Reasoning over Assumption-Based Argumentation Frameworks via Direct Answer Set Programming Encodings

Tuomo Lehtonen
University of Helsinki, Finland

Johannes P. Wallner
TU Wien, Austria

Matti Järvisalo
University of Helsinki, Finland

Abstract

Focusing on assumption-based argumentation (ABA) as a central structured formalism to AI argumentation, we propose a new approach to reasoning in ABA with and without preferences. While previous approaches apply either specialized algorithms or translate ABA reasoning to reasoning over abstract argumentation frameworks, we develop a direct approach by encoding ABA reasoning tasks in answer set programming. This significantly improves on the empirical performance of current ABA reasoning systems. We also give new complexity results for reasoning in ABA$^+$, suggesting that the integration of preferential information into ABA results in increased problem complexity for several central argumentation semantics.

1 Introduction

The study of computational models for argumentation is a vibrant area of AI and knowledge representation (KR) research. As KR formalisms, computational models of argumentation capture various related paradigms, including nonmonotonic reasoning and logic programming (Dung 1995), and furthermore open up avenues for various types of applications (Atkinson et al. 2017).

Arguments most often have an intrinsic structure made explicit through derivations from more basic structures. Computational models for *structured argumentation* (Besnard et al. 2014; Bondarenko et al. 1997; García and Simari 2004; Besnard and Hunter 2008; Prakken 2010) provide tools for making the internal structure of arguments explicit. This is in contrast to abstract argumentation, where the structure of individual arguments is completely abstract, and reasoning over abstract argumentation frameworks (AFs) (Dung 1995) is restricted to the level of pairwise knowledge of attacks between conflicting arguments. While there has been noticeable attention on computational approaches for AFs (Charwat et al. 2015), advancing understanding of the complexity of and algorithms for reasoning over structured argumentation frameworks has received less attention and can be considered more challenging.

In this paper we focus on assumption-based argumentation (ABA) (Bondarenko et al. 1997; Dung, Kowalski, and Toni 2009; Toni 2014; Cyras et al. 2018) as one of the central structured argumentation formalisms which provides a suitable model for various application scenarios (Craven et al. 2012; Fan et al. 2014; Matt et al. 2008). We propose a new computational approach to reason in ABA with and without preferences. While previous approaches are based on either specialized algorithms or translating ABA reasoning to reasoning over AFs, we propose a direct approach to ABA reasoning based on encoding ABA reasoning tasks in answer set programming (ASP) (Gelfond and Lifschitz 1988; Niemelä 1999). While related encodings have been considered previously focusing on representational aspects in the context of ABA (Egly and Woltran 2006; Caminada and Schulz 2017), motivated by the success of ASP encodings for AFs (Toni and Sergot 2011) we present novel types of ASP encodings for ABA, and for ABA$^+$ (Cyras and Toni 2016a; 2016c; Bao, Cyras, and Toni 2017) that integrates preferences into ABA, and in particular provide a first empirical evaluation of this approach. Our approach significantly improves on the empirical performance of currently available ABA reasoning systems, in particular the dispute derivation approach (Gaertner and Toni 2007a; 2007b; 2008; Dung, Mancarella, and Toni 2007; Craven et al. 2012; Toni 2013; Craven, Toni, and Williams 2013) implemented in the abagraph system (Craven and Toni 2016) and the translation-based approach (Dung, Mancarella, and Toni 2007; Caminada et al. 2013) implemented in the aba2af system (Lehtonen, Wallner, and Järvisalo 2017) for ABA, as well as the ABAplus system (Bao, Cyras, and Toni 2017) for ABA$^+$ to the extent applicable. Furthermore, while the complexity of reasoning over ABA frameworks is well-understood (Dimopoulos, Nebel, and Toni 2002), the complexity of reasoning in the ABA$^+$ formalism is currently less understood. Towards bridging this gap, we also give new complexity results for reasoning in ABA$^+$, suggesting that the integration of preferential information into ABA results in increased problem complexity for several central argumentation semantics. The complexity results justify why normal ASP programs are enough to encode ABA$^+$ acceptance under stable semantics, and also explain why NP-reasoning (e.g., ASP over normal logic programs) is not expected to be powerful enough to allow for direct polynomial-size encodings for the other semantics; in particular, verification of admissibility under preferences turns out to be coNP-complete.

2 Assumption-Based Argumentation

We recall assumption-based argumentation (ABA) (Bondarenko et al. 1997; Toni 2014; Cyras et al. 2018) and ABA$^+$ (Cyras and Toni 2016a; 2016c; Bao, Cyras, and Toni 2017) that equips ABA with preferences over assumptions. Since ABA$^+$ is a strict generalization of ABA, we directly define the former and point out differences to the latter.

We assume a deductive system $(\mathcal{L}, \mathcal{R})$ with $\mathcal{L}$ a formal language, i.e., a 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 central concept in ABA$^+$ is how a sentence can be derived from a given set of assumptions and a set of rules. In ABA, several notions of derivability are studied (Dung, Kowalski, and Toni 2006; Dung, Toni, and Mancarella 2010), with tree-derivability ($\models$) the most commonly considered one. ABA (without preferences) can be equivalently defined via forward-derivability, and we will make use of this fact here for developing ASP encodings for ABA under various semantics. In contrast, tree-derivability is central for defining ABA$^+$, and the equivalence with forward-derivability does not carry over in general to ABA$^+$. However, as we will show, tree-derivability and forward-derivability remain equivalent also for ABA$^+$ under specific semantics, which allows for an ASP encoding similar to those we develop for ABA.

A sentence $s \in \mathcal{L}$ is tree-derivable from a set of assumptions $X \subseteq \mathcal{A}$ and rules $R \subseteq \mathcal{R}$, denoted by $X \models_R s$, if there is a finite tree with the root labeled by s, the leaves labeled by elements of X, and for each internal node labeled with t there is a rule $r \in R$ s.t. the set of labels of the children of this node is $body(r)$ and the node itself is labeled by $head(r)$. For each rule $r \in R$ there is a node labeled in this way. In brief, there is a derivation tree from assumptions X and rules R to sentence s. Unless not clear from the context, we will write $\models_R$ without explicitly defining R and assume that $R \subseteq \mathcal{R}$ from the given deductive system.

A sentence $a \in \mathcal{L}$ is forward-derivable from a set $X \subseteq \mathcal{A}$ via rules $\mathcal{R}$, denoted by $X \vdash_{\mathcal{R}} a$, if there is a sequence of rules $(r_1, \ldots, r_n)$ such that $head(r_n) = a$, for each rule r_i we have $r_i \in \mathcal{R}$, and each sentence in the body of r_i is derived from rules earlier in the sequence or in X, i.e., $body(r_i) \subseteq X \cup \bigcup_{j<i} head(r_j)$.

If $X \models_R s$ then $X \vdash_{\mathcal{R}} s$, and if $X \vdash_{\mathcal{R}} s$ there is an $X' \subseteq X$ and $R \subseteq \mathcal{R}$ such that $X' \models_R s$. In words, $\models$ is "stricter" (requires a tree derivation where all assumptions and rules are required for derivation), while $\vdash$ is simpler (no witness tree is required and redundant assumptions and rules are allowed). The deductive closure for an assumption set X w.r.t. rules $\mathcal{R}$ is given by $Th_{\mathcal{R}}(X) = \{a \mid X \vdash_{\mathcal{R}} a\}$ (there is an equivalent definition via $\models$). If the type of derivation is not relevant, we generally refer to derivations.

An ABA$^+$ framework is a tuple $F = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-, \leq)$ with $(\mathcal{L}, \mathcal{R})$ a deductive system, a set of assumptions $\mathcal{A} \subseteq \mathcal{L}$, a function $^-$ mapping assumptions $\mathcal{A}$ to sentences $\mathcal{L}$, and a preorder $\leq$ on $\mathcal{A}$. The strict counterpart $<$ of $\leq$ is defined as usual by $a < b$ iff $a \leq b$ and $b \not\leq a$, for $a, b \in \mathcal{A}$. In this paper, we focus on so-called flat ABA$^+$ frameworks where assumptions cannot be derived, i.e., do not occur in heads of rules. We also assume that each set in a tuple defining an ABA$^+$ is finite.

Definition 1. *Let $(\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-, \leq)$ be an ABA$^+$ framework, and $A, B \subseteq \mathcal{A}$ be two sets of assumptions. Assumption set A $<$-attacks B if*

- *$A' \models_R \overline{b}$ for some $A' \subseteq A$, $b \in B$, and $\nexists a' \in A'$ with $a' < b$, or*
- *$B' \models_R \overline{a}$ for some $a \in A$ and $B' \subseteq B$ s.t. $\exists b' \in B'$ with $b' < a$.*

In words, set A attacks B if (i) from a subset A' of A, one can tree-derive a contrary of an assumption $b \in B$ and no member in A' is strictly less preferred than b, or (ii) from B, via subset B' one can tree-derive a contrary of an assumption $a \in A$ and some member of B' is strictly less preferred than a. Attacks of type (i) are *normal* $<$-attacks and those of type (ii) *reverse* $<$-attacks, with the intuition that the (non-preference based) conflict in (i) succeeds and in case of (ii) is countered and reversed by the preference relation.

Definition 2. *Let $F = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-, \leq)$ be an ABA$^+$ framework. An assumption set $A \subseteq \mathcal{A}$ is called conflict-free if A does not $<$-attack itself. Set A defends assumption set $B \subseteq \mathcal{A}$ if for all $C \subseteq \mathcal{A}$ that $<$-attack B it holds that A $<$-attacks C.*

Definition 3. *Let $F = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-, \leq)$ be an ABA$^+$ framework. Further, let $A \subseteq \mathcal{A}$ be a conflict-free set of assumptions in F. Set A is*

- $<$-admissible *in F if A defends itself;*
- $<$-complete *in F if A is admissible in F and contains every assumption set defended by A;*
- $<$-grounded *in F if A is the intersection of all $<$-complete assumption sets;*[1]
- $<$-preferred *in F if A is $<$-admissible and there is no $<$-admissible set of assumptions B in F with $A \subset B$; and*
- $<$-stable *in F if each $\{x\} \subseteq \mathcal{A} \setminus A$ is $<$-attacked by A.*

We use the term $<$-σ-assumption-set for an assumption set under a semantics $\sigma \in \{adm, com, grd, stb, prf\}$, i.e., $<$-admissible, $<$-complete, $<$-grounded, $<$-stable, and $<$-preferred semantics, respectively.

An ABA framework, that is ABA$^+$ without preferences, is a tuple $F = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-, \emptyset)$ (i.e. $\leq = \emptyset$). We will denote ABA frameworks by $F = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$, i.e., omitting the empty preference relation, and refer to the corresponding semantics, likewise, without the preference relation $<$ (e.g., complete semantics instead of $<$-complete semantics). Attacks ($\emptyset$-attacks) in ABA frameworks simplify to attacks from assumption sets A to assumption sets B when $A \vdash_{\mathcal{R}} \overline{b}$ for $b \in B$ (these are normal $<$-attacks when no preference information is available; reverse $<$-attacks are not present in ABA). Note that forward-derivability is sufficient for ABA.

Main reasoning tasks on ABA$^+$ are the following.

[1] In non-flat ABA$^+$ grounded semantics is referred to as well-founded semantics. We consider here only flat frameworks.

Definition 4. *Let $F = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-, \leq)$ be an ABA$^+$ framework and $<$-σ a semantics. A sentence $s \in \mathcal{L}$ is*

- *credulously accepted in F under semantics $<$-σ if there is a $<$-σ-assumption-set A s.t. $s \in Th_\mathcal{R}(A)$; and*
- *skeptically accepted in F under semantics $<$-σ if $s \in Th_\mathcal{R}(A)$ for all $<$-σ-assumption-sets A.*

The tasks for ABA are analogous, the preference relation $<$ is simply disregarded. For ABA, credulous reasoning under admissible, complete, and preferred semantics coincide (Bondarenko et al. 1997; Cyras et al. 2018).

Example 1. *Let $F = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA with assumptions $\mathcal{A} = \{a, b, c, d\}$, sentences $\mathcal{L} = \{x, b^c, c^c, d^c\} \cup \mathcal{A}$, rules $(x \leftarrow a)$, $(c^c \leftarrow b, x)$, $(b^c \leftarrow c)$, and $(d^c \leftarrow a, b)$, where b^c, c^c, and d^c are the contraries of b, c, and d, respectively. The complete assumption sets are $\{a\}$, $\{a, b\}$, and $\{a, c, d\}$, with $\{a\}$ the grounded assumption set (this assumption is not attacked). The sets $\{a, b\}$ and $\{a, c, d\}$ are also stable: e.g., the first set attacks via the second and fourth rules the assumptions $\{c\}$ and $\{d\}$. Sentence x is credulously accepted under admissible, complete, stable, and preferred. Under preference $a < d$, in the resulting ABA$^+$ $F' = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-, \leq)$ the set $\{a, b\}$ is not $<$-stable (the attack from $\{a, b\}$ to d is reversed), while set $\{a, c, d\}$ is $<$-stable. By the further preference $c < b$, the set $\{c\}$ does not $<$-attack $\{b\}$, resulting in no $<$-stable assumption sets.*

Computational complexity of reasoning over ABA is well-established (Dimopoulos, Nebel, and Toni 2002): credulous reasoning for admissible, stable, and preferred semantics is NP-complete, while skeptical reasoning is P-complete under admissible, coNP-complete under stable, and Π_2^p-complete under preferred semantics. Complexity of ABA with preferences has, to our knowledge, not being studied in depth; Wakaki (2017) showed complexity results for p_ABA, an alternative way of handling preferences in ABA. However, the results do not directly transfer to ABA$^+$.

Finally, the so-called Axiom of Weak Contraposition (WCP) bridges ABA$^+$ with the semantics of ABA. Namely, if an ABA$^+$ framework satisfies WCP, then further properties on the semantics hold (Cyras and Toni 2016b), e.g., the grounded assumption set is guaranteed to exist.

Definition 5. *An ABA$^+$ $(\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-, \leq)$ satisfies the Axiom of Weak Contraposition (WCP) if for each $A \subseteq \mathcal{A}$, $R \subseteq \mathcal{R}$, and $b \in \mathcal{A}$, if $A \models_R \overline{b}$ and $\exists a' \in A$ such that $a' < b$, then there is a $\leq$-minimal $a \in A$ such that $a < b$ and $A' \models_{R'} \overline{a}$ for some $A' \subseteq (A \setminus \{a\}) \cup \{b\}$ and $R' \subseteq \mathcal{R}$.*

3 Properties and Complexity Results

We discuss properties essential for developing our ASP encodings, as well as establish complexity results for ABA$^+$.

ABA. We begin with slightly re-stating some of the ABA semantics in terms that are suited for subsequent encodings. Let $(\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework and $E \subseteq \mathcal{A}$ be a set of assumptions. Define $D_E = \{a \in \mathcal{A} \mid E \text{ attacks } \{a\}\}$. Recall that in ABA E attacks $\{a\}$ iff $E \vdash_\mathcal{R} \overline{a}$.

Proposition 1. *Let $(\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework and $E \subseteq \mathcal{A}$ be a conflict-free set of assumptions. It holds that*

- *set E defends an assumption set $\{a\} \subseteq \mathcal{A}$ iff $\mathcal{A} \setminus D_E$ does not attack $\{a\}$;*
- *set E is admissible iff $\mathcal{A} \setminus D_E$ does not attack E; and*
- *set E is complete iff E is admissible and for each $b \in \mathcal{A} \setminus E$ we have $\{b\}$ is attacked by $\mathcal{A} \setminus D_E$.*

For Proposition 1, which follows from the definitions, the main ingredient in checking if a set is admissible or complete is the set D_E (which can be computed in polynomial time since $\vdash_\mathcal{R}$ is decidable in polynomial time). Further, for ABA frameworks, the grounded semantics can be characterized by the function $def(A) = \{a \in \mathcal{A} \mid A \text{ defends } a\}$, as shown in (Bondarenko et al. 1997, Theorem 6.2). The grounded assumption set of an ABA framework is the least fixed point of def. The complexity of grounded and complete semantics of ABA has, to our knowledge, not been made explicit, but since D_E, and thus def, can be computed in polynomial time, one can straightforwardly infer their complexity.

Corollary 2. *For a given ABA framework, one can in polynomial time (i) compute the grounded assumption set, and (ii) verify whether a given set of assumptions is complete.*

ABA$^+$. Tree-derivability is used for defining $<$-attacks in ABA$^+$. Unlike in ABA, tree-derivations and forward-derivations are not equivalent when using them to define $<$-attacks. This can be seen when A normally $<$-attacks B: adding redundant assumptions to A (not used for deriving a sentence) may weaken the set and make it open for reverse $<$-attacks that are not present in the original definition via $\models$ (e.g. by $A' = A \cup \{a'\}$ with $a' < b$ for a $b \in B$ and $A' \vdash_\mathcal{R} \overline{b}$ but $A' \not\models_R \overline{b}$). However, normal $<$-attacks can be defined via $\vdash$, which follows from (Dung, Kowalski, and Toni 2006; Dung, Toni, and Mancarella 2010), and reverse $<$-attacks in the special case of conflict-free sets A that reversely $<$-attack a singleton set $\{b\}$.

Lemma 3. *Let $(\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-, \leq)$ be an ABA$^+$ framework, and $A, B \subseteq \mathcal{A}$ be two sets of assumptions, and $b \in \mathcal{A}$.*

- *Set A normally $<$-attacks B iff $A' \vdash_\mathcal{R} \overline{b}$, for some $A' \subseteq A$, $b \in B$, and $\nexists a' \in A'$ with $a' < b$.*
- *If A is conflict-free, we have A reversely $<$-attacks $\{b\}$ iff $\{b\} \vdash_\mathcal{R} \overline{a}$ for some $a \in A$ and $b < a$.*

To see why a conflict-free set A reversely $<$-attacks $\{b\}$ as stated in the second condition of the lemma, first note that, since A is conflict-free, we have $\emptyset \not\models_R \overline{a}$ for all $a \in A$. Thus also $\emptyset \not\vdash_\mathcal{R} \overline{a}$. This means that $\{b\} \vdash_\mathcal{R} \overline{a}$ iff $\{b\} \models_R \overline{a}$, for some $a \in A$, since there is no subset of $\{b\}$ that forward-derives $\overline{a}$, by presumption of A being conflict-free.

A corollary of Lemma 3 is that one can check whether a set of assumptions is $<$-stable via relying only on forward-derivability. To make this explicit, we slightly re-state $<$-stable semantics in the following proposition.

Proposition 4. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-, \leq)$ be an ABA$^+$ framework. A conflict-free set $E \subseteq \mathcal{A}$ is $<$-stable iff for all $\{b\} \subseteq \mathcal{A}$ that are not normally $<$-attacked by E, either $b \in E$ or $\{b\}$ is reversely $<$-attacked by E.*

In this proposition we have explicitly distinguished between normal and reverse $<$-attacks (which will also be useful for our ASP encodings), and explicated that the required conditions only rely on the notion of forward-derivability.

We move on to investigating the complexity of ABA$^+$ semantics. Central to understanding the complexity of ABA$^+$ is understanding the complexity of $<$-attacks. In ABA, attacks take a relatively simple form: sets of assumptions attack single assumptions in the definition of an attack in ABA. In both ABA and ABA$^+$ attacks satisfy $\subseteq$-monotonicity of the following form: if A ($<$-)attacks B then A' ($<$-)attacks B' whenever $A \subseteq A'$ and $B \subseteq B'$ (Cyras and Toni 2016b, Lemma 3). That is, in ABA, if A attacks b, then A attacks any B with $b \in B$. Attacks in ABA$^+$, while also $\subseteq$-monotone, may in cases not attack singleton sets of assumptions. Normal $<$-attacks in an ABA$^+$ framework are similar to attacks in an ABA, reverse $<$-attacks *originate* from singleton sets of assumptions, yet attack a set of assumptions. Put differently, in ABA, if a set B is attacked, by A, then there is at least one singleton set $\{b\} \subseteq B$ such that $\{b\}$ is attacked by A. The same does not hold for reverse $<$-attacks in ABA$^+$.

As we show, the difference in attacks between ABA and ABA$^+$ is significant from a computational perspective. We classify four types of (counter) $<$-attacks that are useful for understanding complexity of ABA$^+$. For two sets of assumptions, A and B, we distinguish whether (i) A normally $<$-attacks B, (ii) A reversely $<$-attacks B, (iii) A normally $<$-attacks all subsets $B' \subseteq B$ that $<$-attack A, and (iv) A reversely $<$-attacks all subsets $B' \subseteq B$ that $<$-attack A.

Proposition 5. *Let $(\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-}, \leq)$ be an ABA$^+$ framework, and A and B two sets of assumptions. One can decide in polynomial time whether set A*

1. *normally $<$-attacks B,*
2. *reversely $<$-attacks B, or*
3. *normally $<$-attacks all subsets $B' \subseteq B$ that $<$-attack A.*

Proof. (sketch) For (1), consider each $b \in B$ separately and compute $A_b = \{a \in A \mid a \not< b\}$. If $A_b \vdash_{\mathcal{R}} \overline{b}$ then A_b normally $<$-attacks $\{b\}$, and by monotonicity A normally $<$-attacks B. For (2), compute $Th_{\mathcal{R}}(B)$. For each $a \in A$, if $\overline{a} \in Th_{\mathcal{R}}(B)$, construct a directed graph with node set equal to $Th_{\mathcal{R}}(B)$ and iteratively add edges from $\overline{a}$ to sentences x whenever there is a rule applicable from $Th_{\mathcal{R}}(B)$ with $\overline{a}$ in the head and x in the body; continue this process for the body elements x as rules' heads and their bodies (until a fixed point is reached). There is a tree-derivation $B' \models_R \overline{a}$ with $B' \subseteq B$ such that there is a $b \in B'$ with $b < a$ iff there is a path from $\overline{a}$ to such a b. For (3), compute $X = \{b \in B \mid A$ normally $<$-attacks $\{b\}\}$, which can be done in polynomial time according to (1). Next, check whether $B \setminus X$ $<$-attacks A (via 1 and 2). If so, then A does not counter all $<$-attacks via normal $<$-attacks. $\square$

Note that the item 2 of Proposition 5 is polynomial-time decidable, since one does not need to find subset-minimal $B' \subseteq B$ that are reversely $<$-attacked by A. Item 3 is likewise polynomial-time decidable, since normal $<$-attacks target only singleton sets of assumptions. Before delving into attacks of type (iv), we derive complexity of verifying whether a set of assumptions is $<$-stable based on Proposition 5. Note that conflict-freeness does not depend on the preference relation (Cyras and Toni 2016b, Theorem 5), and can be checked in polynomial time.

Theorem 6. *Verifying whether a set of assumptions is $<$-stable in an ABA$^+$ framework is in P. Checking credulous (skeptical) acceptance of a sentence under $<$-stable semantics in ABA$^+$ is NP-complete (coNP-complete).*

This means that $<$-stable semantics exhibits the same computational complexity as (non-preference-based) stable semantics in ABA. Complexity does not carry over from ABA to ABA$^+$ for the other semantics under consideration. In fact, when considering counterattacks (defense) against potentially non-singleton sets of assumptions, i.e., scenario (iv) (from before Proposition 5), it is coNP-hard to decide.

Proposition 7. *Deciding whether a given set A of assumptions in an ABA$^+$ framework reversely counterattacks all assumption sets that $<$-attack A is coNP-complete. Hardness holds even when assuming WCP.*

Proof. (sketch) Membership in coNP can be seen from considering the complementary problem: for assumption set A, to check non-$<$-admissibility, guess set B, check whether B $<$-attacks A and whether A does not $<$-reversely attack B. Let $\phi = c_1 \wedge \cdots \wedge c_m$ be a Boolean formula in conjunctive normal form over vocabulary $X = \{x_1, \ldots, x_n\}$ and clauses $C = \{c_1, \ldots, c_m\}$. For a clause $c_j = l_1 \vee \cdots \vee l_s$ let $neg(c_j) = \neg l_1, \ldots, \neg l_s$ with $\neg l_i = x$ if $l_i = \neg x$ and $\neg l_i = \neg x$ if $l_i = x$. Construct ABA$^+$ $F = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-}, \leq)$ with $\mathcal{A} = \{a, b\} \cup X \cup \neg X$ for $\neg X = \{\neg x \mid x \in X\}$ and the following rules.

$$
\begin{array}{ll}
d_i \leftarrow x_i \quad \text{and} \quad d_i \leftarrow \neg x_i & \text{for } 1 \leq i \leq n; \\
a^c \leftarrow d_1, \ldots, d_m; & \\
b^c \leftarrow x_i, \neg x_i, a^c & \text{for } 1 \leq i \leq n; \text{ and} \\
b^c \leftarrow neg(c_j), a^c & \text{for } 1 \leq j \leq m,
\end{array}
$$

with a^c and b^c the contraries of a and b respectively (no other contraries are needed). Complete the instance with setting $b > x_i$ and $b > \neg x_i$ for $1 \leq i \leq n$.

We claim that $\{a, b\}$ is reversely $<$-attacking all assumption sets X that $<$-attack $\{a, b\}$ in F iff ϕ is unsatisfiable. Note that one cannot derive a contrary of any assumption from $\{a, b\}$. Thus $\{a, b\}$ does not normally $<$-attack any assumption set and no assumption set can reversely $<$-attack $\{a, b\}$ Thus, $\{a, b\}$ is reversely $<$-attacking all assumption sets X that $<$-attack $\{a, b\}$ in F iff $\{a, b\}$ is $<$-admissible.

Assume that $\{a, b\}$ is not $<$-admissible. Since $\{a, b\}$ is conflict-free, there is a $B \subseteq \mathcal{A}$ s.t. B $<$-attacks $\{a, b\}$ and $\{a, b\}$ does not $<$-attack B. Thus, if B $<$-attacks $\{a, b\}$, then B normally $<$-attacks $\{a, b\}$ and $B \vdash_{\mathcal{R}} a^c$, since if B derives b^c (either by $\vdash$ or $\models$), the only other contrary, then B also derives a^c. If $B \vdash_{\mathcal{R}} b^c$, then $\{a, b\}$ reversely $<$-attacks B (since b is preferred to any assumption required to derive either a^c or b^c). By presumption, B does not derive b^c. Since B does derive a^c, B contains exactly one of x_i or $\neg x_i$ for each $1 \leq i \leq n$ (one of each is required to derive a^c, if both are present b^c is derived). This defines a truth assignment on X: true if x_i is present in B, false otherwise. Since B does not derive b^c, all bodies of rules $b^c \leftarrow neg(c_j), a^c$ for $1 \leq j \leq m$ are not satisfied (at least one element of these bodies is missing/non-derivable from B). Thus B satisfies each clause: B does not contain all elements from $neg(c_j)$ iff B satisfies at least one literal in clause c_j iff B satisfies c_j. Since this holds for all clauses, B satisfies ϕ.

For the other direction, assume that ϕ is satisfiable with satisfying truth assignment τ. Construct $B = \{x_i \mid \tau(x_i) = 1\} \cup \{\neg x_i \mid \tau(x_i) = 0\}$. Since τ assigns each variable in X to either true or false, $B \vdash_{\mathcal{R}} a^c$ (see above). Consider clause c_j. Then B does not contain all sentences of body $neg(c_j)$. Thus, $B \not\vdash_{\mathcal{R}} b^c$. This implies that $\{a, b\}$ is not admissible.

The proof for coNP-hardness holds also when adding the following rules which enforce that WCP is satisfied by the constructed ABA$^+$ instance: $x_i^c \leftarrow b, \neg x_i$ and $\neg x_i^c \leftarrow b, x_i$ for each $1 \leq i \leq n$, and for each clause $c_j = l_1 \vee l_2 \vee l_3$ adding $(l_1^c \leftarrow b, l_2, l_3)$, $(l_2^c \leftarrow b, l_1, l_3)$, and $(l_3^c \leftarrow b, l_1, l_2)$, with these new contraries. $\square$

The previous proposition suggests that, in order to verify whether A reversely $<$-counterattacks $<$-attacks from B, it is required to check for *each subset* B' of B whether B' is $<$-attacking A and, if so, whether A $<$-attacks B' (knowledge of $<$-attacks on B or any $\{b\} \subseteq B$ is not sufficient). In particular, this contrasts complexity of $<$-admissibility and $<$-stability in ABA$^+$.

Theorem 8. *Verifying whether a set of assumptions is $<$-admissible in ABA$^+$ is coNP-complete. Hardness holds even when assuming WCP.*

Verification complexity under admissibility is quite generally a cornerstone of complexity of argumentative reasoning, as a majority of the central semantics require admissibility. An algorithm that implements a non-deterministic guess of an $<$-admissible assumption set (e.g. to check credulous acceptance) requires a coNP-hard (sub)procedure to verify admissibility. This intuition leads to the next hardness result.

Theorem 9. *Checking whether a sentence is credulously accepted w.r.t. $<$-admissible semantics in ABA$^+$ is Σ_2^P-hard.*

Finally, we establish coNP-hardness for assumption-set verification for $<$-complete or $<$-grounded in ABA$^+$.

Theorem 10. *Verifying whether a set of assumptions is $<$-complete or $<$-grounded in ABA$^+$ is coNP-hard.*

4 ASP Encodings for ABA and ABA$^+$

Complementing our theoretical results, we develop answer set programming (Gelfond and Lifschitz 1988; Niemelä 1999) encodings for ABA and ABA$^+$, enabling solving σ-assumption set enumeration and acceptance problems directly using state-of-the-art ASP solvers.

Answer Set Programming. An answer set program π consists of rules r of the form $h \leftarrow b_1, \ldots, b_k, not\, b_{k+1}, \ldots, not\, b_m$, where h and each b_i is an atom. A rule is positive if $k = m$, a fact if $m = 0$, and a constraint if there is no head h. An atom b_i is a predicate $p(t_1, \ldots, t_n)$ with each t_j either a constant or a variable. An answer set program, a rule, and an atom, respectively, is ground if it is free of variables. For a non-ground program, GP is the set of rules obtained by applying all possible substitutions from the variables to the set of constants appearing in the program. An interpretation I, i.e., a subset of all the ground atoms, satisfies a positive rule $r = h \leftarrow b_1, \ldots, b_k$ iff all positive body elements $b_1, \ldots, b_k$ are in I implies that the head atom is in I. For a program π consisting only of positive rules, let $Cl(\pi)$ be the uniquely determined interpretation I that satisfies all rules in π and no subset of I satisfies all rules in π. Interpretation I is an answer set of a ground program π if $I = Cl(\pi^I)$ where $\pi^I = \{(h \leftarrow b_1, \ldots, b_k) \mid (h \leftarrow b_1, \ldots, b_k, not\, b_{k+1}, \ldots, not\, b_m) \in \pi, \{b_{k+1}, \ldots, b_m\} \cap I = \emptyset\}\}$ is the reduct; and of a non-ground program π if I is an answer set of GP of π.

We make use of specific conditional literals (Gebser et al. 2015) from the extended ASP language (`https://potassco.org/`), namely, $p(X) : q(X, Y)$ (or $p(X) : q(X)$) only with $q(X, Y)$ ($q(X)$) given as facts and the variables of predicate p being a subset of the variables of predicate q. Semantically, conditional literals simplify for our purposes to a conjunction $p(t_1), \ldots, p(t_n)$ for all groundings to variable X for $q(X, Y)$ ($q(X)$). We also use optimization statements from asprin (Brewka et al. 2015). For our purposes, a small subset of features is sufficient: we augment answer set programs with optimization statements that enforce that only answer sets that are $\subseteq$-maximal w.r.t. a specified predicate p of arity one are returned (i.e., I is an optimal answer set if there is no answer set J such that $\{p(t) \mid p(t) \in I\} \subset \{p(t) \mid p(t) \in J\}$).

ASP Encodings for ABA. We represent an ABA framework $F = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ with $\mathcal{R} = \{r_1, \ldots, r_n\}$ via the ASP facts

$$\begin{aligned}
&\{\mathbf{asm}(a). \mid a \in \mathcal{A}\} \cup \\
&\{\mathbf{head}(i, b). \mid r_i \in \mathcal{R}, b \in head(r_i)\} \cup \\
&\{\mathbf{body}(i, b). \mid r_i \in \mathcal{R}, b \in body(r_i)\} \cup \\
&\{\mathbf{contrary}(a, b). \mid b \in \overline{a}, a \in \mathcal{A}\},
\end{aligned}$$

i.e., rules are associated with a unique index, and a predicate **body** contains all body elements.

We develop ASP encodings π_σ under a semantics σ s.t. A is a σ-assumption set iff there is an answer set M of π_σ with $A = \{a \mid \mathbf{in}(a) \in M\}$. We begin with an ASP module (subprogram) π_{common} (see Listing 1) common to several semantics. The first two lines encode a non-deterministic guess of a subset of the given assumptions, with **in** and **out** denoting what is inside and outside the set, resp. The next three lines encode forward-derivations via the ASP predicate **supp** (recall that in ABA we can focus on forward-derivations only). Formally, for an assumption set A represented via **in**, if $A \vdash_{\mathcal{R}} x$, then $\mathbf{supp}(x)$ is included in an answer set. The third line encodes the base case, i.e., derivable assumptions. The fourth line encodes that whenever a rule is "triggered", i.e., all its body elements are derivable from A represented via $\mathbf{trig\_in}(R)$, then the **head** of that rule shall be derived. The fifth rule encodes triggering of rules, where we make use of the ASP conditional construct. An ABA rule in the framework (checked with usage of **head**) is triggered whenever all its body elements are supported (derivable). Intuitively, the conditional $\mathbf{supp}(X) : \mathbf{body}(R, X)$ holds if

Listing 1: Module π_{common}

$\mathbf{in}(X) \leftarrow \mathbf{asm}(X),\ not\ \mathbf{out}(X).$
$\mathbf{out}(X) \leftarrow \mathbf{asm}(X),\ not\ \mathbf{in}(X).$
$\mathbf{supp}(X) \leftarrow \mathbf{asm}(X),\ \mathbf{in}(X).$
$\mathbf{supp}(X) \leftarrow \mathbf{head}(R,X),\ \mathbf{trig\_in}(R).$
$\mathbf{trig\_in}(R) \leftarrow \mathbf{head}(R,\_),\ \mathbf{supp}(X) : \mathbf{body}(R,X).$
$\leftarrow \mathbf{in}(X),\ \mathbf{contrary}(X,Y),\ \mathbf{supp}(Y).$
$\mathbf{defeated}(X) \leftarrow \mathbf{supp}(Y),\ \mathbf{contrary}(X,Y).$

all ASP atoms $\mathbf{supp}(X)$ are present for each $\mathbf{body}(R, X)$ of the current rule index R. Analogously, one can view the ASP conditional for $R = i$ by expanding it to a list $\mathbf{supp}(x_1), \dots, \mathbf{supp}(x_n)$ if $r_i = h \leftarrow x_1, \dots, x_n$. The sixth rule ensures conflict-freeness of the guessed assumption set: whenever assumption X is included, one can derive sentence Y from a guessed assumption set, and Y is the contrary of X, we have a conflict. The last rule derives attacked (defeated) assumptions by the guessed assumption set.

With the preceding ASP rules, we encode ABA stable semantics by conjoining π_{common} with a rule that all assumptions not part of the guessed assumption set (i.e., those that are **out**) must be defeated: $\leftarrow \mathbf{out}(X), not\ \mathbf{defeated}(X)$.

A conflict-free assumption set A is admissible iff the set of assumptions B that are not attacked by A do not attack A (Proposition 1). Accordingly, in Listing 2, we check whether a contrary of an in assumption is derivable from undefeated assumptions. For π_{adm} we make use of the same scheme using conditionals as for π_{common}, except referring to different sets of assumptions (the undefeated assumptions). The ASP encoding for admissibility is $\pi_{common} \cup \pi_{adm}$.

Complete semantics is encoded by conjoining π_{common}, π_{adm} and the rule $\leftarrow \mathbf{out}(X), not\ \mathbf{att\_undef}(X)$ which encodes the check for an assumption X not present in the guessed set of assumptions A and that is not attacked by the undefeated assumptions, which implies that X is in fact defended by A. In case such an X exists, this ASP constraint ensures that such a guess does not lead to an answer set.

An admissible (complete) assumption set A is preferred if there is no superset of A that is also admissible (complete). To compute preferred assumption sets, we make use of asprin (Brewka et al. 2015) and utilize the optimization statement $\#\mathbf{preference}(\mathbf{p1}, \mathbf{superset})\ \{\mathbf{in}(\mathbf{X}) : \mathbf{asm}(\mathbf{X})\}$ that, together with $\#\mathbf{optimize}(\mathbf{p1})$, enforces that only answer sets are returned that are subset-maximal w.r.t. **in** (mirroring subset maximality)

Listing 3 gives an ASP encoding of grounded ABA semantics. We directly encode the iterative defense of assumptions, starting from the empty set. The first three rules derive all rules and sentences present in the ABA framework. The next three ASP rules encode via predicate **deriv** derivable sentences (in principle from the whole set of assumptions). In short, unlike for the other semantics, the remaining rules partition assumptions, rules, and sentences into three parts that are iteratively adapted: the set in, the set out, and elements that are neither. Interestingly, this encoding is within a tractable ASP fragment: the unique answer set of the corresponding ground program is computable in polynomial time, since default negation not in the encoding is stratifiable. Hence polynomial-time computability of the grounded assumption set (Corollary 2) is preserved under this encoding.

Listing 2: Module π_{adm}

$\mathbf{deriv\_undef}(X) \leftarrow \mathbf{asm}(X),\ not\ \mathbf{defeated}(X).$
$\mathbf{deriv\_undef}(X) \leftarrow \mathbf{head}(R,X),\ \mathbf{trig\_undef}(R).$
$\mathbf{trig\_undef}(R) \leftarrow \mathbf{head}(R,\_),\ \mathbf{deriv\_undef}(X) : \mathbf{body}(R,X).$
$\mathbf{att\_undef}(X) \leftarrow \mathbf{contrary}(X,Y),\ \mathbf{deriv\_undef}(Y).$
$\leftarrow \mathbf{in}(X),\ \mathbf{att\_undef}(X).$

Listing 3: Module π_{grd}

$\mathbf{rule}(R) \leftarrow \mathbf{head}(R,\_).$
$\mathbf{sentence}(S) \leftarrow \mathbf{head}(\_,S).$
$\mathbf{sentence}(S) \leftarrow \mathbf{contrary}(\_,S),\ not\ \mathbf{asm}(S).$
$\mathbf{deriv}(X) \leftarrow \mathbf{asm}(X).$
$\mathbf{deriv}(X) \leftarrow \mathbf{head}(R,X),\ \mathbf{deriv\_rule}(R).$
$\mathbf{deriv\_rule}(R) \leftarrow \mathbf{head}(R,\_),\ \mathbf{deriv}(X) : \mathbf{body}(R,X).$
$\mathbf{in}(X) \leftarrow \mathbf{asm}(X),\ \mathbf{out}(Y) : \mathbf{contrary}(X,Y).$
$\mathbf{in}(R) \leftarrow \mathbf{rule}(R),\ \mathbf{in}(X) : \mathbf{body}(R,X).$
$\mathbf{in}(S) \leftarrow \mathbf{in}(R),\ \mathbf{head}(R,S).$
$\mathbf{out}(X) \leftarrow \mathbf{in}(Y),\ \mathbf{contrary}(X,Y).$
$\mathbf{out}(R) \leftarrow \mathbf{out}(X),\ \mathbf{body}(R,X).$
$\mathbf{out}(S) \leftarrow \mathbf{sentence}(S),\ \mathbf{out}(R) : \mathbf{head}(R,S).$
$\mathbf{out}(S) \leftarrow \mathbf{sentence}(S),\ not\ \mathbf{deriv}(S).$

ASP Encoding for ABA$^+$ Stable Semantics. A conflict-free assumption set A is $<$-stb iff each assumption $b \in \mathcal{A}$ that is not $<$-normally attacked by A is either in A or reversely $<$-attacked by A (Prop. 4). By Lemma 3, one can check if a set of assumptions is $<$-stb via only using forward-derivability. Thus, we can implement $<$-stb semantics via forward-derivability (as for ABA above): perform an ASP guess of a set, check conflict-freeness via the corresponding rules in π_{common}, compute each $\{b\}$ normally $<$-attacked (see also proof of Prop. 5), and from the remaining assumptions $\{c\}$ not normally $<$-attacked whether they are reversely $<$-attacked. The encoding is independent of whether the given framework satisfies WCP or not.

5 Experiments

We present empirical results comparing the performance of the state-of-the-art ASP solver Clingo on our ASP encodings to currently available systems for ABA and ABA$^+$. For ABA, abagraph implements the so-called dispute derivation approach (Craven and Toni 2016) using Prolog, while aba2af translates the ABA to abstract argumentation frameworks and uses AF-level ASP encodings to solve the ABA reasoning problem (Lehtonen, Wallner, and Järvisalo 2017). For ABA$^+$, ABAplus (Bao, Cyras, and Toni 2017) translates to the AF-level and employs the Aspartix ASP encodings (Egly, Gaggl, and Woltran 2010) to perform ABA$^+$ reasoning. We note that abagraph only supports the admissible and grounded semantics, while aba2af supports admissible, stable and preferred. We used Clingo v5.2.2 (Gebser et al. 2016) as the ASP solver, and SICStus Prolog v4.4.1 for 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 time limit per instance.

We used the ABA frameworks (which contain up to 90 sentences) and queries used by Craven and Toni (2016) and Lehtonen, Wallner, and Järvisalo (2017) in experiments on abagraph and aba2af (`http://robertcraven.org/proarg/experiments.html`), and also followed their setup when applicable, considering the enumeration of all solutions w.r.t. a query sentence for ABA adm, and skeptical acceptance of a query for ABA stb. Within the capabilities of the systems under comparison, we consider credulous acceptance for ABA grd, and enumeration of all solutions (no query) for ABA prf and ABA$^+$ $<$-stb. Following (Lehtonen, Wallner, and Järvisalo 2017), for ABA adm we filtered out trivial instances. For ABA$^+$, the ABAplus algorithm requires WCP (recall Def. 5), and enforces WCP if not satisfied by changing the input framework, which can be time-consuming. Hence for a fair comparison on ABA$^+$, we generated smaller instances following (Craven and Toni 2016) over 10, 14, 18, 22, 26, 30 sentences with 10 frameworks per number of sentences and two preference relations per framework, and used the frameworks as modified by ABAplus as input to both ABAplus and our approach. We generated preferences by choosing a random permutation $(a_i)_{0<i\leq n}$ of the assumptions, and for each $j < i$, set a_i to be preferred to a_j with two fixed probabilities, 15% and 40%.

Table 1 overviews the results. ASP, without any timeouts and very small cumulative runtimes, clearly outperforms the other systems on each problem (and supports more problem types than the competing ABA systems). In contrast, the other systems exhibit high numbers of timeouts, especially on the ABA problems. Our approach benefits from not constructing an AF unlike ABAplus and aba2af, for which the AF translation can often be a main bottleneck.

We studied the scalability of our approach by generating larger instances in the style of series 4 benchmarks in (Craven and Toni 2016) (the other series of (Craven and Toni 2016) gave either very easy instances or fixed sets of assumptions). In particular, we generated three frameworks for each number of sentences, with up to 4000 sentences and rule heads, 37% assumptions, and rules per head and body lengths, resp., randomly chosen within $[1, \min(n_s/7, 20)]$ and $[1, \min(n_s/8, 20)]$ with number of sentences n_s. The results are shown in Table 2 for credulous reasoning on 10 arbitrary queries per ABA framework under adm, com, and stb; for assumption set enumeration under prf (as asprin does not directly support queries); and for ABA$^+$ credulous reasoning under $<$-stb without enforcing WCP. Here we can routinely solve instances with up to 3000 sentences for ABA and up to 1000 for ABA$^+$. Within ABA, for prf the limit is somewhat lower due to both computational complexity and the task, but nevertheless significantly higher than for the other systems; even under adm abagraph could only solve the 50-sentence instances and none of the larger ones.

Table 1: Runtime comparison. Mean (**mean**), median (**med.**) and cumulative running times (**cum.**) over solved instances, **#to** is the number of timeouts. Number of instances: 1466 (adm), 6709 (stb, grd), 680 (prf), 120 (ABA$^+$ $<$-stb).

			Running times (s)		
Problem	**Approach**	**#to**	**mean**	**med.**	**cum.**
ABA adm, *enum. w/query*	ASP	**0**	**0.020**	**0.008**	**29**
	abagraph	394	18.500	1.175	19832
	aba2af	402	15.357	1.390	16341
ABA stb, *skept. accep.*	ASP	**0**	**0.009**	**0.008**	**59**
	aba2af	648	7.246	0.900	43918
ABA grd *cred. accep.*	ASP	**0**	**0.020**	**0.012**	**29**
	abagraph	202	10.254	0.972	12961
ABA prf *enum. wo/query*	ASP	**0**	**0.333**	**0.328**	**226**
	aba2af	255	6.082	0.464	2585
ABA$^+$ $<$-stb *enum. wo/query*	ASP	**0**	**0.018**	**0.008**	**2**
	ABAplus	9	15.583	0.268	1729

Table 2: Scalability of ASP on larger frameworks.

	#timeouts (mean running time over solved (s))				
$\lvert\mathcal{L}\rvert$	**ABA** adm	**ABA** com	**ABA** stb	**ABA** prf	**ABA**$^+$ $<$-stb
50	0 (0.02)	0 (0.01)	0 (0.01)	0 (0.3)	0 (0.1)
250	0 (0.4)	0 (0.4)	0 (0.3)	0 (1.5)	0 (12.6)
500	0 (0.9)	0 (0.9)	0 (0.6)	0 (2.7)	0 (53.1)
1000	0 (4.4)	0 (3.7)	0 (1.7)	0 (13.1)	0 (241.0)
1500	0 (28.1)	0 (20.2)	0 (5.5)	0 (107.9)	60 (0.0)
2000	0 (38.5)	0 (30.7)	0 (7.1)	0 (125.0)	60 (0.0)
2500	0 (165.4)	0 (124.3)	0 (30.4)	2 (141.9)	60 (0.0)
3000	24 (331.3)	23 (283.5)	4 (268.4)	3 (0.0)	60 (0.0)
3500	28 (225.7)	27 (262.1)	19 (182.8)	3 (0.0)	60 (0.0)
4000	28 (44.6)	28 (35.2)	28 (14.2)	3 (0.0)	60 (0.0)

6 Conclusions

We proposed a new approach to reasoning in assumption-based argumentation with and without preferences via non-trivial ASP encodings of ABA reasoning tasks under several central argumentation semantics. Our approach extends and significantly improves on the empirical performance of the current state-of-the-art approaches to ABA reasoning. This motivates further study of alternative encodings based on e.g. the labelling-based view (Sakama and Rienstra 2017; Schulz and Toni 2017). Towards bridging the gap between the current knowledge on complexity of reasoning in ABA and ABA$^+$, we provided complexity lower bounds for ABA$^+$ reasoning under several semantics. While credulous acceptance in ABA and ABA$^+$ have the same complexity under the stable semantics, our results on the complexity of the verification task for ABA$^+$ strongly suggests that the integration of preferential information into ABA may increase the computational complexity of acceptance problems.

Acknowledgements

Work supported in part by Academy of Finland (276412 and 312662) and Austrian Science Fund (FWF): P30168, I2854.

References

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

Bao, Z.; Cyras, K.; and Toni, F. 2017. ABAplus: Attack reversal in abstract and structured argumentation with preferences. In *Proc. PRIMA*, volume 10621 of *LNCS*, 420–437. Springer.

Besnard, P., and Hunter, A. 2008. *Elements of Argumentation*. MIT Press.

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

Bondarenko, A.; Dung, P. M.; Kowalski, R. A.; and Toni, F. 1997. An abstract, argumentation-theoretic approach to default reasoning. *Artif. Intell.* 93:63–101.

Brewka, G.; Delgrande, J. P.; Romero, J.; and Schaub, T. 2015. asprin: Customizing answer set preferences without a headache. In *Proc. AAAI*, 1467–1474. AAAI Press.

Caminada, M., and Schulz, C. 2017. On the equivalence between assumption-based argumentation and logic programming. *J. Artif. Intell. Res.* 60:779–825.

Caminada, M.; Sá, S.; Alcântara, J.; and Dvořák, W. 2013. On the difference between assumption-based argumentation and abstract argumentation. In *Proc. BNAIC*, 25–32.

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

Craven, R., and Toni, F. 2016. Argument graphs and assumption-based argumentation. *Artif. Intell.* 233:1–59.

Craven, R.; Toni, F.; Cadar, C.; Hadad, A.; and Williams, M. 2012. Efficient argumentation for medical decision-making. In *Proc. KR*, 598–602. AAAI Press.

Craven, R.; Toni, F.; and Williams, M. 2013. Graph-based dispute derivations in assumption-based argumentation. In *TAFA 2013 Revised Selected Papers*, volume 8306 of *LNCS*, 46–62. Springer.

Cyras, K., and Toni, F. 2016a. ABA+: assumption-based argumentation with preferences. In *Proc. KR*, 553–556. AAAI Press.

Cyras, K., and Toni, F. 2016b. ABA+: assumption-based argumentation with preferences. *CoRR* abs/1610.03024.

Cyras, K., and Toni, F. 2016c. Properties of ABA+ for non-monotonic reasoning. In *Proc. NMR*.

Cyras, K.; Fan, X.; Schulz, C.; and Toni, F. 2018. Assumption-based argumentation: Disputes, explanations, preferences. In *Handbook of Formal Argumentation*. College Publications. chapter 7, 365–408.

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

Dung, P. M.; Kowalski, R. A.; and Toni, F. 2006. Dialectic proof procedures for assumption-based, admissible argumentation. *Artif. Intell.* 170(2):114–159.

Dung, P. M.; Kowalski, R. A.; and Toni, F. 2009. Assumption-based argumentation. In *Argumentation in Artificial Intelligence*. Springer. 25–44.

Dung, P. M.; Mancarella, P.; and Toni, F. 2007. Computing ideal sceptical argumentation. *Artif. Intell.* 171(10-15):642–674.

Dung, P. M.; Toni, F.; and Mancarella, P. 2010. Some design guidelines for practical argumentation systems. In *Proc. COMMA*, volume 216 of *FAIA*, 183–194. IOS Press.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2):321–358.

Egly, U., and Woltran, S. 2006. Reasoning in argumentation frameworks using quantified Boolean formulas. In *Proc. COMMA*, volume 144 of *FAIA*, 133–144. IOS Press.

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

Fan, X.; Toni, F.; Mocanu, A.; and Williams, M. 2014. Dialogical two-agent decision making with assumption-based argumentation. In *Proc. AAMAS*, 533–540. IFAAMAS/ACM.

Gaertner, D., and Toni, F. 2007a. CaSAPI: a system for credulous and sceptical argumentation. In *Proc. NMR*, 80–95.

Gaertner, D., and Toni, F. 2007b. Computing arguments and attacks in assumption-based argumentation. *IEEE Intelligent Systems* 22(6):24–33.

Gaertner, D., and Toni, F. 2008. Hybrid argumentation and its properties. In *Proc. COMMA*, volume 172 of *FAIA*, 183–195. IOS Press.

García, A. J., and Simari, G. R. 2004. Defeasible logic programming: An argumentative approach. *Theor. Pract. Log. Prog.* 4(1-2):95–138.

Gebser, M.; Harrison, A.; Kaminski, R.; Lifschitz, V.; and Schaub, T. 2015. Abstract gringo. *Theor. Pract. Log. Prog.* 15(4-5):449–463.

Gebser, M.; Kaminski, R.; Kaufmann, B.; Ostrowski, M.; Schaub, T.; and Wanko, P. 2016. Theory solving made easy with Clingo 5. In *Technical Communications of ICLP*, OASICS, 2:1–2:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Proc. ICLP/SLP*, 1070–1080. MIT Press.

Lehtonen, T.; Wallner, J. P.; and Järvisalo, M. 2017. From structured to abstract argumentation: Assumption-based acceptance via AF reasoning. In *Proc. ECSQARU*, volume 10369 of *LNCS*, 57–68. Springer.

Matt, P.; Toni, F.; Stournaras, T.; and Dimitrelos, D. 2008. Argumentation-based agents for eprocurement. In *Proc. AAMAS Industry and Applications Track*, 71–74. IFAAMAS.

Niemelä, I. 1999. Logic programs with stable model semantics as a constraint programming paradigm. *Ann. Math. Artif. Intell.* 25(3-4):241–273.

Prakken, H. 2010. An abstract framework for argumentation with structured arguments. *Argument & Computation* 1(2):93–124.

Sakama, C., and Rienstra, T. 2017. Representing argumentation frameworks in answer set programming. *Fundam. Inform.* 155(3):261–292.

Schulz, C., and Toni, F. 2017. Labellings for assumption-based and abstract argumentation. *Int. J. Approx. Reasoning* 84:110–149.

Toni, F., and Sergot, M. 2011. Argumentation and answer set programming. In *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning: Essays Dedicated to Michael Gelfond on the Occasion of His 65th Birthday*. Springer. 164–180.

Toni, F. 2013. A generalised framework for dispute derivations in assumption-based argumentation. *Artif. Intell.* 195:1–43.

Toni, F. 2014. A tutorial on assumption-based argumentation. *Argument & Computation* 5(1):89–117.

Wakaki, T. 2017. Assumption-based argumentation equipped with preferences and its application to decision making, practical reasoning, and epistemic reasoning. *Comput. Intell.* 33(4):706–736.