in $D$ under semantics $\sigma$ if there is a $\sigma$-assumption-set $\Delta$ s.t. $s \in Th_{\mathcal{R}}(\Delta)$; and skeptically accepted in $D$ under semantics $\sigma$ if it holds that $s \in Th_{\mathcal{R}}(\Delta)$ for all $\sigma$-assumption-sets $\Delta$.

Complexity of reasoning of (flat) ABA frameworks [10] is shown in Fig. 1.

| | ABA | | AF | |
|---|---|---|---|---|
| semantics | cred | skept | cred | skept |
| admissible | NP-c | P-c | NP-c | trivial |
| stable | NP-c | coNP-c | NP-c | coNP-c |
| preferred | NP-c | $\Pi_2^p$-c | NP-c | $\Pi_2^p$-c |

**Fig. 1.** Complexity of reasoning.

*Example 1.* An ABA framework is shown in Fig. 2 (left) with $\mathcal{L} = \{a, b, c, d, e, f, g, h, i\}$, as well as the admissible, stable, and preferred assumption sets. Sentences $g$ and $h$ are credulously accepted under $\sigma \in \{adm, prf, stb\}$, since they can be derived from $\{a\}$ and $\{c\}$. Further, $i$ is skeptically accepted under $\sigma$, since $i$ is derivable from $\emptyset$.

**Abstract Argumentation Frameworks** An abstract argumentation framework (AF) [11] is a pair $F = (A, R)$, where $A$ is a finite non-empty set of arguments and $R \subseteq A \times A$ is the attack relation. The pair $(a, b) \in R$ indicates that $a$ attacks $b$. A set $S \subseteq A$ attacks an argument $b$ (in $F$) if there is an $a \in S$ s.t. $(a, b) \in R$. An argument $a \in A$ is *defended* (in $F$) by a set $S \subseteq A$ if, for each $b \in A$ such that $(b, a) \in R$, it holds that $S$ attacks $b$.

AF semantics are defined through functions $\sigma$ which assign to each AF $F = (A, R)$ a set $\sigma(F) \subseteq 2^A$ of extensions. We consider for $\sigma$ the functions $adm$, $stb$, and $prf$.

**Definition 2.** *Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is* conflict-free (in $F$) *if there are no $a, b \in S$ such that $(a, b) \in R$. We denote the collection of conflict-free sets of $F$ by $cf(F)$. For a conflict-free set $S \in cf(F)$ it holds that*

– *$S \in adm(F)$ iff each $a \in S$ is defended by $S$;*
– *$S \in prf(F)$ iff $S \in adm(F)$ and $\nexists S' \in adm(F)$ with $S \subset S'$; and*
– *$S \in stb(F)$ iff each $a \in A \setminus S$ is attacked by $S$.*

We use "$\sigma$-extension" to denote an extension under a semantics $\sigma$. Let $F = (A, R)$ be an AF. An argument $a \in A$ is credulously accepted in $F$ under $\sigma$ if there is an $E \in \sigma(F)$ s.t. $a \in E$. An argument $a$ is skeptically accepted in $F$ under $\sigma$ if $a$ is contained in every $E \in \sigma(F)$. For complexity of AF reasoning [15] see Fig. 1.

[3] We call, for reasons of uniformity and brevity, admissible sets a semantics; this is not meant to prescribe a particular logical stance to the frameworks.

| rules $\mathcal{R}$ | | contr. | ass. sets $\sigma$ | |
|---|---|---|---|---|
| $d \leftarrow a$ | $g \leftarrow e$ | $\overline{a} = h$ | $\emptyset$ | $adm$ |
| $e \leftarrow a, b$ | $d \leftarrow g$ | $\overline{b} = e$ | $\{a\}$ | $adm, prf, stb$ |
| $f \leftarrow c$ | $h \leftarrow f$ | $\overline{c} = d$ | $\{c\}$ | $adm$ |
| $e \leftarrow d$ | $i \leftarrow$ | | $\{b, c\}$ | $adm, prf, stb$ |

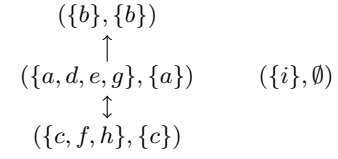



**Fig. 2.** Example ABA with $\mathcal{A} = \{a, b, c\}$ (left) and the corresponding AF (right).

## 3 From ABA to AF

The focus of this work is on studying the applicability of abstract argumentation reasoning tools for reasoning about acceptance of sentences in assumption-based argumentation frameworks. Given an ABA and a credulous/skeptical query as a sentence in the ABA, our approach to answer the query consists of the following two steps.

1. Translate the ABA framework into an AF in a way that the ABA query can be answered by applying AF reasoning principles on the resulting AF.
2. Adjust an AF reasoning system to answer the ABA query on the AF from step 1.

In this section we adapt translations of ABA frameworks [4, 14] to AFs to suit our goal of computational feasibility. The idea of the approach is to view subsets of the assumptions, and sentences derived from these sets, as abstract arguments. The assumptions of such an argument are called support of the argument. A key point for this translation, to ensure correctness, is to construct arguments so that all assumption sets are sufficiently covered, not missing crucial parts of the ABA framework. Sentences contained in an argument in a $\sigma$-extension of the resulting AF will be derivable in a $\sigma$-assumption-set of the original ABA framework and vice versa, thereby aligning the corresponding reasoning tasks of ABA frameworks and AFs.

In order to make step 2 computationally feasible, care needs to be taken in order to ensure that the AF resulting from step 1 does not become restrictively large (in terms of the number of arguments) in order to enable reasoning on the AF. To this end, we consider constructing only those arguments, which we call *relevant arguments*, whose support is minimal, in the sense that there is a sentence derivable from the support, but the sentence is not derivable from any proper subset of the support. However, we will show that the complexity of computing (exactly) the set of relevant arguments is restrictive for practical purposes. Motivated by both the computational hardness result and the need for restriction of the number of arguments, we then, in the subsequent sections, propose an algorithm for over-approximating the set of relevant arguments of a given ABA, and detail an approach to step 2 via answer set programming.

Key to the translation of ABA frameworks to AFs are the arguments for the AF, which are viewed as pairs of a set of assumptions and sentences derived from the set of assumptions. With the aim of focusing on relevant arguments, we generalize and adapt the concept of support-minimality [7, Definition 4.11]. In [7] support-minimality is defined for arguments with a single claim (derivation for a single sentence).

**Definition 3.** *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework. We define the set of sets of assumptions $minsupp(D)$ by $\Delta \in minsupp(D)$ iff $\bigcup_{\Delta' \subset \Delta} Th_{\mathcal{R}}(\Delta') \subset Th_{\mathcal{R}}(\Delta)$.*

In words, a set of assumptions $\Delta$ is a minimal support if there is a sentence derivable from $\Delta$ via $\mathcal{R}$ but not from any proper subset $\Delta' \subset \Delta$. Relevant arguments are defined as pairs of a set of sentences and a minimal support.

**Definition 4.** *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework, $L \subseteq \mathcal{L}$, and $\Delta \subseteq \mathcal{A}$. A pair $(L, \Delta)$ is a relevant argument (for $D$) if the following two conditions hold: (i) $\Delta \in minsupp(D)$; and (ii) $L = Th_{\mathcal{R}}(\Delta) \setminus (\bigcup_{\Delta' \subset \Delta} Th_{\mathcal{R}}(\Delta'))$.*

In words, a pair $(L, \Delta)$ is a relevant argument for a given ABA if $\Delta$ is in $minsupp(D)$ (first item), and $L$ contains those sentences that are derivable from $\Delta$ but not any proper subset of $\Delta$ (second item).

*Example 2.* Consider the ABA framework from Example 1. The sets in $minsupp(D)$ are $\{a\}$, $\{b\}$, $\{c\}$, and $\emptyset$. The admissible assumption set $\{b, c\}$ is not in $minsupp(D)$ since all sentences derivable from $\{b, c\}$ are derivable from $\{b\}$ or $\{c\}$. For each set in $minsupp(D)$ there is a relevant argument, e.g., $(\{a, d, e, g\}, \{a\})$ is a relevant argument for the ABA framework and all sentences in $\{a, d, e, g\}$ can be derived from $a$, and all sentences derivable from $\{a\}$ but not $\emptyset$ are contained in the first component.

**Definition 5.** *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework. An AF $F = (A, R)$ corresponds to the ABA $D$ if the following two conditions hold. (i) $A$ is the set of relevant arguments for $D$; and (ii) $R = \{((L, \Delta), (L', \Delta')) \mid L \cap \{\overline{x} \mid x \in \Delta'\} \neq \emptyset\}$.*

Briefly put, a corresponding AF for a given ABA framework contains the relevant argument for each set of assumptions in $minsupp(D)$, i.e., $|A| = |minsupp(D)|$, and attacks based on the supports and the derived sentences. In Fig. 2, the corresponding AF (right) for the ABA framework (left) is shown. In the following formal result, that follows the spirit of [14, Theorem 2.2] and [4, Theorem 6], we show that we have a correspondence between the ABA framework and the corresponding AF in terms of the semantics, which allows for utilization of AF reasoners on the AF to answer ABA queries. We define $sentences(E) = \bigcup_{(L,\Delta)\in E} L$.

**Theorem 1.** *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework, $\Delta \subseteq \mathcal{A}$, and $\sigma \in \{adm, stb, prf\}$. For an AF $F = (A, R)$ that corresponds to $D$, and $E \subseteq A$, it holds that*

- *if $\Delta$ is a $\sigma$-assumption-set of $D$, then $E = \{(L, \Delta') \in A \mid \Delta' \subseteq \Delta\}$ is a $\sigma$-extension of $F$, and $Th_{\mathcal{R}}(\Delta) = sentences(E)$;*
- *if $E$ is a $\sigma$-extension of $F$, then $\Delta = \bigcup_{(L,\Delta')\in E} \Delta'$ is a $\sigma$-assumption-set of $D$, and $Th_{\mathcal{R}}(\Delta) \supseteq sentences(E)$ for $\sigma = adm$, and $Th_{\mathcal{R}}(\Delta) = sentences(E)$ for $\sigma \in \{stb, prf\}$.*

Based on this formal correspondence, we can answer credulous (skeptical) acceptability queries in an ABA framework as specified in the next corollary.

**Corollary 1.** *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework, $l \in L$, $\sigma = \{adm, stb, prf\}$, $\sigma' = \{stb, prf\}$, and AF $F = (A, R)$ the corresponding AF for $D$. It holds that*

- *$l$ is credulously accepted under $\sigma$ in $D$ iff there is a credulously accepted argument $(L,\Delta)$ under $\sigma$ in $F$ with $l \in L$;*
- *$l$ is skeptically accepted under $\sigma'$ in $D$ iff for each $\sigma'$-extension $E$ of $F$ it holds that $l \in sentences(E)$.*

Skeptical acceptance under admissible semantics for ABA frameworks is polynomial-time decidable (Table 1), while our focus here is on the NP-hard acceptance problems. Omitting a relevant argument in a corresponding AF can directly lead to incorrect results w.r.t. acceptance queries of the original ABA framework. For instance, considering the corresponding AF shown in Example 2, removal of any of the relevant arguments of this AF would lead to missing sentences in the AF which are credulously accepted under, e.g., admissible semantics in the original ABA framework.

## 4 Computing Relevant Arguments

The authors of [7] conjecture that computing minimal supports may be computationally costly. We provide a formal result backing up this conjecture: we show that counting the number of minimal supports for a given ABA framework is intractable, in fact $\#\mathsf{P}$-complete under subtractive reductions [16] often used for showing hardness for counting complexity classes. (The prototypical $\#\mathsf{P}$-complete problem is that of counting satisfying assignments of a Boolean formula.)

**Theorem 2.** *For a given ABA framework, counting the number of minimal supports is $\#\mathsf{P}$-complete under subtractive reductions.*

To overcome this obstacle, we give an algorithm that overapproximates the set of relevant arguments. The algorithm traverses the rules backwards towards the assumptions. The underlying data structure operated on is a directed graph with vertices being both heads and bodies of rules in the ABA. There is a directed edge from a body to a head if there is a corresponding rule, and from a head to a body if the former is contained in the latter. We filter out non-derivable sentences. If the rules are acyclic, we can straightforwardly backward chain from the sinks to create all needed arguments. For the general (i.e. possibly cyclic) case, the presented algorithm also takes all heads of rules that are in non-trivial strongly connected components (SCCs), i.e., non-singleton SCCs, denoted by $SCC(D)$, as starting points. We store (partial) arguments with a set of sets of sentences, $Arg(X) = \{S_1, \ldots, S_n\}$ for a head or body $X$, indicating that $X$ is derivable from any $S_1, \ldots, S_n$.

**Algorithm 1** Argument Construction

**Require:** ABA $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$
1: Compute $SCC(D)$ //non-trivial SCCs
2: $S = sinks(G) \cup (\bigcup SCC(D) \cap \mathcal{L})$
3: **while** $S \neq \emptyset$ **do**
4: remove $s$ from $S$
5: PROCESS-HEAD$(s, \emptyset)$
6: mark $s$ visited

The main Algorithm 1 computes non-trivial SCCs, stores starting points in $S$, and recurses in the while loop (call by reference) with a picked sentence and the current derivation path $P$ (for detecting cyclic derivations; initialized with $\emptyset$). After processing, the sentence is marked, indicating that all derivations have been exhausted. Algorithm 2, PROCESS-HEAD, marks the head $s$ if not in a non-trivial SCC and adds $s$ to the derivation path $P$. In case $s$ is an assumption (or $\top$ for sentences derived from $\emptyset$), we add a new argument $\{s\}$ for $s$. Otherwise call PROCESS-BODY$(B, P)$ for each non-visited body $B$ from which $s$ can be derived, excluding $P$. Afterwards, we extend arguments for the bodies and add these arguments to $Arg(s)$.

Algorithm 3 takes care of bodies. We mark $B$ if it is not a non-trivial SCC or when each element in the body is either marked or not in non-trivial SCC. To avoid cyclic derivations we check if an element in the body is contained in path $P$. If not, after adding body $B$ to $P$, we call PROCESS-HEAD for each non-visited element. We collect all possible ways of deriving body $B$ by taking all minimal combinations of arguments from which to directly derive each $s \in B$ (SUBDERIVATIONS). Arguments with the same support are then merged (there is at most one argument per set of assumptions) by calling MERGE-BY-SUP.

Each constructed argument contains only sentences derivable from its set of assumptions. For each $\Delta \in minsupp(D)$ an argument with $\Delta$ as its assumptions is con-

**Algorithm 2** PROCESS-HEAD$(s, P)$

1: **if** $s \notin \bigcup SCC(D)$ **then** mark $s$ visited
2: $P = P \cup \{s\}$
3: **if** $s \in \mathcal{A} \cup \{\top\}$ **then**
4: $\quad Arg(s) = Arg(s) \cup \{\{s\}\}$
5: **else**
6: $\quad$ **for each** $B \in \{body(r) \mid head(r) = s\}$
7: $\quad\quad$ **if** $B$ not visited **then**
8: $\quad\quad\quad$ PROCESS-BODY$(B, P)$
9: $\quad\quad Arg(B) = \{A \cup \{s\} \mid A \in Arg(B)\}$
10: $\quad\quad Arg(s) = Arg(s) \cup Arg(B)$
11: $P = P \setminus \{s\}$

**Algorithm 3** PROCESS-BODY$(B, P)$

1: **if** $B \notin \bigcup SCC(D)$ **then** mark $B$ visited
2: **if** each $s' \in B$ is marked or not in SCC **then** mark $B$ visited
3: **if** $B \cap P \neq \emptyset$ **then** return
4: $P = P \cup \{B\}$
5: **for each** $s' \in B$
6: $\quad$ **if** $s'$ not visited **then**
7: $\quad\quad$ PROCESS-HEAD$(s', P)$
8: $Arg(B)$ = SUBDERIVATIONS$(B)$
9: MERGE-BY-SUPP$(Arg(B))$
10: $P = P \setminus \{B\}$

structed. Our algorithm approximates the set of relevant arguments in two senses. First, arguments with minimal support might contain more derived sentences, i.e., sentences also derivable from subsets of their support. Secondly, we might compute arguments with assumption sets not in $minsupp(D)$. Correctness of the overall approach is not affected by either approximation as long as the attacks are as specified in Definition 5.

**Special Cases** ABA acceptance can, in cases, be decided during the AF translation. Assume an ABA $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ and a sentence $l \in \mathcal{L}$. For admissible and preferred semantics it holds that if $l \in Th_{\mathcal{R}}(\emptyset)$, then $l$ is both credulously and skeptically accepted; and if $l \notin Th_{\mathcal{R}}(\mathcal{A})$ or each $(L, \Delta)$ with $l \in L$ is self-attacking, then $l$ is neither credulously nor skeptically accepted. For stable semantics, it holds that if $l \in Th_{\mathcal{R}}(\emptyset)$, then $l$ is skeptically accepted, and credulously accepted iff $D$ has a stable assumption set. If $l \notin Th_{\mathcal{R}}(\mathcal{A})$ or each $(L, \Delta)$ with $l \in L$ is self-attacking, then $l$ is not credulously accepted, and skeptically accepted iff $D$ has no stable assumption set. In our approach, existence of stable assumption sets can be checked with an AF reasoner.

## 5 Reasoning about ABA Acceptance on AFs

For reasoning over the AFs (step 2) obtained from our ABA-to-AF translation (step 1) we encode the ABA acceptance problem over the AF obtained via step 1 using answer set programming (ASP). Interchangeably, one could apply essentially any of the e.g. SAT-based AF reasoning systems via similar minor modifications. We focus here on encodings for admissible and stable semantics; other central argumentation semantics can be encoded with relatively minor changes. Our encodings are similar to ones used in the ASP-based AF reasoning system ASPARTIX [18], except for one seemingly minor but essential difference: we represent the AF attack relation via its *complement*, using the predicate `natt`$/2$ (not attack) which is true for a pair $(a, b)$ of nodes iff $a$ *does not* attack $b$. This complement representation is vital as the edge relations of the AFs obtained via step 1 are typically very dense. This is in stark contrast to typical AF reasoning benchmark instances with relatively sparse attack relations [27].

Our encoding of the admissible semantics is

$\texttt{in}(X)$ :- $not$ $\texttt{out}(X), \texttt{arg}(X).$ :- $\texttt{in}(X), \texttt{in}(Y), not\ \texttt{natt}(X, Y).$

$\texttt{out}(X)$ :- $not$ $\texttt{in}(X), \texttt{arg}(X).$ $\texttt{defeated}(X)$ :- $\texttt{in}(Y), not\ \texttt{natt}(Y, X), \texttt{arg}(X).$

$\texttt{out}(X)$ :- $\texttt{arg}(X), not\ \texttt{natt}(X, X).$ :- $\texttt{in}(X), \texttt{arg}(Y), not\ \texttt{natt}(Y, X), not\ \texttt{defeated}(Y).$

Further minor changes to the original ASPARTIX encoding are that we include the rule on the lower left, and collapsed two rules to what is now here the last rule. For stable semantics, we simply replace the last rule by :- $\texttt{out}(X), not\ \texttt{defeated}(X)$.

**Implementing credulous ABA queries** under NP-complete semantics such as admissible, preferred, and stable on the AF side is achieved by checking if there is an extension which includes an argument that contains the queried sentence during argument construction. This is implemented with the ASP constraint :- not $\texttt{in}(a_1)$, ..., not $\texttt{in}(a_n)$. for arguments $a_i$ that contain queried sentence $l$, i.e., $a_i = (L, \Delta)$ with $l \in L$.

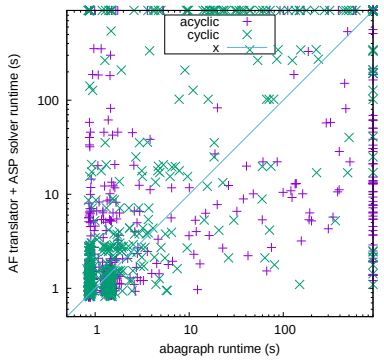
**Implementing skeptical ABA queries** for coNP-complete semantics such as stable is achieved by checking if there is a counterexample to the query, i.e., whether there is an extension of the AF that does not include any arguments containing the queried sentence. This is implemented by constraint :- $\texttt{in}(a_i)$. for each argument $a_i$ that contains the queried sentence, pruning the search space by partial instantiation. An alternative approach to skeptical acceptance would be to enumerate all AF extensions, and check whether each of them includes some argument that contains the queried sentence.

## 6 Experiments

For a first evaluation of the two-step approach to answering ABA queries via AF reasoning, we implemented a prototype translation (step 1) in Java 8. We compare our approach to the recently published state-of-the-art graph-based ABA reasoning system abagraph (`http://www.doc.ic.ac.uk/~rac101/proarg/abagraph.html`) implemented in Prolog, using SICStus Prolog 4.3.3. We used the "default" search strategy of abagraph. The experiments were run on 2.83-GHz Intel Xeon E5440 quad-core machines with 32-GB RAM under Linux using a 600-second timeout and a 16-GB memory limit per instance. As the running times of our approach, we report the combined time of the translation and the ASP solver Clingo 4.5.4 [23].

As benchmarks we use the 680 ABA frameworks provided by the authors of abagraph [7]. A benchmark instance consists of an ABA graph and a query on whether a given sentence in the ABA framework is credulously accepted under a specific semantics (recall that abagraph supports only credulous queries under admissible and grounded semantics). For each ABA framework, we used 10 queries per ABA. After filtering out 90 duplicate queries and trivial instances wrt the special cases outlined for admissible semantics in the previous section, we obtained 1466 final benchmark instances. The benchmarks are explicitly categorized wrt whether the rules of a framework give rise to cyclic dependencies, i.e., whether a framework is cyclic (804) or acyclic (662).

A comparison of abagraph and our approach is shown in Fig. 3 left. Here we consider the credulous task of enumerating all admissible assumption sets containing the




| | Timeouts | | Uniquely solved | |
|---|---|---|---|---|
| | abagraph | us | abagraph | us |
| acyclic | 93 | **56** | 20 | **57** |
| cyclic | **394** | 402 | **86** | 78 |

**Fig. 3.** Left: running time comparison of abagraph and our approach on credulous reasoning under admissible semantics. Right: numbers of timeouts and uniquely solved instances.

given query. The same task was used to evaluate abagraph in [7] and shown to outperform earlier state of the art. For enumeration in our approach, we used the built-in enumeration mode of Clingo. Fig. 3 (left) shows that our approach and that of the dedicated abagraph approach are complementary in that there are instances on which each of the approaches is clearly better than the other. Fig. 3 (right) corroborates this observation. The relative performance is essentially on-par on cyclic instances, while our approach is somewhat better on acyclic instances. To illustrate the generality of our approach, we also experimented on skeptical acceptance of sentences under stable semantics (a task not supported by abagraph). Our approach solved 6228 of the 6710 instances. The per-instance runtime was $< 10$ s on over 6000 instances. A majority of runtime was used in the AF translation on every instance, AF translation taking over 80% of the total runtime on approximately 95% of the solved instances. The ASP solving part was very efficient, finishing within 65 seconds on each instance.

## 7 Conclusions

We studied an approach to reasoning about acceptance in assumption-based argumentation via translating ABA frameworks into argumentation frameworks. We considered relevant ABA arguments as a sought after small yet sufficient set for reasoning about acceptance of ABA sentences on AFs. However, we showed that counting the number of relevant arguments is #P-complete, and hence proposed an algorithm for overapproximating the set of relevant arguments in order to translate ABAs to AFs, and ASP encodings specifically suited for the types of AFs obtained through the translation. Our prototype implementation yields complementary performance wrt the state-of-the-art dedicated ABA reasoning system abagraph. As a further benefit, our approach also allows for deciding skeptical acceptance in ABA, not supported by abagraph.

## References

1. Besnard, P., García, A.J., Hunter, A., Modgil, S., Prakken, H., Simari, G.R., Toni, F.: Introduction to structured argumentation. Argument & Computation 5(1), 1–4 (2014)

2. Besnard, P., Hunter, A.: Elements of Argumentation (2008)
3. Bondarenko, A., Dung, P.M., Kowalski, R.A., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. Artif. Intell. 93, 63–101 (1997)
4. Caminada, M., Sá, S., Alcântara, J., Dvořák, W.: On the difference between assumption-based argumentation and abstract argumentation. In: Proc. BNAIC. pp. 25–32 (2013)
5. Cerutti, F., Dunne, P.E., Giacomin, M., Vallati, M.: Computing preferred extensions in abstract argumentation: A SAT-based approach. In: TAFA 2013 Revised Selected Papers. LNCS, vol. 8306, pp. 176–193 (2014)
6. Cerutti, F., Giacomin, M., Vallati, M.: ArgSemSAT: Solving argumentation problems using SAT. In: Proc. COMMA. FAIA, vol. 266, pp. 455–456 (2014)
7. Craven, R., Toni, F.: Argument graphs and assumption-based argumentation. Artif. Intell. 233, 1–59 (2016)
8. Craven, R., Toni, F., Cadar, C., Hadad, A., Williams, M.: Efficient argumentation for medical decision-making. In: Proc. KR. pp. 598–602 (2012)
9. Craven, R., Toni, F., Williams, M.: Graph-based dispute derivations in assumption-based argumentation. In: TAFA 2013 Revised Selected Papers. LNCS, vol. 8306, pp. 46–62 (2014)
10. Dimopoulos, Y., Nebel, B., Toni, F.: On the computational complexity of assumption-based argumentation for default reasoning. Artif. Intell. 141(1/2), 57–78 (2002)
11. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. 77(2), 321–358 (1995)
12. Dung, P.M., Kowalski, R.A., Toni, F.: Dialectic proof procedures for assumption-based, admissible argumentation. Artif. Intell. 170(2), 114–159 (2006)
13. Dung, P.M., Kowalski, R.A., Toni, F.: Assumption-based argumentation. In: Rahwan, I., Simari, G.R. (eds.) Argumentation in Artificial Intelligence, pp. 25–44 (2009)
14. Dung, P.M., Mancarella, P., Toni, F.: Computing ideal sceptical argumentation. Artif. Intell. 171(10-15), 642–674 (2007)
15. Dunne, P.E., Wooldridge, M.: Complexity of abstract argumentation. In: Rahwan, I., Simari, G.R. (eds.) Argumentation in Artificial Intelligence, pp. 85–104 (2009)
16. Durand, A., Hermann, M., Kolaitis, P.G.: Subtractive reductions and complete problems for counting complexity classes. Theor. Comput. Sci. 340(3), 496–513 (2005)
17. Dvořák, W., Järvisalo, M., Wallner, J.P., Woltran, S.: Complexity-sensitive decision procedures for abstract argumentation. Artif. Intell. 206, 53–78 (2014)
18. Egly, U., Gaggl, S.A., Woltran, S.: Answer-set programming encodings for argumentation frameworks. Argument & Computation 1(2), 147–177 (2010)
19. Gaertner, D., Toni, F.: CaSAPI: a system for credulous and sceptical argumentation. In: Proc. NMR. pp. 80–95 (2007)
20. Gaertner, D., Toni, F.: Computing arguments and attacks in assumption-based argumentation. IEEE Intell. Syst. 22(6), 24–33 (2007)
21. Gaertner, D., Toni, F.: Hybrid argumentation and its properties. In: Proc. COMMA. FAIA, vol. 172, pp. 183–195 (2008)
22. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. TPLP 4(1-2), 95–138 (2004)
23. Gebser, M., Kaufmann, B., Kaminski, R., Ostrowski, M., Schaub, T., Schneider, M.T.: Potassco: The potsdam answer set solving collection. AI Comm. 24(2), 107–124 (2011)
24. Nofal, S., Atkinson, K., Dunne, P.E.: Algorithms for decision problems in argument systems under preferred semantics. Artif. Intell. 207, 23–51 (2014)
25. Nofal, S., Atkinson, K., Dunne, P.E.: Looking-ahead in backtracking algorithms for abstract argumentation. Int. J. Approx. Reasoning 78, 265–282 (2016)
26. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument & Computation 1(2), 93–124 (2010)

27. Thimm, M., Villata, S., Cerutti, F., Oren, N., Strass, H., Vallati, M.: Summary report of the first international competition on computational models of argumentation. AI Magazine 37(1), 102 (2016)
28. Toni, F.: A generalised framework for dispute derivations in assumption-based argumentation. Artif. Intell. 195, 1–43 (2013)
29. Toni, F.: A tutorial on assumption-based argumentation. Argument & Computation 5(1), 89–117 (2014)