

ABApplus: Attack Reversal in Abstract and Structured Argumentation with Preferences

Ziyi Bao, Kristijonas Čyras, and Francesca Toni

Department of Computing, Imperial College London, UK
{ziyi.bao14, k.cyras13, ft}@imperial.ac.uk

Abstract. We present ABApplus, a system that implements reasoning with the argumentation formalism ABA^+ . ABA^+ is a structured argumentation (SA) formalism that extends Assumption-Based Argumentation (ABA) with preferences and accounts for preferences via attack reversal. ABA^+ also admits as instances Preference-based Argumentation Frameworks which account for preferences by reversing attacks in abstract argumentation (AA). ABApplus readily implements attack reversal in both AA and ABA-style SA. ABApplus affords computation, visualisation and comparison of extensions under five argumentation semantics. It is available both as a stand-alone system and as a web application.

1 Introduction

Approaches to preferences in abstract argumentation (AA) [8] and structured argumentation (SA) [3] can be roughly classified as follows: 1. *discarding* attacks from attackers that are less preferred than attackees (see e.g. [1] for AA and e.g. ASPIC⁺ [14,15] for SA); 2. *reversing* attacks from attackers that are less preferred than attackees (see Preference-based Argumentation Frameworks (PAFs) [2] for AA and Assumption-Based Argumentation with Preferences (ABA^+) [6] for SA); 3. comparing extensions by aggregating preferences over their elements (see e.g. [2] for AA and e.g. [18] for SA); 4. incorporating numerical weights of arguments or attacks into the definition of semantics (see e.g. [5] for AA and e.g. [11] for SA). Implementations of several approaches in classes 1. and 4. exist (see Section 6), but, to the best of our knowledge, implementations of approaches in classes 2. and 3. are lacking. In this paper, we present an implementation of approaches in class 2., i.e. implementation of *attack reversal* in both AA and SA with preferences.

Our system, *ABApplus*, implements reasoning with the recently proposed formalism ABA^+ . ABA^+ extends Assumption-Based Argumentation (ABA) with preferences and is the only SA formalism (to the best of our knowledge) to reverse attacks in SA due to preferences. To implement attack reversal in SA, ABApplus uses a semantics-preserving mapping from ABA^+ to AA and employs an off-the-shelf AA implementation, namely ASPARTIX [9], for determining extensions. To this end, we advance a new mapping from ABA^+ frameworks to AA frameworks, which we call *assumption graphs*. In addition, we identify a novel

property of ABA^+ frameworks, called *Weak Contraposition*, which distinguishes a class of ABA^+ frameworks that allow for semantic correspondence with their assumption graphs. Subject to Weak Contraposition, assumption graphs guarantee a correct representation and implementation of ABA^+ (as well as ABA) under five semantics, and allow $ABAplus$ to provide concise graphical visualisation and comparison of ABA^+ (as well as ABA) frameworks.

To implement attack reversal in AA , $ABAplus$ relies on a semantics-preserving mapping from PAFs into ABA^+ . To this end, we consider a simple mapping from PAFs to ABA^+ frameworks, showing that ABA^+ admits PAFs as instances. Thus, $ABAplus$ readily implements attack reversal in AA with preferences too.

$ABAplus$ is freely available at github.com/kcyras/ABAplus as a stand-alone system, and at www-abaplus.doc.ic.ac.uk as a web application.

2 Background

ABA^+ We base the background on ABA^+ on [6].

An ABA^+ framework is a tuple $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq)$, where:

- $(\mathcal{L}, \mathcal{R})$ is a deductive system with \mathcal{L} a language and \mathcal{R} a set of rules of the form $\varphi_0 \leftarrow \varphi_1, \dots, \varphi_m$ with $m \geq 0$ and $\varphi_i \in \mathcal{L}$ for $i \in \{0, \dots, m\}$; φ_0 is the *head* and $\varphi_1, \dots, \varphi_m$ the *body* of the rule; if $m = 0$, then $\varphi_0 \leftarrow \varphi_1, \dots, \varphi_m$ has an empty body, and is written as $\varphi_0 \leftarrow \top$, where $\top \notin \mathcal{L}$;
- $\mathcal{A} \subseteq \mathcal{L}$ is a non-empty set of *assumptions*;
- $\bar{\cdot} : \mathcal{A} \rightarrow \mathcal{L}$ is a total map: for $\mathbf{a} \in \mathcal{A}$, $\bar{\mathbf{a}}$ is referred to as the *contrary* of \mathbf{a} ;
- \leq is a preorder (i.e. reflexive and transitive) on \mathcal{A} , called a *preference relation*.

As usual, the strict (asymmetric) counterpart $<$ of \leq is given by $\alpha < \beta$ iff $\alpha \leq \beta$ and $\beta \not\leq \alpha$, for any α and β . (We assume this for all preorders in this paper.) For assumptions $\mathbf{a}, \mathbf{b} \in \mathcal{A}$, $\mathbf{a} \leq \mathbf{b}$ means that \mathbf{b} is at least as preferred as \mathbf{a} , and $\mathbf{a} < \mathbf{b}$ means that \mathbf{a} is strictly less preferred than \mathbf{b} .

Assumptions in ABA^+ model defeasible information. For instance, assumptions can represent beliefs of an agent. In such a case, preferences in ABA^+ can be seen to represent the relative degrees of belief.

Example 1 (Preferences over beliefs). At a party, Zed is having a discussion about the outcome of a possible referendum in the Netherlands on whether to remain in the EU. Two of his interlocutors, Ann and Bob, have diverging views on the outcome of the referendum. Ann claims that the Dutch would vote to leave, whereas Bob maintains that they would vote to stay. Suppose Zed knows that Ann likes big claims based on dubious assumptions, so he trusts Bob more than Ann. This preference information should conceivably lead Zed to accepting Bob's argument, rather than Ann's.

We model Zed's knowledge as an ABA^+ framework $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq)$ with

- $\mathcal{L} = \{\mathbf{a}, \mathbf{b}, \textit{leave}, \textit{stay}\}$,
- $\mathcal{R} = \{\textit{leave} \leftarrow \mathbf{a}, \textit{stay} \leftarrow \mathbf{b}\}$,
- $\mathcal{A} = \{\mathbf{a}, \mathbf{b}\}$,
- $\bar{\mathbf{a}} = \textit{stay}, \bar{\mathbf{b}} = \textit{leave}$,

– $a \leq b$, $b \not\leq a$.¹

Here, assumptions a and b stand for believing in Ann and Bob, respectively. Rules $leave \leftarrow a$ and $stay \leftarrow b$ represent the statements of Zed’s interlocutors: for instance, $leave \leftarrow a$ represents that if Zed were to believe in Ann, the outcome of the referendum would be the Dutch leaving the EU. The contraries indicate which information is conflicting: for instance, the contrary of b being $leave$ models that the Dutch leaving the EU— $leave$ —conflicts with believing in Bob— b .² The degree of Zed’s beliefs is represented through the preference $a < b$ (i.e. $a \leq b$, $b \not\leq a$ and nothing else holds), which means that Zed trusts Ann strictly less than he trusts Zed.

Throughout the paper, we assume as given a fixed but otherwise arbitrary ABA^+ framework $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq)$, unless specified otherwise.

We next give notions of arguments (as deduction trees) and attacks in ABA^+ .

An *argument for* $\varphi \in \mathcal{L}$ *supported by* $A \subseteq \mathcal{A}$ *and* $R \subseteq \mathcal{R}$, denoted $A \vdash^R \varphi$, is a finite tree with: the root labelled by φ ; leaves labelled by \top or assumptions, with A being the set of all such assumptions; the children of non-leaves ψ labelled by the elements of the body of some ψ -headed rule in \mathcal{R} , with R being the set of all such rules. $A \vdash \varphi$ is a shorthand for an argument $A \vdash^R \varphi$ with some $R \subseteq \mathcal{R}$.

Let $A, B \subseteq \mathcal{A}$. Then $A \subseteq \mathcal{A} <-attacks B \subseteq \mathcal{A}$, denoted $A \rightsquigarrow_{<} B$, iff

- either there is an argument $A' \vdash \bar{b}$, for some $b \in B$, supported by $A' \subseteq A$, and $\nexists a' \in A'$ with $a' < b$;
- or there is an argument $B' \vdash \bar{a}$, for some $a \in A$, supported by $B' \subseteq B$, and $\exists b' \in B'$ with $b' < a$.

We call an $<$ -attack formed as in the first bullet point above a *normal attack*, and an $<$ -attack formed as in the second bullet point above a *reverse attack*. If it is not the case that $A <-attacks B$, we may write $A \not\rightsquigarrow_{<} B$. (We will adopt an analogous convention for other attack relations in this paper.)

To illustrate, in \mathcal{F} from Example 1, $\{a\}$ ‘tries’ to attack $\{b\}$, but is prevented by the preference $a < b$. Instead, $\{b\} <-attacks \{a\}$ (and also $\{a, b\}$) via reverse attack. Likewise, $\{a, b\} <-attacks$ both itself and $\{a\}$ via reverse attack.

We next give auxiliary notions that will be used to define ABA^+ semantics.

Let $A \subseteq \mathcal{A}$. The *conclusions of* A is $Cn(A) = \{\varphi \in \mathcal{L} : \exists A' \vdash \varphi, A' \subseteq A\}$. We say that A is *closed* if $A = Cn(A) \cap \mathcal{A}$. We say that \mathcal{F} is *flat* if every $A \subseteq \mathcal{A}$ is closed. We assume ABA^+ frameworks to be flat, unless specified otherwise.

Note that \mathcal{F} from Example 1 is flat: no assumption can be deduced from the empty set of assumptions, so $Cn(\emptyset) = \emptyset$; the only assumptions deducible from $\{a\}$ and $\{b\}$ are a and b , respectively, so both $\{a\}$ and $\{b\}$ are closed; clearly, \mathcal{A} is closed; hence, all sets of assumptions are closed.

¹ As a preorder, \leq has to be reflexive, but for brevity purposes we often omit to specify the reflexive instances of any preorder.

² Other ways of formalising such examples in ABA are possible; we chose a natural and simple representation. Generally, knowledge representation in argumentation is an open problem, discussion of which is beyond the scope of this paper.

Further, for $A \subseteq \mathcal{A}$, we say that: A is *<-conflict-free* if $A \not\prec_{<} A$; also, A *<-defends* $A' \subseteq \mathcal{A}$ if for all $B \subseteq \mathcal{A}$ with $B \rightsquigarrow_{<} A'$ it holds that $A \rightsquigarrow_{<} B$; and A is *<-admissible* if it is <-conflict-free and <-defends itself.

We consider the following five ABA⁺ semantics. A set $E \subseteq \mathcal{A}$, also called an *extension*, is:

- *<-complete* if E is <-admissible and contains every set of assumptions it <-defends;
- *<-preferred* if E is \subseteq -maximally <-admissible;
- *<-stable* if E is <-conflict-free and for all $\mathbf{b} \in \mathcal{A} \setminus E$ it holds that $E \rightsquigarrow_{<} \{\mathbf{b}\}$;
- *<-ideal* if E is \subseteq -maximal among sets of assumptions that are <-admissible and contained in all <-preferred sets of assumptions;
- *<-grounded* if E is a \subseteq -minimal <-complete set of assumptions.

Throughout the paper, $\sigma \in \{\text{grounded, ideal, stable, preferred, complete}\}$, and we assume that $<-\sigma$ denotes a specified ABA⁺ semantics.

To illustrate with Example 1, it is easy to see that $\{\mathbf{b}\}$ is a unique $<-\sigma$ extension of \mathcal{F} , leading Zed to accept Bob's argument, just as intended.

Note well that ABA⁺ conservatively extends ABA in that when preferences are absent, ABA⁺ frameworks behave exactly like ABA frameworks [6]. Therefore, our implementation of ABA⁺ will be an implementation of ABA too.

Abstract Argumentation (AA) We base the background on AA on [8].

An *AA framework* is a pair $(Args, \rightsquigarrow)$ with a set $Args$ of arguments and a binary attack relation \rightsquigarrow on $Args$. Notions of conflict-freeness, defence and admissibility, as well as semantics of σ extensions, are defined verbatim as for ABA⁺, but with sets of arguments replacing sets of assumptions.

Preference-based Argumentation Frameworks (PAFs) We base the background on PAFs on [2].

A *Preference-based Argumentation Framework* (PAF) is a tuple $(Args, \rightsquigarrow, \preceq)$, where $(Args, \rightsquigarrow)$ is an AA framework and \preceq is a preorder over $Args$. Given a PAF $(Args, \rightsquigarrow, \preceq)$, its *repaired framework* is an AA framework $(Args, \hookrightarrow)$ such that for $\mathbf{a}, \mathbf{b} \in Args$, $\mathbf{a} \hookrightarrow \mathbf{b}$ iff

- either $\mathbf{a} \rightsquigarrow \mathbf{b}$ and $\mathbf{a} \not\prec \mathbf{b}$,
- or $\mathbf{b} \rightsquigarrow \mathbf{a}$ and $\mathbf{b} \prec \mathbf{a}$.

The attack formed as in the second bullet pertains to attack reversal in AA.

From now on, unless specified otherwise, we assume a PAF $(Args, \rightsquigarrow, \preceq)$ as given, and that for $(Args, \rightsquigarrow, \preceq)$, its repaired framework is denoted by $(Args, \hookrightarrow)$.

Semantics of PAFs is defined via the semantics of their repaired frameworks: $E \subseteq Args$ is a σ extension of $(Args, \rightsquigarrow, \preceq)$ iff E is a σ extension of $(Args, \hookrightarrow)$.

Example 2. Zed's knowledge as in Example 1 can be modelled in PAFs as follows. The arguments in $Args = \{\mathbf{a}, \mathbf{b}\}$ represent the statements of Zed's interlocutors, and the attacks $\mathbf{a} \rightsquigarrow \mathbf{b}$, $\mathbf{b} \rightsquigarrow \mathbf{a}$ represent the conflict between the two statements. The preference of Bob's argument over Ann's argument is expressed by $\mathbf{a} \prec \mathbf{b}$. For the resulting PAF $(Args, \rightsquigarrow, \preceq)$, the attack $\mathbf{a} \rightsquigarrow \mathbf{b}$ is reversed in $(Args, \hookrightarrow)$ to yield only $\mathbf{b} \hookrightarrow \mathbf{a}$. So $(Args, \rightsquigarrow, \preceq)$ has a unique σ extension $\{\mathbf{b}\}$.

3 Implementing Attack Reversal in ABA^+

The idea behind implementing attack reversal in ABA^+ is to use a mapping from ABA^+ to AA that preserves semantic correspondence, and then use an off-the-shelf AA solver, particularly ASPARTIX, to compute extensions of ABA^+ frameworks by computing extensions of the corresponding AA frameworks. In this section, we provide one such mapping.

3.1 Assumption Graphs

Given an ABA^+ framework, we construct its *assumption graph*—an AA framework with arguments being either singleton sets of assumptions, or sets of assumptions supporting (ABA^+) arguments for contraries of assumptions, and with the attack relation being $\rightsquigarrow_{<}$ restricted to those arguments, as follows.

Definition 1. Let \mathcal{D} be the collection of sets of assumptions that support arguments for contraries of assumptions, i.e. $\mathcal{D} = \{S \subseteq \mathcal{A} : S \vdash \bar{a}, a \in \mathcal{A}\}$. The **assumption graph** of \mathcal{F} is an AA framework $\mathcal{G} = (\text{Args}, \hookrightarrow)$ with

- $\text{Args} = \mathcal{D} \cup \{\{a\} : a \in \mathcal{A}\}$,
- $\hookrightarrow = \rightsquigarrow_{<} \cap (\text{Args} \times \text{Args})$,

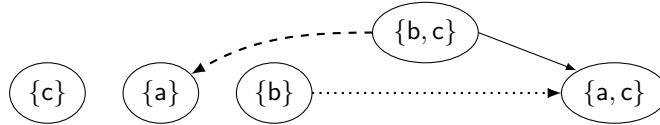
where $\rightsquigarrow_{<}$ is the $<$ -attack relation of \mathcal{F} .

Example 3. Consider an ABA^+ framework $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq)$ with

- $\mathcal{L} = \{a, b, c, e, d, \bar{c}\}$,
- $\mathcal{R} = \{d \leftarrow a, c, e \leftarrow b, c\}$,
- $\mathcal{A} = \{a, b, c\}$,
- $\bar{a} = e, \bar{b} = d$,
- $a < b$ (i.e. \leq is a preorder with $a \leq b, b \not\leq a$).

In \mathcal{F} , $\{b, c\}$ supports an argument for the contrary e of a , and no assumption in $\{b, c\}$ is strictly less preferred than a . Thus, $\{b, c\}$ $<$ -attacks $\{a\}$, as well as any set containing a , via normal attack. On the other hand, $\{a, c\}$ (supporting an argument for the contrary d of b) is prevented from $<$ -attacking $\{b\}$, due to the preference $a < b$. Instead $\{b\}$, as well as any set containing b , $<$ -attacks $\{a, c\}$ via reverse attack. Overall, \mathcal{F} has a unique $<$ - σ extension, namely $E = \{b, c\}$, with conclusions $Cn(E) = \{b, c, e\}$.

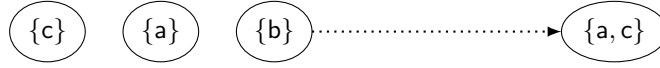
The assumption graph $\mathcal{G} = (\text{Args}, \hookrightarrow)$ of \mathcal{F} has the set of arguments $\mathcal{D} = \{\{a\}, \{b\}, \{c\}, \{b, c\}, \{a, c\}\}$ with attacks $\{b, c\} \hookrightarrow \{a\}$, $\{b, c\} \hookrightarrow \{a, c\}$, $\{b\} \hookrightarrow \{a, c\}$, and is depicted below (here and henceforth, dashed arrows indicate normal attacks, dotted arrows indicate reverse attacks and solid arrows indicate $<$ -attacks that are both normal and reverse).



$(\text{Args}, \hookrightarrow)$ has a unique σ extension $\mathcal{E} = \{\{b\}, \{c\}, \{b, c\}\}$. Note: $\bigcup \mathcal{E} = E$.

We would like establish semantic correspondence between ABA⁺ frameworks and their assumptions graphs as in Example 3: $\mathcal{E} \subseteq \text{Args}$ is a σ extension of $(\text{Args}, \hookrightarrow)$ iff $E = \bigcup \mathcal{E}$ is a $<$ - σ extension of \mathcal{F} . However, this is not possible in general, as the following example shows.

Example 4. Modify the ABA⁺ framework \mathcal{F} from Example 3 by removing the rule $e \leftarrow \mathbf{b}, \mathbf{c}$ from \mathcal{R} to obtain the ABA⁺ framework $\mathcal{F}' = (\mathcal{L}, \mathcal{R}', \mathcal{A}, \bar{\cdot}, \leq)$ with $\mathcal{R}' = \mathcal{R} \setminus \{e \leftarrow \mathbf{b}, \mathbf{c}\}$. Observe that in \mathcal{F}' , all the singletons $\{\mathbf{a}\}$, $\{\mathbf{b}\}$ and $\{\mathbf{c}\}$ are $<$ -unattacked. Hence, all these singletons are $<$ -defended by any set. However, $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ is not $<$ -conflict-free, as $\{\mathbf{a}, \mathbf{c}\} \vdash \bar{\mathbf{b}}$. Consequently, no set $A \subseteq \mathcal{A}$ can contain all sets of assumptions it $<$ -defends and be $<$ -conflict-free at the same time. Thus, \mathcal{F}' has no $<$ -complete extensions. Meanwhile, the assumption graph \mathcal{G}' of \mathcal{F}' (depicted below) has a unique complete extension $\{\{\mathbf{a}\}, \{\mathbf{b}\}, \{\mathbf{c}\}\}$.



In the next section, we identify a property of ABA⁺ frameworks, called the Axiom of Weak Contraposition, satisfaction of which allows to preserve semantic correspondence between ABA⁺ frameworks and their assumption graphs.

3.2 Weak Contraposition

The following axiom concerns contrapositive reasoning as understood in classical logic and is (strictly) weaker than *contraposition* as defined for ASPIC⁺ in [15].

Axiom. $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq)$ satisfies the **Axiom of Weak Contraposition (WCP)** (for short) just in case for all $A \subseteq \mathcal{A}$, $R \subseteq \mathcal{R}$ and $\mathbf{b} \in \mathcal{A}$ it holds that

if $A \vdash \bar{\mathbf{b}}$ and there exists $\mathbf{a}' \in A$ such that $\mathbf{a}' < \mathbf{b}$,

then, for some $\mathbf{a} \in A$ which is \leq -minimal such that $\mathbf{a} < \mathbf{b}$, there is $A_{\mathbf{a}} \vdash \bar{\mathbf{a}}$, for some $A_{\mathbf{a}} \subseteq (A \setminus \{\mathbf{a}\}) \cup \{\mathbf{b}\}$.

This axiom insists on contrapositive reasoning when an argument involves assumptions less preferred than the one whose contrary it supports.

Note well that ABA⁺ frameworks with empty preferences (which can be seen as ABA frameworks) satisfy WCP trivially.

As an illustration, consider \mathcal{F} from Example 3. We find $\{\mathbf{a}, \mathbf{c}\} \vdash \bar{\mathbf{b}}$ and $\mathbf{a} < \mathbf{b}$, where \mathbf{a} is \leq -minimal in $\{\mathbf{a}, \mathbf{c}\}$ with $\mathbf{a} < \mathbf{b}$. We also find $\{\mathbf{b}, \mathbf{c}\} \vdash \bar{\mathbf{a}}$, where $\{\mathbf{b}, \mathbf{c}\} = (\{\mathbf{a}, \mathbf{c}\} \setminus \{\mathbf{a}\}) \cup \{\mathbf{b}\}$. It is thus easy to see that \mathcal{F} satisfies WCP. By contrast, in \mathcal{F}' from Example 4, we find $\{\mathbf{a}, \mathbf{c}\} \vdash \bar{\mathbf{b}}$ and $\mathbf{a} < \mathbf{b}$, but there is no $A_{\mathbf{a}} \vdash \bar{\mathbf{a}}$ with $A_{\mathbf{a}} \subseteq (\{\mathbf{a}, \mathbf{c}\} \setminus \{\mathbf{a}\}) \cup \{\mathbf{b}\} = \{\mathbf{b}, \mathbf{c}\}$; hence, \mathcal{F}' violates WCP.

We next show that ABA⁺ frameworks satisfying WCP are in semantic correspondence with their assumption graphs, in the following sense. (We omit lengthy proofs for space reasons.)

Theorem 1. Suppose $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq)$ satisfies WCP and let $\mathcal{G} = (\text{Args}, \leftrightarrow)$ be the assumption graph of \mathcal{F} .

- If $E \subseteq \mathcal{A}$ is a $<-\sigma$ extension of \mathcal{F} , then $\{S \in \mathcal{D} : S \subseteq E\} \cup \{\{a\} : a \in E\}$ is a σ extension of \mathcal{G} ;
- If $\mathcal{E} \subseteq \text{Args}$ is a σ extension of \mathcal{G} , then $\bigcup \mathcal{E}$ is a $<-\sigma$ extension of \mathcal{F} .

For illustration, we saw in Example 3 that the assumption graph \mathcal{G} of \mathcal{F} has a unique σ extension $\mathcal{E} = \{\{b\}, \{c\}, \{b, c\}\}$, and $E = \bigcup \mathcal{E} = \{b, c\}$ is a unique $<-\sigma$ extension of \mathcal{F} . We also saw that WCP is necessary in Theorem 1: in Example 4, the assumption graph \mathcal{G}' of \mathcal{F}' has a unique complete extension $\{\{a\}, \{b\}, \{c\}\}$, while \mathcal{F}' has no $<-\sigma$ complete extensions.

Theorem 1 provides a theoretical underpinning for the ABAPlus system which we will describe in Section 5. Before that, we discuss how an ABA^+ framework that violates WCP to begin with, can be modified *by adding new rules* so as to satisfy WCP. We call this *enforcing* WCP. A situation where an argument satisfies the antecedent of WCP while the consequent is false is called *instance* of WCP. More formally:

Definition 2. $A \vdash \bar{b}$ is an *instance of WCP* just in case

- $\exists a' \in A$ such that $a' < b$, and
- for no $a \in A$ which is \leq -minimal such that $a < b$ there is $A_a \vdash \bar{a}$, for some $A_a \subseteq (A \setminus \{a\}) \cup \{b\}$.

As an illustration, in \mathcal{F}' from Example 4, $\{a, c\} \vdash d$ is an instance of WCP.

For WCP to be satisfied, it suffices, for every instance of WCP, to ensure *one* additional argument for the contrary of *some single* \leq -minimal assumption among those less preferred than the one whose contrary an instance of WCP is an argument for. We call any such \leq -minimal assumption a *witness*:

Definition 3. Let $A \vdash \bar{b}$ be an instance of WCP and let $a \in A$. Then a is a *witness* to $A \vdash \bar{b}$ just in case a is \leq -minimal such that $a < b$.

So, in \mathcal{F}' from Example 4, a is a witness to the instance of WCP $\{a, c\} \vdash d$.

A witness to an instance of WCP can be seen as a candidate assumption with regards to which an additional argument is needed in order to satisfy WCP. This can be achieved by adding *enforcing rules*, defined as follows.

Definition 4. Let $A \vdash \bar{b}$ be an instance of WCP with a witness $a \in A$. Say $A = \{a_1, \dots, a_n\}$, where $a = a_i$ for some i . The *enforcing rule*, denoted by $\bar{a} \leftarrow A \setminus a, b$, is the rule $\bar{a} \leftarrow a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n, b$.

In Example 4, for the only instance of WCP $\{a, c\} \vdash d$ and its sole witness a , the enforcing rule is $e \leftarrow b, c$.

WCP can be enforced by adding an enforcing rule for every instance of WCP and its witness, as shown next.

Theorem 2. Let $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq)$ be an ABA^+ framework, let

$\mathcal{R}_{A \vdash \bar{b}} = \{\bar{a} \leftarrow A \setminus a, b \text{ is an enforcing rule} : A \vdash \bar{b} \text{ is an instance of WCP (in } (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq) \text{ with a witness } a \in A)\}$

be the set of enforcing rules for an instance $A \vdash \bar{b}$ of WCP in $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq)$, and let f be a function, defined for finite non-empty sets, that selects any one element from a given set. The ABA^+ framework $(\mathcal{L}, \mathcal{R} \cup \mathcal{R}', \mathcal{A}, \bar{\cdot}, \leq)$, where

$$\mathcal{R}' = \{f(\mathcal{R}_{A \vdash \bar{b}}) : A \vdash \bar{b} \text{ is an instance of WCP (in } (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq))\},$$

satisfies WCP.

Proof. Let $A \vdash \bar{b}$ be an instance of WCP (in $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq)$) with a witness $a \in A$. The rule $\bar{a} \leftarrow A \setminus a, b$ guarantees that in $(\mathcal{L}, \mathcal{R} \cup \mathcal{R}', \mathcal{A}, \bar{\cdot}, \leq)$ we can find the argument $(A \setminus \{a\}) \cup \{b\} \vdash^{\{\bar{a} \leftarrow A \setminus a, b\}} \bar{a}$. Note that no such argument can result in an instance of WCP (in $(\mathcal{L}, \mathcal{R} \cup \mathcal{R}', \mathcal{A}, \bar{\cdot}, \leq)$), precisely because the witness a is \leq -minimal. Therefore, $(\mathcal{L}, \mathcal{R} \cup \mathcal{R}', \mathcal{A}, \bar{\cdot}, \leq)$ satisfies WCP.

For illustration, to enforce WCP on \mathcal{F}' from Example 4, add the enforcing rule $e \leftarrow b, c$ to \mathcal{R}' to obtain \mathcal{F} from Example 3, which satisfies WCP.

Several remarks regarding WCP are in place.

First, note well that using WCP *does not* amount to ‘making attacks symmetric’. Indeed, consider $A \rightsquigarrow \{b\}$ and suppose that $A \vdash \bar{b}$ is an instance of WCP with a witness $a' \in A$ such that $a' < b$. Making this attack symmetric means imposing $\{b\} \rightsquigarrow_{<} A$. However, WCP *does not* require $\{b\} \rightsquigarrow_{<} A$. Instead, WCP requires that $(A \setminus \{a'\}) \cup \{b\} \vdash \bar{a}'$, which in general amounts to $(A \setminus \{a'\}) \cup \{b\} \rightsquigarrow_{<} \{a'\}$ (and hence $(A \setminus \{a'\}) \cup \{b\} \rightsquigarrow_{<} A$).

Second, the \leq -minimality of a witness assumption in enforcing WCP plays a crucial role. In particular, it saves from generating redundant arguments when enforcing the axiom. For instance, consider $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq)$ with $\mathcal{R} = \{\bar{c} \leftarrow a, b\}$, $\mathcal{A} = \{a, b, c\}$ and $a < b < c$.³ The argument $\{a, b\} \vdash \bar{c}$ is an instance of WCP. If \leq -minimality were not required in the conditions of the consequent of WCP, one could end up choosing b and adding the rule $\bar{b} \leftarrow a, c$ to \mathcal{R} so as to generate the argument $\{a, c\} \vdash \bar{b}$ in $\mathcal{F}' = (\mathcal{L}, \mathcal{R} \cup \{\bar{b} \leftarrow a, c\}, \mathcal{A}, \bar{\cdot}, \leq)$. This would result in the argument $\{a, b\} \vdash \bar{c}$ not being an instance of WCP in \mathcal{F}' .

However, $\{a, c\} \vdash \bar{b}$ would then also make the antecedent of the WCP true (because $a < b$) while keeping the consequent false, thus yielding an instance of WCP in \mathcal{F}' . Consequently, to enforce WCP, one would additionally need to ensure existence of yet another argument, for example, $\{b, c\} \vdash \bar{a}$, by, for instance, adding the rule $\bar{a} \leftarrow b, c$. By contrast, choosing a (necessarily \leq -minimal) witness to begin with, in this case the only one such being a , enables one to add a single rule, say the enforcing rule $\bar{a} \leftarrow b, c$, so as to generate the argument $\{b, c\} \vdash \bar{a}$ in $\mathcal{F}'' = (\mathcal{L}, \mathcal{R} \cup \{\bar{a} \leftarrow b, c\}, \mathcal{A}, \bar{\cdot}, \leq)$. Thus, the instance of WCP in question is eliminated in \mathcal{F}'' , no further instances of WCP are obtained in \mathcal{F}'' , and so WCP is enforced on \mathcal{F} by modifying it into \mathcal{F}'' .

³ Unless specified otherwise, we omit \mathcal{L} and $\bar{\cdot}$, and adopt the following conventions: unless \bar{x} appears in either \mathcal{A} or \mathcal{R} , it is different from the sentences appearing in \mathcal{A} or \mathcal{R} ; thus, \mathcal{L} consists of all the sentences appearing in \mathcal{R} , \mathcal{A} and $\{\bar{a} : a \in \mathcal{A}\}$.

The third remark concerns other ways to enforce WCP on a given ABA^+ framework. For example, given an instance of WCP $\mathcal{A} \vdash \bar{b}$ with a witness $a \in \mathcal{A}$, one could add the rule $\bar{a} \leftarrow \top$ to obtain the argument $\emptyset \vdash_{\{\bar{a} \leftarrow \top\}} \bar{a}$ as required to eliminate the instance of WCP in question, at the same time avoiding to create additional instances. This particular way seems rather ad hoc and also quite radical with respect to knowledge representation: it seems unintuitive to have assumptions immediately ‘rejected’ (by arguing for their contraries from the empty set) just because they are involved in the argumentative process of arguing for contraries of more preferred assumptions. In contrast, enforcing WCP as in Theorem 2 is the least restrictive way that both ensures satisfaction of WCP and leaves the user the option to restrict the knowledge further, if needed.

Finally, observe that, technically, imposing WCP on a flat ABA^+ framework may yield a non-flat ABA^+ framework. Indeed, consider $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq)$ with $\mathcal{L} = \{a, b, c, x, y\}$, $\mathcal{R} = \{x \leftarrow a\}$, $\mathcal{A} = \{a, b, c\}$, $\bar{a} = c$, $\bar{b} = x$, $\bar{c} = y$, $a < b$. We find $\{a\} \vdash b$ and $a < b$, so to modify the framework so that it satisfies WCP, we need an argument $S \vdash \bar{a}$, for some $S \subseteq \{b\}$. Whatever the support S , we get an argument $S \vdash c$, whence S is not closed.

This behaviour can be easily avoided. Indeed, instead of defining the contrary mapping $\bar{\cdot} : \mathcal{A} \rightarrow \mathcal{L}$ to map assumptions into the elements of the language, we can assign new symbols for contraries of assumptions, while retaining the same behaviour (semantically) of ABA^+ frameworks: for each assumption a we take a new symbol a^c not in \mathcal{L} , and define the contrary mapping \mathcal{C} so that $\mathcal{C}(a) = a^c$; then, for any intended contrary x of a in \mathcal{L} , we add a rule $a^c \leftarrow x$. We omit the details due to lack of space, and assume, without loss of generality, that enforcing WCP on an arbitrary flat ABA^+ framework always yields a flat ABA^+ framework that satisfies WCP.

4 Implementing Attack Reversal in PAFs

In this section, following the way ABA admits AA as an instance [17], we show how ABA^+ admits PAFs as instances. To instantiate an ABA^+ framework with $(Args, \rightsquigarrow, \preceq)$, we map each argument $a \in Args$ into an assumption $a \in \mathcal{A}$, together with a new symbol \bar{a} for the contrary, map each attack $a \rightsquigarrow b$ into a rule $\bar{b} \leftarrow a$, and transfer the preference ordering \preceq to constitute \leq , as follows.

Definition 5. *Given a PAF $(Args, \rightsquigarrow, \preceq)$, an ABA^+ framework corresponding to $(Args, \rightsquigarrow, \preceq)$ is $\mathcal{F}_{PAF} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \leq)$ with:*

- $\mathcal{L} = Args \cup \{\bar{a} : a \in Args, \bar{a} \notin Args\}$;
- $\mathcal{R} = \{\bar{b} \leftarrow a : a \rightsquigarrow b\}$;
- $\mathcal{A} = Args$;
- for $a \in \mathcal{A}$, \bar{a} is the contrary of a ;
- $\leq = \preceq$.

Note that \mathcal{F} from Example 1 is an ABA^+ framework corresponding to the PAF $(Args, \rightsquigarrow, \preceq)$ from Example 2.

Henceforth, \mathcal{F}_{PAF} is an ABA^+ framework corresponding to $(Args, \rightsquigarrow, \preceq)$.

Note that \mathcal{F}_{PAF} is flat. Moreover, since all (ABA^+) arguments for contraries in \mathcal{F}_{PAF} are supported by singleton sets, due to attack reversal attacks in $(\text{Args}, \rightsquigarrow, \preceq)$ coincide with $<$ -attacks in \mathcal{F}_{PAF} , in the sense that for $\mathbf{a}, \mathbf{b} \in \text{Args}$, $\mathbf{a} \hookrightarrow \mathbf{b}$ iff $\{\mathbf{a}\} \rightsquigarrow_{<} \{\mathbf{b}\}$. We thus obtain the following correspondence result, which says that, under any semantics σ , every PAF is an instance of ABA^+ .

Theorem 3. *$E \subseteq \text{Args}$ is a σ extension of $(\text{Args}, \rightsquigarrow, \preceq)$ iff E is a $<$ - σ extension of \mathcal{F}_{PAF} .*

To illustrate Theorem 3, $(\text{Args}, \rightsquigarrow, \preceq)$ from Example 2 has a unique σ extension $\{\mathbf{b}\}$, and this is precisely the unique $<$ - σ extension of the corresponding ABA^+ framework \mathcal{F} from Example 1.

Observe that \mathcal{F}_{PAF} need not satisfy WCP. Nonetheless, given that all (ABA^+) arguments for contraries in \mathcal{F}_{PAF} are supported by singleton sets, every instance of WCP $\{\mathbf{a}\} \vdash \bar{\mathbf{b}}$ (with $\mathbf{a} < \mathbf{b}$) has a unique witness \mathbf{a} , so that enforcing WCP on \mathcal{F}_{PAF} as in Theorem 2 yields a unique ABA^+ framework $\mathcal{F}'_{\text{PAF}}$ which has the same $<$ -attack relation as \mathcal{F}_{PAF} : for $A, B \subseteq \mathcal{A}$, $A \rightsquigarrow_{<} B$ iff $A \rightsquigarrow'_{<} B$, where $\rightsquigarrow_{<}$ and $\rightsquigarrow'_{<}$ are $<$ -attack relations of \mathcal{F}_{PAF} and $\mathcal{F}'_{\text{PAF}}$, respectively. Hence, determining extensions of $\mathcal{F}'_{\text{PAF}}$ with WCP enforced amounts to determining extensions of \mathcal{F}_{PAF} , and hence of $(\text{Args}, \rightsquigarrow, \preceq)$. Thus, by implementing ABA^+ , the ABAPlus system described in the next section can readily compute σ extensions of PAFs.

5 ABAPlus

ABAPlus, both a stand-alone system and a web application, implements reasoning in (flat) ABA^+ (and its instances, including PAFs) subject to WCP. In this section, we describe and illustrate ABAPlus. First, we describe the web application as well as the back-end of ABAPlus. Then, we illustrate the use of ABAPlus (as well as ABA^+) with a pair of examples that show how preferences over goals and rules can be accommodated in ABA^+ through preferences over assumptions.

5.1 System Description

To compute extensions of $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\cdot}, \preceq)$, ABAPlus feeds the assumption graph $\mathcal{G} = (\text{Args}, \hookrightarrow)$ (Definition 1) of \mathcal{F} into ASPARTIX to compute extensions of \mathcal{G} and maps the extensions obtained to extensions of \mathcal{F} . Such strategy is sound and complete, given that extensions of \mathcal{F} and \mathcal{G} are in one-to-one correspondence (Theorem 1), as long as ASPARTIX correctly computes the extensions of AA frameworks under any semantics σ . The following is a summary of how both the stand-alone system and the web application of ABAPlus work.

Web Application The web application `www-abaplus.doc.ic.ac.uk` takes a single ABA^+ framework \mathcal{F} as input in the Prolog-like format:

- `myAsm(a)`. specifies that \mathbf{a} is an assumption from \mathcal{A} ;
- `contrary(a, x)`. specifies that $x \in \mathcal{L}$ is the contrary $\bar{\mathbf{a}}$ of assumption \mathbf{a} ;

- `myRule(h, [b1, ..., bn])`. specifies that $h \leftarrow b_1, \dots, b_n$ is a rule from \mathcal{R} ;
- `myPrefLE(b, a)`. specifies, for assumptions \mathbf{a}, \mathbf{b} , that $\mathbf{b} \leq \mathbf{a}$;
- `myPrefLT(b, a)`. specifies, for assumptions \mathbf{a}, \mathbf{b} , that $\mathbf{b} < \mathbf{a}$.

The input can either be entered in a textbox or uploaded as a file (.pl extension). Upon loading an ABA⁺ framework, its assumption graph is computed and visualised as follows. 1. Nodes hold sets of assumptions. 2. Arrows represent attacks: dashed arrows for normal attacks, dotted arrows for reverse attacks, solid arrows for $<$ -attacks that are both normal and reverse. 3. An extension under any semantics can be selected to highlight. 4. A second graph can be displayed with another extension (under any semantics) highlighted for comparison.

Back-end ABAPlus back-end parses the Prolog-like representation of \mathcal{F} into Python format. The next steps are as follows.

1. The input \mathcal{F} is pre-processed: (a) checking if \mathcal{F} is flat; (b) calculating and updating \mathcal{F} with the transitive closure of \leq ; (c) checking if the strict counterpart $<$ of (the updated) \leq is asymmetric; (d) computing (ABA⁺) arguments for contraries of assumptions (bottom-up) to check whether \mathcal{F} satisfies WCP, and if not, enforcing WCP as in Theorem 2.
2. Generation of *Args* goes thus (a top-down recursive procedure is used to find the sentences that could label argument trees until assumptions are found):
 - (a) For every assumption $\mathbf{a} \in \mathcal{A}$, store $\{\mathbf{a}\}$ in *Args*;
 - (b) For every assumption $\mathbf{a} \in \mathcal{A}$, generate all arguments for $\bar{\mathbf{a}}$ and store the supports of those arguments in *Args*.
3. Generation of \hookrightarrow goes thus:
 - (a) For every $\mathbf{a} \in \mathcal{A}$, for every $B \vdash \bar{\mathbf{a}}$, check if $\exists \mathbf{b} \in B$ such that $\mathbf{b} < \mathbf{a}$: (i) if no, store $B \hookrightarrow \{\mathbf{a}\}$; (ii) if yes, store $\{\mathbf{a}\} \hookrightarrow B$;
 - (b) For any $A', B' \in \textit{Args}$ such that $\{\mathbf{a}\} \subseteq A'$ and $B \subseteq B'$, (i) if $B \hookrightarrow \{\mathbf{a}\}$, store $B' \hookrightarrow A'$; (ii) if $\{\mathbf{a}\} \hookrightarrow B$, store $A' \hookrightarrow B'$.
4. The assumption graph $\mathcal{G} = (\textit{Args}, \hookrightarrow)$ thus constructed is fed to ASPARTIX (using clingo and DLV ASP solvers), represented via Prolog-like sentences:
 - `arg(A)`. represents an argument $A \in \textit{Args}$;
 - `att(A, B)`. represents an attack $A \hookrightarrow B$.
5. For every semantics $\sigma \in \{\text{grounded, ideal, stable, preferred, complete}\}$, σ extensions of \mathcal{G} are computed.
6. Each σ extension \mathcal{E} of \mathcal{G} is unpacked into a $<$ - σ extension $E = \bigcup \mathcal{E}$ of \mathcal{F} .

Tools ABAPlus uses the following tools: Python 3.4.3; Unicorn 19.6; Clingo 4.5.4; DLV (“deductive database system”) version 17/12/2012; encodings of semantics (stable.dl, ideal.dl, comp.dl, prefix_gringo.lp, ground.dl) from ASPARTIX system page; D3 (graph visualisation) 3.5.17.

5.2 Examples

In this section, we exemplify the use of ABA⁺ and ABAPlus in two different scenarios. Specifically, in addition to having illustrated in Example 1 how pref-

erences over beliefs are accommodated in ABA^+ , we now consider how preferences over assumptions in ABA^+ can be used to model preferences over goals and rules. This illustrates how ABAPlus supports computations pertaining to reasoning with preferences not only over beliefs and abstract arguments (by indirectly implementing PAFs), but also over goals and rules.

Preferences over goals express that certain goals are more desirable to be achieved in a particular situation. Goals in ABA^+ can be represented via assumptions, whence preferences over assumptions represent preferences over goals.

Example 5 (Preferences over goals). Consider ABAPlus as a system system to schedule meetings. Imagine a user who needs to schedule a meeting on one of two suggested time slots, t_1 and t_2 . Suppose that t_1 is the time when the user usually has lunch, and that t_2 covers the user’s standard coffee break. The user prefers to have lunch as usual over scheduling the meeting, but also deems the meeting to be more important than having coffee.

In ABA^+ , we can represent the situation as follows. Let m , l and c be assumptions standing for having the meeting, lunch and coffee, respectively. Further, let t_1 and t_2 be assumptions standing for the two time slots in question. The rules $\bar{t}_1 \leftarrow l$ and $\bar{t}_2 \leftarrow c$ express that having lunch and coffee as usual make the two respective time slots unavailable. Additionally, the rule $\bar{m} \leftarrow \bar{t}_1, \bar{t}_2$ expresses that the meeting will not be scheduled if none of the time slots are available. Finally, user’s preferences are expressed by letting $c < m < l$.

The resulting ABA^+ framework, call it \mathcal{F} , can be input into ABAPlus via the following specification:

$\text{myAsm}(c)$.	$\text{contrary}(c, \bar{c})$.	$\text{myRule}(\bar{t}_1, [l])$.
$\text{myAsm}(l)$.	$\text{contrary}(l, \bar{l})$.	$\text{myRule}(\bar{t}_2, [c])$.
$\text{myAsm}(m)$.	$\text{contrary}(m, \bar{m})$.	$\text{myRule}(\bar{m}, [\bar{t}_1, \bar{t}_2])$.
$\text{myAsm}(t_1)$.	$\text{contrary}(t_1, \bar{t}_1)$.	$\text{myPrefLT}(c, m)$.
$\text{myAsm}(t_2)$.	$\text{contrary}(t_2, \bar{t}_2)$.	$\text{myPrefLT}(m, l)$.

Given this input, ABAPlus recognizes that \mathcal{F} does not satisfy WCP: there is $\{c, l\} \vdash \bar{m}$ with $c < m$, but there is no argument for \bar{c} at all. Thus, ABAPlus informs the user of this, and proposes to automatically enforce WCP on \mathcal{F} . ABAPlus enforces WCP by adding the rule $\bar{c} \leftarrow m, l$, which expresses that having lunch and the meeting prevents having coffee. This results into a new framework, call it \mathcal{F}' . ABAPlus then determines that \mathcal{F}' has a unique σ extension $\{m, l, t_2\}$, with conclusions $\{m, l, t_2, \bar{t}_1, \bar{c}\}$, which indicate that the meeting should be scheduled at time t_2 , at the expense of having coffee. The outcome of feeding \mathcal{F} into ABAPlus is depicted in the screenshot in Figure 1.

Preferences over rules indicate which rules should be followed in case application of multiple rules is impossible. In ABA^+ , references over rules can be expressed by adding new assumptions that stand for the applicability of the rules, and by imposing preferences over those assumptions.

Example 6 (Preferences over rules). Consider two general rules regarding healthy living: ‘if you can afford, you should follow a healthy diet’ and ‘if you can afford,

you should exercise regularly’. These two rules can be represented in ABA^+ as $d \leftarrow c$ and $e \leftarrow c$, where e and d stand for exercising and diet, respectively, and the sentence c stands for affordability. Suppose that you are not able to both eat healthily and exercise regularly, as these habits require more time than you can afford. This can be considered as a constraint and modelled via the rule $\bar{c} \leftarrow d, e$.

Suppose that a certain authority declares that exercising regularly is more important than eating healthily. Thus, a preference of the second rule over the first rule can be formed, and given that you cannot follow both rules, you should prefer exercising regularly. In ABA^+ this preference can be modelled as follows. First, add new assumptions a and b representing the applicability of the two rules. Second, modify $e \leftarrow c$ and $d \leftarrow c$ into new rules $d \leftarrow a, c$ and $e \leftarrow b, c$, respectively. Finally, add the preference $a < b$ over the new assumptions.

Overall, the ABA^+ framework representing this information is like the framework \mathcal{F} from Example 3, but with the additional rule $\bar{c} \leftarrow d, e$. $ABAplus$ determines that it satisfies WCP and has a unique $<-\sigma$ extension $\{b, c\}$, with conclusions $Cn(\{b, c\}) = \{b, c, e\}$, which suggest exercising regularly.

6 Related Work

We discuss argumentation systems that account for preferences of some form. Note that none of them implement attack reversal in argumentation.

TOAST [16] is a web application (toast.arg-tech.org) implementing (an early version [15] of) $ASPIC^+$. *TOAST* computes and visualises $ASPIC^+$ arguments, attacks and extensions (under four semantics). In contrast to *ABAplus*, *TOAST* accommodates preferences over defeasible rules, but not over premises, even though $ASPIC^+$ allows preferences over premises. *TOAST* lifts preferences from defeasible rules to arguments, whence attacks from less preferred arguments are *discarded*, rather than reversed. A feature of *TOAST* is the ability to *transpose* rules, which is related (but not equivalent) to contraposition. *ABAplus* instead enforces WCP, and by a simple modification can deal with contraposition too.

Gorgias-B (gorgiasb.tuc.gr/index.html) is a stand-alone system implementing *Gorgias* [13]—an argumentation formalism based on logic programming without negation as failure [7], combining preferences and abduction. Given an application scenario, *Gorgias-B* guides the user through a decision problem by incremental refinements, where the user is presented with several (usually conflicting) alternatives (i.e. arguments, which amount to sets of rules) and is asked for preference information in order to determine which attacks succeed: a variant of *discarding* attacks is employed. Reasoning outcomes are evaluated essentially via preferred semantics, in contrast to multiple semantics available in *ABAplus*.

Gorgias-B asks the user to input preferences on the go whenever needed to solve conflicts, whereas *ABAplus* takes user information at once and provides reasoning outcomes, without the need for the user to specify any further information. Nonetheless, it may be a useful feature of *ABAplus* to be able to query the user for preferences. We leave this for future work.

DeLPclient (lidia.cs.uns.edu.ar/delp_client) is a web application implementing reasoning in Defeasible Logic Programming (DeLP) [10]. It allows to specify logic programs with strict and defeasible rules, and preferences over the latter, which are accounted by *discarding* attacks. Given a program, DeLPclient answers queries and can also provide explanations of the answers in terms of arguments and counter-arguments for the warrant status of the query. We will explore whether explanations could be implemented in ABAPlus.

Carneades [12] is a web application (carneades.fokus.fraunhofer.de/carneades) and a stand-alone system implementing the argumentation formalism of the same name [11]. Carneades supports weights on arguments (which are instantiations of argumentation schemes), and employs proofs standards and weighting functions to *balance* arguments and evaluate their acceptance via grounded semantics. Carneades also visualises argument graphs and indicates structural links within.

ConArg [4] is a web application (hdmi.unipg.it/conarg) and a stand-alone system implementing *Weighted Argumentation Frameworks* (WAFs) [5]. ConArg allows for specifying (via graphical interface) WAFs—AA frameworks with weights on attacks—and computes their extensions under various semantics. Weights on attacks are accounted by specifying *budgets* of how much conflict (within extensions) can be tolerated and defence can be relaxed.

7 Conclusions

We presented the system ABAPlus that implements ABA^+ (and by extension, ABA), a formalism of structured argumentation with preferences. ABAPlus implements attack reversal in ABA^+ as well as its instances, particularly Preference-based Argumentation Frameworks (PAFs). More specifically, ABAPlus applies a new principle of Weak Contraposition (WCP) on flat ABA^+ frameworks, computes their extensions, visualises and allows for juxtaposing their assumption graphs. The theoretical backbone of the system is a semantics-preserving mapping from ABA^+ to abstract argumentation (AA), which allows to use off-the-shelf AA solvers (particularly, ASPARTIX) to determine extensions of ABA^+ (as well as ABA) frameworks. ABAPlus is a freely available stand-alone system and a web application.

We aim to analyse the scalability and performance of ABAPlus and to use it in applications of reasoning with preferences. It would also be interesting to find other classes of ABA^+ frameworks (like those satisfying WCP) and/or mappings to e.g. AA that allow to determine ABA^+ extensions via AA or other formalisms (e.g. answer set programming). Implementing non-flat ABA^+ frameworks is a future research direction too. In addition to studying whether features of some other systems implementing argumentative reasoning with preferences can be of use in ABAPlus (as discussed in Section 6), we plan to implement the following features: a) relaxed syntactic requirements for input; b) saving ABA^+ frameworks on the server; c) query-based interface and computations; d) interactive graphical representations; e) improved session data handling.

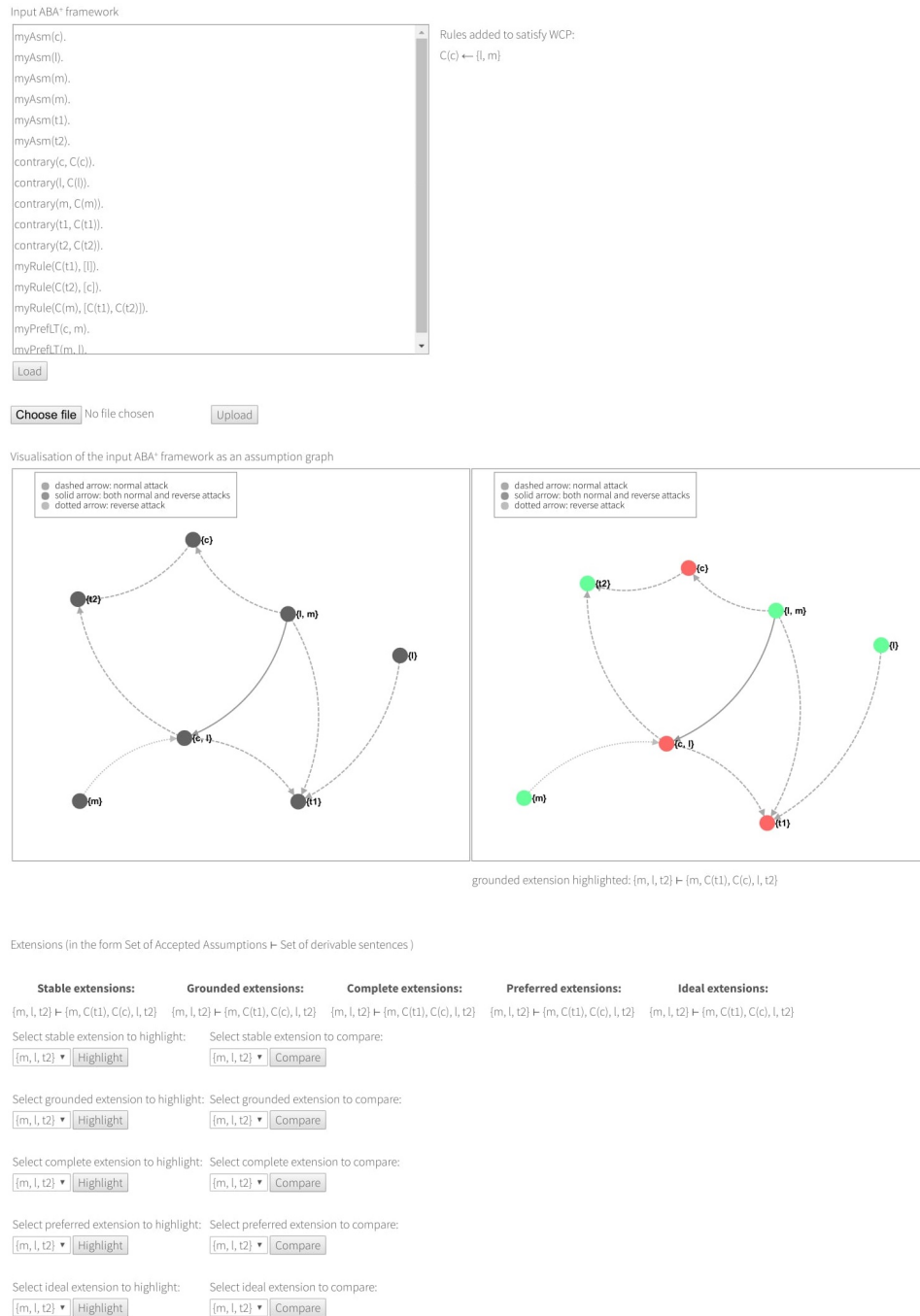


Fig. 1. Screenshot of ABAPlus outcome for ABA⁺ framework \mathcal{F} from Example 5

References

1. Amgoud, L., Cayrol, C.: A Reasoning Model Based on the Production of Acceptable Arguments. *Ann. Math. Artif. Intell.* 34(1-3), 197–215 (2002)
2. Amgoud, L., Vesic, S.: Rich Preference-Based Argumentation Frameworks. *Int. J. Approx. Reason.* 55(2), 585–606 (2014)
3. Besnard, P., García, A.J., Hunter, A., Modgil, S., Prakken, H., Simari, G.R., Toni, F.: Introduction to Structured Argumentation. *Arg. & Comp.* 5(1), 1–4 (2014)
4. Bistarelli, S., Rossi, F., Santini, F.: ConArg: A Tool for Classical and Weighted Argumentation. In: Baroni, P., Gordon, T., Scheffler, T., Stede, M. (eds.) *COMMA. Frontiers in Artificial Intelligence and Applications*, vol. 287, pp. 463–464. IOS Press, Potsdam (2016)
5. Bistarelli, S., Santini, F.: A Common Computational Framework for Semiring-based Argumentation System. In: Coelho, H., Studer, R., Wooldridge, M. (eds.) *ECAI. Frontiers in Artificial Intelligence and Applications*, vol. 215, pp. 131–136. IOS Press, Lisbon (2010)
6. Čyras, K., Toni, F.: ABA+: Assumption-Based Argumentation with Preferences. In: Baral, C., Delgrande, J.P., Wolter, F. (eds.) *KR*. pp. 553–556. AAAI Press, Cape Town (2016)
7. Dimopoulos, Y., Kakas, A.C.: Logic Programming without Negation as Failure. In: Lloyd, J.W. (ed.) *ISLP*. pp. 369–383. MIT Press, Portland (1995)
8. Dung, P.M.: On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-person Games. *Artif. Intell.* 77, 321–357 (1995)
9. Gaggl, S.A., Manthey, N., Ronca, A., Wallner, J.P., Woltran, S.: Improved Answer-Set Programming Encodings for Abstract Argumentation. *Theory Pract. Log. Program.* 15(4-5), 434–448 (2015)
10. García, A.J., Simari, G.R.: Defeasible Logic Programming: DeLP-servers, Contextual Queries, and Explanations for Answers. *Arg. & Comp.* 5(1), 63–88 (2014)
11. Gordon, T.F., Prakken, H., Walton, D.: The Carneades Model of Argument and Burden of Proof. *Artif. Intell.* 171(10-15), 875–896 (2007)
12. Gordon, T.F., Walton, D.: Formalizing Balancing Arguments. In: Baroni, P., Gordon, T.F., Scheffler, T., Stede, M. (eds.) *COMMA. Frontiers in Artificial Intelligence and Applications*, vol. 287, pp. 327–338. IOS Press, Potsdam (2016)
13. Kakas, A.C., Moraitis, P.: Argumentation Based Decision Making for Autonomous Agents. In: *AAMAS*. pp. 883–890. ACM Press, Melbourne (2003)
14. Modgil, S., Prakken, H.: A General Account of Argumentation with Preferences. *Artif. Intell.* 195, 361–397 (2013)
15. Prakken, H.: An Abstract Framework for Argumentation with Structured Arguments. *Arg. & Comp.* 1(2), 93–124 (2010)
16. Snaith, M., Reed, C.: TOAST: Online ASPIC+ Implementation. In: Verheij, B., Szeider, S., Woltran, S. (eds.) *COMMA. Frontiers in Artificial Intelligence and Applications*, vol. 245, pp. 509–510. IOS Press, Vienna (2012)
17. Toni, F.: Reasoning on the Web with Assumption-Based Argumentation. In: Eiter, T., Krennwallner, T. (eds.) *Reasoning Web (Summer School). Lecture Notes in Computer Science*, vol. 7487, pp. 370–386. Springer, Vienna (2012)
18. Wakaki, T.: Assumption-Based Argumentation Equipped with Preferences. In: Dam, H.K., Pitt, J.V., Xu, Y., Governatori, G., Ito, T. (eds.) *PRIMA. Lecture Notes in Computer Science*, vol. 8861, pp. 116–132. Springer, Gold Coast (2014)