ABAplus: Attack Reversal in Abstract and Structured Argumentation with Preferences

Ziyi Bao, Kristijonas Čyras(✉), and Francesca Toni

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

Abstract. We present ABAplus, a system that implements reasoning with the argumentation formalism ABA^+. ABA^+ is a structured argumentation formalism that extends Assumption-Based Argumentation (ABA) with preferences and accounts for preferences via attack reversal. ABA^+ also admits as instance Preference-based Argumentation which accounts for preferences by reversing attacks in abstract argumentation (AA). ABAplus readily implements attack reversal in both AA and ABA-style structured argumentation. ABAplus 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) [9] and structured argumentation [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 $ASPIC^+$ [16,17] for structured argumentation); 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^+) [7] for structured argumentation); 3. comparing extensions by aggregating preferences over their elements (see e.g. [2] for AA and [21] for structured argumentation); 4. incorporating numerical weights of arguments or attacks into the definition of semantics (see e.g. [5] for AA and [12] for structured argumentation). Implementations of several approaches in classes 1. and 4. exist (see Sect. 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 structured argumentation with preferences.

Our system, *ABAplus*, implements reasoning with the recently proposed formalism ABA^+ [7]. ABA^+ extends Assumption-Based Argumentation (ABA) [6,20] with preferences and is the only structured argumentation formalism (to the best of our knowledge) to reverse attacks in structured argumentation due to preferences. To implement attack reversal in structured argumentation, ABAplus uses a semantics-preserving mapping from ABA^+ to AA and employs an off-the-shelf AA implementation, namely ASPARTIX [10], for determining extensions.

B. An et al. (Eds.): PRIMA 2017, LNAI 10621, pp. 420–437, 2017.
https://doi.org/10.1007/978-3-319-69131-2_25

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 in 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 are used in 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 https://github.com/kcyras/ABAplus as a stand-alone system, and at http://www-abaplus.doc.ic.ac.uk as a web application.

2 Background

ABA$^+$. We base the background on ABA$^+$ on [7].

An *ABA$^+$ framework* is a tuple $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}, \leqslant)$, where:

- $(\mathcal{L}, \mathcal{R})$ is a deductive system with $\mathcal{L}$ a language and $\mathcal{R}$ a set of rules of the form $\varphi_0 \leftarrow \varphi_1, \ldots, \varphi_m$ with $m \geqslant 0$ and $\varphi_i \in \mathcal{L}$ for $i \in \{0, \ldots, m\}$; φ_0 is the *head* and $\varphi_1, \ldots, \varphi_m$ the *body* of the rule; if $m = 0$, then $\varphi_0 \leftarrow \varphi_1, \ldots, \varphi_m$ 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*;
- $\overline{} : \mathcal{A} \to \mathcal{L}$ is a total map: for $\mathsf{a} \in \mathcal{A}$, $\overline{\mathsf{a}}$ is referred to as the *contrary* of a;
- $\leqslant$ is a preorder (i.e. reflexive and transitive) on $\mathcal{A}$, called a *preference relation*.

As usual, the strict (asymmetric) counterpart $<$ of $\leqslant$ is given by $\alpha < \beta$ iff $\alpha \leqslant \beta$ and $\beta \nleqslant \alpha$, for any α and β. (We assume this for all preorders in this paper.) For assumptions $\mathsf{a}, \mathsf{b} \in \mathcal{A}$, $\mathsf{a} \leqslant \mathsf{b}$ means that b is at least as preferred as a, and $\mathsf{a} < \mathsf{b}$ means that a is strictly less preferred than 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 $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}, \leqslant)$ with

– $\mathcal{L} = \{\mathsf{a}, \mathsf{b}, \mathit{leave}, \mathit{stay}\}$,
– $\mathcal{R} = \{\mathit{leave} \leftarrow \mathsf{a}, \quad \mathit{stay} \leftarrow \mathsf{b}\}$,
– $\mathcal{A} = \{\mathsf{a}, \mathsf{b}\}$,
– $\overline{\mathsf{a}} = \mathit{stay}, \quad \overline{\mathsf{b}} = \mathit{leave}$,
– $\mathsf{a} \leqslant \mathsf{b}, \quad \mathsf{b} \nleqslant \mathsf{a}$.[1]

Here, assumptions a and b stand for believing in Ann and Bob, respectively. Rules $\mathit{leave} \leftarrow \mathsf{a}$ and $\mathit{stay} \leftarrow \mathsf{b}$ represent the statements of Zed's interlocutors: for instance, $\mathit{leave} \leftarrow \mathsf{a}$ 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 $\mathsf{a} < \mathsf{b}$ (i.e. $\mathsf{a} \leqslant \mathsf{b}$, $\mathsf{b} \nleqslant \mathsf{a}$), 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 $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}, \leqslant)$, 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 $A \leadsto_< B$, iff

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

We call $<$-attack formed as in the first bullet point above a *normal attack*, and $<$-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 $A \not\leadsto_< B$. (We will adopt an analogous convention for other attack relations in this paper.)

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

[1] As a preorder, $\leqslant$ 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 (and other formalisms) may be a complex problem, discussion of which is beyond the scope of this paper.

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* iff $A = Cn(A) \cap \mathcal{A}$. We say that $\mathcal{F}$ is *flat* iff 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 $\{\mathsf{a}\}$ and $\{\mathsf{b}\}$ are a and b, respectively, so both $\{\mathsf{a}\}$ and $\{\mathsf{b}\}$ are closed; clearly, $\mathcal{A}$ is closed; hence, all sets of assumptions are closed.

Further, for $A \subseteq \mathcal{A}$, we say that: A is $<$-*conflict-free* iff $A \not\rightsquigarrow_< A$; also, A $<$-*defends* $A' \subseteq \mathcal{A}$ iff for all $B \subseteq \mathcal{A}$ with $B \rightsquigarrow_< A'$ it holds that $A \rightsquigarrow_< B$; and A is $<$-*admissible* iff 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* iff E is $<$-admissible and contains every set of assumptions it $<$-defends;
- $<$-*preferred* iff E is $\subseteq$-maximally $<$-admissible;
- $<$-*stable* iff E is $<$-conflict-free and for all $\mathsf{b} \in \mathcal{A} \setminus E$ it holds that $E \rightsquigarrow_< \{\mathsf{b}\}$;
- $<$-*ideal* iff E is $\subseteq$-maximal among sets of assumptions that are $<$-admissible and contained in all $<$-preferred sets of assumptions;
- $<$-*grounded* iff E is a $\subseteq$-minimal $<$-complete set of assumptions.

Throughout the paper, $\sigma \in \{$grounded, ideal, stable, preferred, complete$\}$, and we assume that $<$-σ denotes any of the above ABA$^+$ semantics.

To illustrate with Example 1, it is easy to see that $\{\mathsf{b}\}$ is a unique $<$-σ 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 [7]. Therefore, our implementation of ABA$^+$ will be an implementation of ABA too.

Abstract Argumentation (AA). We base the background on AA on [9].

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 and with $\rightsquigarrow$ replacing $\rightsquigarrow_<$.

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

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

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

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

From now on, unless specified otherwise, we assume a PAF $(Args, \rightsquigarrow, \preccurlyeq)$ as given, and denote its repaired framework by $(Args, \hookrightarrow)$.

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

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

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. $\mathcal{D} = \{S \subseteq \mathcal{A} \ : \ S \vdash \bar{\mathsf{a}}, \quad \mathsf{a} \in \mathcal{A}\}$. The* ***assumption graph*** *of $\mathcal{F}$ is an AA framework $\mathcal{G} = (Args, \hookrightarrow)$ with*

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

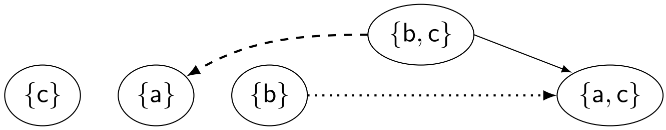
where $\rightsquigarrow_<$ is the $<$-attack relation of $\mathcal{F}$.

Example 3. Consider the ABA$^+$ framework $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-}, \leqslant)$ with

- $\mathcal{L} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, e, d, f\}$,
- $\mathcal{R} = \{d \leftarrow \mathsf{a}, \mathsf{c}, \quad e \leftarrow \mathsf{b}, \mathsf{c}\}$,
- $\mathcal{A} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$,
- $\bar{\mathsf{a}} = e, \ \bar{\mathsf{b}} = d, \ \bar{\mathsf{c}} = f$,
- $\mathsf{a} < \mathsf{b}$ (i.e. $\leqslant$ is a preorder with $\mathsf{a} \leqslant \mathsf{b}$, $\mathsf{b} \nleqslant \mathsf{a}$).

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

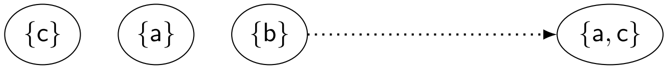
The assumption graph $\mathcal{G} = (Args, \hookrightarrow)$ of $\mathcal{F}$ has $Args = \{\{\mathsf{a}\},\{\mathsf{b}\},\{\mathsf{c}\},\{\mathsf{b},\mathsf{c}\}, \{\mathsf{a},\mathsf{c}\}\}$ and attacks $\{\mathsf{b},\mathsf{c}\} \hookrightarrow \{\mathsf{a}\}$, $\{\mathsf{b},\mathsf{c}\} \hookrightarrow \{\mathsf{a},\mathsf{c}\}$, $\{\mathsf{b}\} \hookrightarrow \{\mathsf{a},\mathsf{c}\}$, 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).



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

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

Example 4. Modify the ABA$^+$ framework $\mathcal{F}$ from Example 3 by removing the rule $e \leftarrow \mathsf{b},\mathsf{c}$ from $\mathcal{R}$ to obtain the ABA$^+$ framework $\mathcal{F}' = (\mathcal{L}, \mathcal{R}', \mathcal{A}, \overline{}, \leqslant)$ with $\mathcal{R}' = \mathcal{R} \setminus \{e \leftarrow \mathsf{b},\mathsf{c}\}$. Observe that, in $\mathcal{F}'$, all singleton sets of assumptions $\{\mathsf{a}\}$, $\{\mathsf{b}\}$ and $\{\mathsf{c}\}$ are $<$-unattacked, and hence $<$-defended by any set. However, $\{\mathsf{a},\mathsf{b},\mathsf{c}\}$ is not $<$-conflict-free, as $\{\mathsf{a},\mathsf{c}\} \vdash \overline{\mathsf{b}}$. 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 $\{\{\mathsf{a}\},\{\mathsf{b}\},\{\mathsf{c}\}\}$.



In the next section, we identify a property of ABA$^+$ frameworks, called the Axiom of Weak Contraposition,[3] 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 [17].

[3] Our notion of 'weak contraposition' bears no relationship with the notion by the same name used e.g. in [15], inspired by conditional entailment in Deontic Logic.

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

> ***if*** $A \vdash \overline{\mathsf{b}}$ and *there exists* $\mathsf{a}' \in A$ *such that* $\mathsf{a}' < \mathsf{b}$,
> ***then****, for* some $\mathsf{a} \in A$ *which is* $\leqslant$*-minimal such that* $\mathsf{a} < \mathsf{b}$*, there is* $A_\mathsf{a} \vdash \overline{\mathsf{a}}$*,*
> *for some* $A_\mathsf{a} \subseteq (A \setminus \{\mathsf{a}\}) \cup \{\mathsf{b}\}$.

This axiom insists on contrapositive reasoning when an argument involves assumptions less preferred than the one whose contrary it supports. In essence, WCP plays the role of ensuring that, in a conflict arising from assumptions, contraries and rules, preferences help to pinpoint a culprit assumption that should be argued against. A culprit is identified as being least preferred among the assumptions that are used to derive the contrary of an assumption which is more preferred than those assumptions. In this way, WCP ensures an $<$-attack against such a culprit assumption from (some of) the rest of the assumptions.

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

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

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* $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}, \leqslant)$ *satisfies WCP and let* $\mathcal{G} = (Args, \hookrightarrow)$ *be the assumption graph of* $\mathcal{F}$ *with* $Args = \mathcal{D} \cup \{\{\mathsf{a}\} \ : \ \mathsf{a} \in \mathcal{A}\}$*, where* $\mathcal{D} = \{S \subseteq \mathcal{A} \ : \ S \vdash \overline{\mathsf{a}}, \ \ \mathsf{a} \in \mathcal{A}\}$*, as in Definition 1.*

- *If* $E \subseteq \mathcal{A}$ *is a* $<$*-*σ *extension of* $\mathcal{F}$*, then* $\{S \in \mathcal{D} \ : \ S \subseteq E\} \cup \{\{\mathsf{a}\} \ : \ \mathsf{a} \in E\}$ *is a* σ *extension of* $\mathcal{G}$*;*
- *If* $\mathcal{E} \subseteq Args$ *is a* σ *extension of* $\mathcal{G}$*, then* $\bigcup \mathcal{E}$ *is a* $<$*-*σ *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 $\mathcal{E} = \{\{\mathsf{b}\}, \{\mathsf{c}\}, \{\mathsf{b}, \mathsf{c}\}\}$, and $E = \bigcup \mathcal{E} = \{\mathsf{b}, \mathsf{c}\}$ 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 $\{\{\mathsf{a}\}, \{\mathsf{b}\}, \{\mathsf{c}\}\}$, while $\mathcal{F}'$ has no $<$-complete extensions.

Theorem 1 provides a theoretical underpinning for the ABAplus system which we will describe in Sect. 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.

3.3 Enforcing WCP

Adding rules to an ABA$^+$ framework which violates WCP so that the framework with additional rules satisfies WCP is called *enforcing* WCP. In this section we detail how to enforce WCP on an ABA$^+$ framework.

A situation where an argument satisfies the antecedent of WCP while the consequent is false is called an *instance of WCP-violation* (*instance of WCP-v*, for short). More formally:

Definition 2. *$A \vdash \overline{\mathsf{b}}$ is an* ***instance of WCP-v*** *just in case*

- *$\exists \mathsf{a}' \in A$ such that $\mathsf{a}' < \mathsf{b}$, and*
- *for no $\mathsf{a} \in A$ which is $\leqslant$-minimal such that $\mathsf{a} < \mathsf{b}$ there is $A_\mathsf{a} \vdash \overline{\mathsf{a}}$, for some $A_\mathsf{a} \subseteq (A \setminus \{\mathsf{a}\}) \cup \{\mathsf{b}\}$.*

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

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

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

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

A witness to an instance of WCP-v 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 $A \vdash \overline{\mathsf{b}}$ be an instance of WCP-v and $\mathsf{a} \in A$ a witness to $A \vdash \overline{\mathsf{b}}$. Say $A = \{\mathsf{a}_1, \ldots, \mathsf{a}_n\}$, where $\mathsf{a} = \mathsf{a}_i$ for some i. The* ***enforcing rule****, denoted by $\overline{\mathsf{a}} \leftarrow A \setminus \mathsf{a}, \mathsf{b}$, is the rule $\overline{\mathsf{a}} \leftarrow \mathsf{a}_1, \ldots, \mathsf{a}_{i-1}, \mathsf{a}_{i+1}, \ldots, \mathsf{a}_n, \mathsf{b}$.*

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

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

Theorem 2. *Let $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}, \leqslant)$ be an ABA$^+$ framework and let V be the set of instances of WCP-v (in $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}, \leqslant)$). For any $A \vdash \overline{\mathsf{b}} \in V$, let*

$$\mathcal{R}_{A \vdash \overline{\mathsf{b}}} = \{\overline{\mathsf{a}} \leftarrow A \setminus \mathsf{a}, \mathsf{b} \textit{ is an enforcing rule } : \mathsf{a} \in A \textit{ is a witness to } A \vdash \overline{\mathsf{b}}\}$$

be the set of enforcing rules for $A \vdash \overline{\mathsf{b}}$. Let f be a function, defined for finite non-empty sets, that selects any one element from a given set. The ABA$^+$ framework $(\mathcal{L}, \mathcal{R} \cup \mathcal{R}', \mathcal{A}, \overline{}, \leqslant)$, where $\mathcal{R}' = \{f(\mathcal{R}_{A \vdash \overline{\mathsf{b}}}) : A \vdash \overline{\mathsf{b}} \in V\}$, satisfies WCP.

Proof. Let $A \vdash \overline{\mathsf{b}}$ be any instance of WCP-v (in $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}, \leqslant)$) and let $\mathsf{a} \in A$ be a witness to $A \vdash \overline{\mathsf{b}}$. The rule $\overline{\mathsf{a}} \leftarrow A \backslash \mathsf{a}, \mathsf{b}$ guarantees that in $(\mathcal{L}, \mathcal{R} \cup \mathcal{R}', \mathcal{A}, \overline{}, \leqslant)$ we can find the argument $(A \setminus \{\mathsf{a}\}) \cup \{\mathsf{b}\} \vdash^{\{\overline{\mathsf{a}} \leftarrow A \backslash \mathsf{a}, \mathsf{b}\}} \overline{\mathsf{a}}$. Note that no such argument can result in an instance of WCP-v (in $(\mathcal{L}, \mathcal{R} \cup \mathcal{R}', \mathcal{A}, \overline{}, \leqslant)$), precisely because the witness a is $\leqslant$-minimal. Therefore, $(\mathcal{L}, \mathcal{R} \cup \mathcal{R}', \mathcal{A}, \overline{}, \leqslant)$ satisfies WCP.

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

Several remarks regarding WCP are in place.

First, note well that enforcing WCP on an ABA$^+$ framework *does in general change* its semantics. For instance, $\mathcal{F}'$ from Example 4 has no $<$-complete extensions, whereas $\mathcal{F}$ from Example 3 obtained by enforcing WCP on $\mathcal{F}'$ has a unique $<$-complete extension. Using preferences to identify a culprit assumption to be argued against, and thus changing the semantics of an ABA$^+$ framework, is one of the objectives of enforcing WCP. Precisely this allows to obtain semantic correspondence between ABA$^+$ frameworks and their assumption graphs.

Second, observe that using WCP *does not* amount to 'making attacks symmetric'. Indeed, consider $A \rightsquigarrow \{\mathsf{b}\}$ and let $A \vdash \overline{\mathsf{b}}$ be an instance of WCP-v with a witness $\mathsf{a}' \in A$ such that $\mathsf{a}' < \mathsf{b}$. Making this attack symmetric means imposing $\{\mathsf{b}\} \rightsquigarrow_< A$. However, WCP *does not* require $\{\mathsf{b}\} \rightsquigarrow_< A$. Instead, WCP requires that $(A \setminus \{\mathsf{a}'\}) \cup \{\mathsf{b}\} \vdash \overline{\mathsf{a}'}$, which in general amounts to $(A \setminus \{\mathsf{a}'\}) \cup \{\mathsf{b}\} \rightsquigarrow_< \{\mathsf{a}'\}$ (and hence $(A \setminus \{\mathsf{a}'\}) \cup \{\mathsf{b}\} \rightsquigarrow_< A$).

Third, the $\leqslant$-minimality of a witness assumption in enforcing WCP is crucial. In particular, it saves from generating redundant arguments when enforcing the axiom. For instance, consider $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}, \leqslant)$ with $\mathcal{R} = \{\overline{\mathsf{c}} \leftarrow \mathsf{a}, \mathsf{b}\}$, $\mathcal{A} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ and $\mathsf{a} < \mathsf{b} < \mathsf{c}$.[4] The argument $\{\mathsf{a}, \mathsf{b}\} \vdash \overline{\mathsf{c}}$ is an instance of WCP-v.If $\leqslant$-minimality were not required in the conditions of the consequent of WCP, one could end up choosing b and adding the rule $\overline{\mathsf{b}} \leftarrow \mathsf{a}, \mathsf{c}$ to $\mathcal{R}$ so as to generate the argument $\{\mathsf{a}, \mathsf{c}\} \vdash \overline{\mathsf{b}}$ in $\mathcal{F}' = (\mathcal{L}, \mathcal{R} \cup \{\overline{\mathsf{b}} \leftarrow \mathsf{a}, \mathsf{c}\}, \mathcal{A}, \overline{}, \leqslant)$. This would result in $\{\mathsf{a}, \mathsf{c}\} \vdash \overline{\mathsf{b}}$ making the antecedent of the WCP true (because $\mathsf{a} < \mathsf{b}$) while keeping the consequent false, thus yielding an instance of WCP-v in $\mathcal{F}'$. Thus, to enforce WCP, one would need to ensure existence of yet another argument, for example, $\{\mathsf{b}, \mathsf{c}\} \vdash \overline{\mathsf{a}}$, by, for instance, adding the rule $\overline{\mathsf{a}} \leftarrow \mathsf{b}, \mathsf{c}$. By contrast, choosing a (necessarily $\leqslant$-minimal) witness to begin with, i.e. a, and adding a single rule, say the enforcing rule $\overline{\mathsf{a}} \leftarrow \mathsf{b}, \mathsf{c}$, generates the argument $\{\mathsf{b}, \mathsf{c}\} \vdash \overline{\mathsf{a}}$ in $\mathcal{F}'' = (\mathcal{L}, \mathcal{R} \cup \{\overline{\mathsf{a}} \leftarrow \mathsf{b}, \mathsf{c}\}, \mathcal{A}, \overline{}, \leqslant)$. Thus, the instance of WCP-v in question is eliminated in $\mathcal{F}''$ and no further instances of WCP-v are obtained in $\mathcal{F}''$.

The fourth remark concerns other ways to enforce WCP on a given ABA$^+$ framework. For example, given an instance of WCP-v $A \vdash \overline{\mathsf{b}}$ with a witness $\mathsf{a} \in A$, one could add the rule $\overline{\mathsf{a}} \leftarrow \top$ to obtain the argument $\emptyset \vdash^{\{\overline{\mathsf{a}} \leftarrow \top\}} \overline{\mathsf{a}}$ as required to eliminate the instance of WCP-v 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

[4] Unless specified otherwise, we omit $\mathcal{L}$ and $\overline{}$, and adopt the following conventions: unless $\overline{\mathsf{x}}$ 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 $\{\overline{\mathsf{a}} \ : \ \mathsf{a} \in \mathcal{A}\}$.

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 ensures satisfaction of WCP while leaving 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 $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}, \leqslant)$ with $\mathcal{L} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, x, y\}$, $\mathcal{R} = \{x \leftarrow \mathsf{a}\}$, $\mathcal{A} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$, $\overline{\mathsf{a}} = \mathsf{c}$, $\overline{\mathsf{b}} = x$, $\overline{\mathsf{c}} = y$, $\mathsf{a} < \mathsf{b}$. We find $\{\mathsf{a}\} \vdash \overline{\mathsf{b}}$ and $\mathsf{a} < \mathsf{b}$, so to enforce WCP, we need an argument $S \vdash \overline{\mathsf{a}}$, for some $S \subseteq \{\mathsf{b}\}$. Whatever the support S, we get an argument $S \vdash \mathsf{c}$, whence S is not closed. Note, however, that this behaviour can be easily avoided and flatness guaranteed. Indeed, instead of defining the contrary mapping $\overline{} : \mathcal{A} \to \mathcal{L}$ to map assumptions into 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 $\mathcal{C}(\mathsf{a}) = \mathsf{a}^c$; then, for any intended contrary x of a in $\mathcal{L}$, we add a rule $\mathsf{a}^c \leftarrow x$. 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 [19], we show how ABA$^+$ admits PAFs as instances. As a result, implementing attack reversal in ABA$^+$ will automatically give a way to implement attack reversal in PAFs.

To instantiate an ABA$^+$ framework with $(\mathit{Args}, \rightsquigarrow, \preccurlyeq)$, we map each argument $\mathsf{a} \in \mathit{Args}$ into an assumption $\mathsf{a} \in \mathcal{A}$, together with a new symbol $\overline{\mathsf{a}}$ for the contrary, map each attack $\mathsf{a} \rightsquigarrow \mathsf{b}$ into a rule $\overline{\mathsf{b}} \leftarrow \mathsf{a}$, and transfer the preference ordering $\preccurlyeq$ to constitute $\leqslant$, as follows.

Definition 5. *Given a PAF $(\mathit{Args}, \rightsquigarrow, \preccurlyeq)$, an ABA$^+$* ***framework corresponding to*** *$(\mathit{Args}, \rightsquigarrow, \preccurlyeq)$ is $\mathcal{F}_{\mathrm{PAF}} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}, \leqslant)$ with:*

- $\mathcal{L} = \mathit{Args} \cup \{\overline{\mathsf{a}} \ : \ \mathsf{a} \in \mathit{Args},\ \overline{\mathsf{a}} \notin \mathit{Args}\}$;
- $\mathcal{R} = \{\overline{\mathsf{b}} \leftarrow \mathsf{a} \ : \ \mathsf{a} \rightsquigarrow \mathsf{b}\}$;
- $\mathcal{A} = \mathit{Args}$;
- *for* $\mathsf{a} \in \mathcal{A}$, $\overline{\mathsf{a}}$ *is the contrary of* a;
- $\leqslant = \preccurlyeq$.

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

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

Note that $\mathcal{F}_{\mathrm{PAF}}$ is necessarily flat. However, $\mathcal{F}_{\mathrm{PAF}}$ need not in general satisfy WCP. Nonetheless, given that all (ABA$^+$) arguments for contraries in $\mathcal{F}_{\mathrm{PAF}}$ are supported by singleton sets, every instance of WCP-v $\{\mathsf{a}\} \vdash \overline{\mathsf{b}}$ (with $\mathsf{a} < \mathsf{b}$) has a unique witness a, so that enforcing WCP on $\mathcal{F}_{\mathrm{PAF}}$ as in Theorem 2 yields a unique ABA$^+$ framework $\mathcal{F}'_{\mathrm{PAF}}$ which has the same $<$-attack relation as $\mathcal{F}_{\mathrm{PAF}}$: for $A, B \subseteq \mathcal{A}$, $A \leadsto_< B$ iff $A \leadsto'_< B$, where $\leadsto_<$ and $\leadsto'_<$ are $<$-attack relations of $\mathcal{F}_{\mathrm{PAF}}$ and $\mathcal{F}'_{\mathrm{PAF}}$, respectively.

Observe further that since all (ABA$^+$) arguments for contraries in $\mathcal{F}_{\mathrm{PAF}}$ are supported by singleton sets, attacks in $(\mathit{Args}, \leadsto, \preccurlyeq)$ coincide with $<$-attacks in $\mathcal{F}_{\mathrm{PAF}}$ (in that for $\mathsf{a}, \mathsf{b} \in \mathit{Args}$, $\mathsf{a} \hookrightarrow \mathsf{b}$ iff $\{\mathsf{a}\} \leadsto_< \{\mathsf{b}\}$). We thus obtain the following correspondence result, which says that, under any semantics σ, every PAF is an instance of ABA$^+$, and that determining extensions of $(\mathit{Args}, \leadsto, \preccurlyeq)$ amounts to determining extensions of $\mathcal{F}_{\mathrm{PAF}}$, which amounts to determining extensions of $\mathcal{F}_{\mathrm{PAF}}$ with WCP enforced.

Theorem 3. *Let $(\mathit{Args}, \leadsto, \preccurlyeq)$ be a PAF, $\mathcal{F}_{\mathrm{PAF}}$ be an ABA$^+$ framework corresponding to $(\mathit{Args}, \leadsto, \preccurlyeq)$, and $E \subseteq \mathit{Args}$. E is a σ extension of $(\mathit{Args}, \leadsto, \preccurlyeq)$ iff E is a $<$-σ extension of $\mathcal{F}_{\mathrm{PAF}}$ iff E is a $<$-σ extension of $\mathcal{F}'_{\mathrm{PAF}}$, where $\mathcal{F}'_{\mathrm{PAF}}$ is obtained by enforcing WCP on $\mathcal{F}_{\mathrm{PAF}}$ as in Theorem 2.*

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

Theorem 3 implies that 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 an 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 $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\ }, \leqslant)$, ABAplus feeds the assumption graph $\mathcal{G} = (\mathit{Args}, \hookrightarrow)$ (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 http://www-abaplus.doc.ic.ac.uk takes a single ABA$^+$ framework $\mathcal{F}$ as input in the Prolog-like format:

- `myAsm(a).` specifies that a is an assumption from $\mathcal{A}$;

- `contrary(a, `x`).` specifies that $x \in \mathcal{L}$ is the contrary $\bar{\mathsf{a}}$ of assumption a;
- `myRule(`h`, [`$b_1, \ldots, b_n$`]).` specifies that $h \leftarrow b_1, \ldots, b_n$ is a rule from $\mathcal{R}$;
- `myPrefLE(b, a).` specifies, for assumptions a, b, that $\mathsf{b} \leqslant \mathsf{a}$;
- `myPrefLT(b, a).` specifies, for assumptions a, b, that $\mathsf{b} < \mathsf{a}$.

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 be highlighted. 4. A second graph can be displayed with any 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 $\leqslant$; (c) checking whether the strict counterpart $<$ of (the updated) $\leqslant$ 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 $\mathsf{a} \in \mathcal{A}$, store $\{\mathsf{a}\}$ in *Args*;
   (b) For every assumption $\mathsf{a} \in \mathcal{A}$, generate all arguments for $\bar{\mathsf{a}}$ and store the supports of those arguments in *Args*.
3. Generation of $\hookrightarrow$ goes thus:
   (a) For every $\mathsf{a} \in \mathcal{A}$, for every $B \vdash \bar{\mathsf{a}}$, check if $\exists \mathsf{b} \in B$ such that $\mathsf{b} < \mathsf{a}$: (i) if no, store $B \hookrightarrow \{\mathsf{a}\}$; (ii) else, store $\{\mathsf{a}\} \hookrightarrow B$;
   (b) For any $\mathsf{A}', \mathsf{B}' \in \mathit{Args}$ such that $\{\mathsf{a}\} \subseteq \mathsf{A}'$ and $B \subseteq \mathsf{B}'$, (i) if $B \hookrightarrow \{\mathsf{a}\}$, store $\mathsf{B}' \hookrightarrow \mathsf{A}'$; (ii) if $\{\mathsf{a}\} \hookrightarrow B$, store $\mathsf{A}' \hookrightarrow \mathsf{B}'$.
4. The assumption graph $\mathcal{G} = (\mathit{Args}, \hookrightarrow)$ thus constructed is fed to ASPARTIX (using clingo and DLV ASP solvers), represented via Prolog-like sentences:
   - `arg(A).` represents an argument $\mathsf{A} \in \mathit{Args}$;
   - `att(A, B).` represents an attack $\mathsf{A} \hookrightarrow \mathsf{B}$.
5. For every semantics $\sigma \in$ {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; Gunicorn 19.6; Clingo 4.5.4; DLV ("deductive database system") version 17/12/2012; encodings of semantics (stable.dl, ideal.dl, comp.dl, prefex_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 preferences 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.

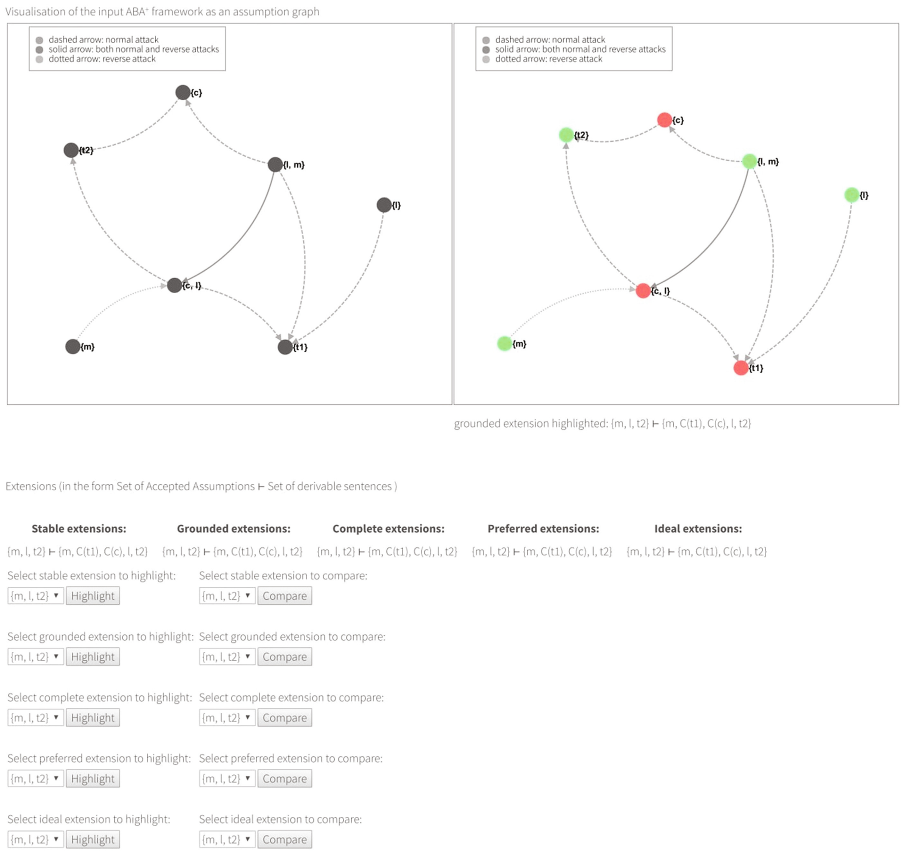
Example 5 (Preferences over goals). Consider ABAplus as a system to schedule meetings. Imagine a user who needs to schedule a meeting on one of two suggested time slots, t_1 and t_2. Suppose that t_1 is the time when the user usually has lunch, and that t_2 covers the user's standard coffee break. The 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 $\overline{t_1} \leftarrow l$ and $\overline{t_2} \leftarrow c$ express that having lunch and coffee as usual make the two respective time slots unavailable. Additionally, the rule $\overline{m} \leftarrow \overline{t_1}, \overline{t_2}$ expresses that the meeting will not be scheduled if none of the time slots are available. Finally, user's preferences are expressed by letting $c < m < l$: having lunch as usual is (strictly) more important than scheduling the meeting, which is in turn (strictly) more important than taking a coffee break at a standard time.

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

```
myAsm($c$).          contrary($c$, $\overline{c}$).          myRule($\overline{t_1}$, [$l$]).
myAsm($l$).          contrary($l$, $\overline{l}$).          myRule($\overline{t_2}$, [$c$]).
myAsm($m$).          contrary($m$, $\overline{m}$).          myRule($\overline{m}$, [$\overline{t_1}$, $\overline{t_2}$]).
myAsm($t_1$).        contrary($t_1$, $\overline{t_1}$).      myPrefLT($c$, $m$).
myAsm($t_2$).        contrary($t_2$, $\overline{t_2}$).      myPrefLT($m$, $l$).
```

Given this input, ABAplus recognizes that $\mathcal{F}$ does not satisfy WCP: there is $\{c, l\} \vdash \overline{m}$ with $c < m$, but there is no argument for $\overline{c}$ at all. Thus, ABAplus informs the user accordingly, and proposes to automatically enforce WCP on $\mathcal{F}$. ABAplus enforces WCP by adding the rule $\overline{c} \leftarrow m, l$, 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 $\{m, l, t_2, \overline{t_1}, \overline{c}\}$, 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 Fig. 1 (cropped by cutting out the part with the editable window for the input).



Visualisation of the input ABA⁺ framework as an assumption graph

● dashed arrow: normal attack
● solid arrow: both normal and reverse attacks
● dotted arrow: reverse attack
{c}
{t2}
{l, m}
{l}
{c, l}
{m}
{t1}
● dashed arrow: normal attack
● solid arrow: both normal and reverse attacks
● dotted arrow: reverse attack
{c}
{t2}
{l, m}
{l}
{c, l}
{m}
{t1}

grounded extension highlighted: {m, l, t2} ⊢ {m, C(t1), C(c), l, t2}

Extensions (in the form Set of Accepted Assumptions ⊢ Set of derivable sentences)

Stable extensions:	Grounded extensions:	Complete extensions:	Preferred extensions:	Ideal extensions:
{m, l, t2} ⊢ {m, C(t1), C(c), l, t2}	{m, l, t2} ⊢ {m, C(t1), C(c), l, t2}	{m, l, t2} ⊢ {m, C(t1), C(c), l, t2}	{m, l, t2} ⊢ {m, C(t1), C(c), l, t2}	{m, l, t2} ⊢ {m, C(t1), C(c), l, t2}

Select stable extension to highlight: Select stable extension to compare:
{m, l, t2} ▾ Highlight {m, l, t2} ▾ Compare

Select grounded extension to highlight: Select grounded extension to compare:
{m, l, t2} ▾ Highlight {m, l, t2} ▾ Compare

Select complete extension to highlight: Select complete extension to compare:
{m, l, t2} ▾ Highlight {m, l, t2} ▾ Compare

Select preferred extension to highlight: Select preferred extension to compare:
{m, l, t2} ▾ Highlight {m, l, t2} ▾ Compare

Select ideal extension to highlight: Select ideal extension to compare:
{m, l, t2} ▾ Highlight {m, l, t2} ▾ Compare

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

Preferences over rules indicate which rules should be followed in case application of multiple rules is impossible. In ABA$^+$, preferences 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 it, you should follow a healthy diet’ and ‘if you can afford it, you should exercise regularly’. These two rules can be represented in ABA$^+$ as $d \leftarrow \mathsf{c}$ and $e \leftarrow \mathsf{c}$, 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 $\bar{\mathsf{c}} \leftarrow d, e$.

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 $e \leftarrow \mathsf{c}$ and $d \leftarrow \mathsf{c}$ into new rules $d \leftarrow \mathsf{a}, \mathsf{c}$ and $e \leftarrow \mathsf{b}, \mathsf{c}$, respectively. Finally, add the preference $\mathsf{a} < \mathsf{b}$ over the new assumptions.

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

6 Related Work

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

TOAST[5] [18] is a web application implementing (an early version [17] of) the structured argumentation formalism 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. TOAST features rule *transposition*, which is related to contraposition. ABAplus instead enforces WCP.

Gorgias-B[6] is a stand-alone system implementing *Gorgias* [14]—an argumentation formalism based on logic programming without negation as failure [8], 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[7] is a web application implementing reasoning in Defeasible Logic Programming (DeLP) [11]. 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 plan to explore in the future whether explanations could be implemented in ABAplus.

[5] http://toast.arg-tech.org.

[6] http://gorgiasb.tuc.gr/index.html.

[7] http://lidia.cs.uns.edu.ar/delp_client.

Carneades[8] [13] is a web application and a stand-alone system implementing the argumentation formalism of the same name [12]. 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[9] [4] is a web application 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 for 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 (possibly 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 Sect. 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. Finally, we have outlined possible uses of ABAplus in a number of settings. In the future we plan to investigate fully fledged applications of the system in general and in medical settings, e.g. to support reasoning with (possibly conflicting) guidelines and clinical pathways as well as preferences derived from resource constraints.

[8] http://carneades.fokus.fraunhofer.de/carneades.

[9] http://www.dmi.unipg.it/conarg.

Acknowledgements. This research was partially funded by the EPSRC project **EP/P029558/1** ROAD2H: Resource Optimisation, Argumentation, Decision Support and Knowledge Transfer to Create Value via Learning Health Systems. We also thank Jeff Thompson for helpful discussions on the use of ABAplus to schedule meetings (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. Approximate Reasoning **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. Argum. Comput. **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.) Computational Models of Argument. 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.) 19th European Conference on Artificial Intelligence. Frontiers in Artificial Intelligence and Applications, vol. 215, pp. 131–136. IOS Press, Lisbon (2010)
6. Bondarenko, A., Dung, P.M., Kowalski, R., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. Artif. Intell. **93**(97), 63–101 (1997)
7. Čyras, K., Toni, F.: ABA+: assumption-based argumentation with preferences. In: Baral, C., Delgrande, J.P., Wolter, F. (eds.) 15th International Conference on Principles of Knowledge Representation and Reasoning, pp. 553–556. AAAI Press, Cape Town (2016)
8. Dimopoulos, Y., Kakas, A.C.: Logic programming without negation as failure. In: Lloyd, J.W. (ed.) International Symposium on Logic Programming, pp. 369–383. MIT Press, Portland (1995)
9. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. Artif. Intell. **77**, 321–357 (1995)
10. 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)
11. García, A.J., Simari, G.R.: Defeasible logic programming: DeLP-servers, contextual queries, and explanations for answers. Argum. Comput. **5**(1), 63–88 (2014)
12. Gordon, T.F., Prakken, H., Walton, D.: The carneades model of argument and burden of proof. Artif. Intell. **171**(10–15), 875–896 (2007)
13. Gordon, T.F., Walton, D.: Formalizing balancing arguments. In: Baroni, P., Gordon, T.F., Scheffler, T., Stede, M. (eds.) Computational Models of Argument. Frontiers in Artificial Intelligence and Applications, vol. 287, pp. 327–338. IOS Press, Potsdam (2016)
14. Kakas, A.C., Moraitis, P.: Argumentation based decision making for autonomous agents. In: 2nd International Joint Conference on Autonomous Agents & Multiagent Systems, pp. 883–890. ACM Press, Melbourne (2003)

15. Liao, B., Oren, N., van der Torre, L., Villata, S.: Prioritized norms and defaults in formal argumentation. In: Roy, O., Tamminga, A., Wille, M. (eds.) 13th International Conference on Deontic Logic and Normative Systems, pp. 139–154. College Publications, Bayreuth (2016)
16. Modgil, S., Prakken, H.: A general account of argumentation with preferences. Artif. Intell. **195**, 361–397 (2013)
17. Prakken, H.: An abstract framework for argumentation with structured arguments. Argum. Comput. **1**(2), 93–124 (2010)
18. Snaith, M., Reed, C.: TOAST: online ASPIC+ implementation. In: Verheij, B., Szeider, S., Woltran, S. (eds.) Computational Models of Argument. Frontiers in Artificial Intelligence and Applications, vol. 245, pp. 509–510. IOS Press, Vienna (2012)
19. Toni, F.: Reasoning on the web with assumption-based argumentation. In: Eiter, T., Krennwallner, T. (eds.) Reasoning Web 2012. LNCS, vol. 7487, pp. 370–386. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33158-9_10
20. Toni, F.: A tutorial on assumption-based argumentation. Argum. Comput. **5**(1), 89–117 (2014)
21. Wakaki, T.: Assumption-based argumentation equipped with preferences. In: Dam, H.K., Pitt, J., Xu, Y., Governatori, G., Ito, T. (eds.) PRIMA 2014. LNCS (LNAI), vol. 8861, pp. 116–132. Springer, Cham (2014). doi:10.1007/978-3-319-13191-7_10