7

Assumption-Based Argumentation: Disputes, Explanations, Preferences

KRISTIJONAS ČYRAS, XIUYI FAN, CLAUDIA SCHULZ, FRANCESCA TONI

ABSTRACT. Assumption-Based Argumentation (ABA) is a form of structured argumentation with roots in non-monotonic reasoning. As in other forms of structured argumentation, notions of argument and attack are not primitive in ABA, but are instead defined in terms of other notions. In the case of ABA these other notions are those of rules in a deductive system, assumptions, and contraries.

ABA is equipped with a range of computational tools, based on dispute trees and amounting to dispute derivations, and benefiting from equivalent views of the semantics of argumentation in ABA, in terms of sets of arguments and, equivalently, sets of assumptions. These computational tools can also provide the foundation for multi-agent argumentative dialogues and explanation of reasoning outputs, in various settings and senses.

ABA is a flexible modelling formalism, despite its simplicity, allowing to support, in particular, various forms of non-monotonic reasoning, and reasoning with some forms of preferences and defeasible rules without requiring any additional machinery. ABA can also be naturally extended to accommodate further reasoning with preferences.

1 Introduction

Assumption-Based Argumentation (ABA) [Bondarenko *et al.*, 1993; Bondarenko *et al.*, 1997; Dung *et al.*, 2009; Toni, 2014] is a form of *structured argumentation* [Besnard *et al.*, 2014] with roots in non-monotonic reasoning [Brewka *et al.*, 1997]. Differently from abstract argumentation [Dung, 1995] but as in other forms of structured argumentation, e.g. DeLP [García and Simari, 2014] and deductive arguments [Besnard and Hunter, 2014], notions of argument and attack are not primitive in ABA, but are instead defined in terms of other notions. In the case of ABA these notions are those of *rules* in an underlying *deductive system*, *assumptions* and their *contraries*: arguments are supported by rules and assumptions and attacks are directed against (assumptions deducible from) assumptions supporting arguments, by building arguments for the contrary of these assumptions. Semantics of ABA frameworks can be characterised in terms of sets of assumptions (or *extensions*) [Bondarenko *et al.*, 1993; Bondarenko *et al.*, 1997; Dung *et al.*, 2007] meeting desirable requirements, including, but not limited to, the two core requirements of *closedness* (where a set of assumptions is *closed* iff it consists of all the assumptions deducible from it) and *conflict-freeness* (where a set of assumptions is *conflict-free* iff it does not attack itself). The closedness requirement is guaranteed to be fulfilled automatically for all sets of assumptions for restricted kinds of *ABA frameworks*, referred to as *flat* [Bondarenko *et al.*, 1997]. The ABA semantics of admissible, preferred, complete, well-founded, stable and ideal extensions [Bondarenko *et al.*, 1997; Dung *et al.*, 2007] differ in which additional desirable requirements they impose upon sets of assumptions, but can all be seen as providing argumentative counterparts of semantics that had previously been defined for non-monotonic reasoning, by appropriately instantiating (flat and non-flat) ABA frameworks [Bondarenko *et al.*, 1993; Bondarenko *et al.*, 1997] to "match" existing frameworks for non-monotonic reasoning.

Flat ABA is equipped with a range of computational tools, based on dispute trees [Dung *et al.*, 2006; Dung *et al.*, 2007] and amounting to dispute derivations [Dung *et al.*, 2006; Dung *et al.*, 2007; Toni, 2013], and benefiting from equivalent views of the semantics of argumentation in flat ABA, in terms of sets of arguments and, equivalently, sets of assumptions [Dung *et al.*, 2007]. These computational tools can also provide the foundation for inter-agent *ABA dialogues* in various settings and senses [Fan and Toni, 2011b; Fan and Toni, 2011a; Fan and Toni, 2011c; Fan and Toni, 2012b; Fan and Toni, 2012a; Fan and Toni, 2012c; Fan *et al.*, 2014; Fan and Toni, 2014b; Fan and Toni, 2016] and explanations of reasoning outputs, in various settings and senses, e.g. to explain (non-)membership in answer sets of logic programs [Schulz and Toni, 2016], to explain "goodness" of decisions [Fan and Toni, 2014a; Fan *et al.*, 2013; Zhong *et al.*, 2014] and, more generically, to explain admissibility of sentences in any flat instance of ABA [Fan and Toni, 2015c].

ABA is a flexible modelling formalism, despite its simplicity, allowing to support, in particular, reasoning with some forms of preferences and defeasible rules without requiring any additional machinery [Kowalski and Toni, 1996; Toni, 2008b; Thang and Luong, 2013; Fan *et al.*, 2013], but accommodating preferences at the "object-level". ABA can also be naturally extended to accommodate further reasoning with preferences, e.g. as in [Wakaki, 2014] or as ABA^+ in [Čyras and Toni, 2016a; Čyras and Toni, 2016b].

This chapter is organised as follows. In Section 2 we recap the basic definitions of ABA frameworks and semantics, focusing on semantics that have been inspired by semantics for non-monotonic reasoning, and summarising properties of semantics, distinguishing amongst generic and flat ABA frameworks. In Section 3 we illustrate two instances of ABA, capturing autoepistemic logic and logic programming, and respectively requiring non-flat and flat ABA frameworks. From Section 4 to Section 7 we focus on flat ABA frameworks. In particular, in Section 4 we summarise how flat ABA frameworks can be equivalently understood, for all semantics considered in this chapter, as abstract argumentation frameworks [Dung, 1995], following the results in [Dung *et al.*, 2009], and, vice versa, abstract argumentation frameworks can be equivalently understood, for all semantics considered in this chapter, as flat ABA frameworks, following the results in [Toni, 2012]. In Section 5 we provide an overview and illustration of the basis of all computational machinery for ABA, in the flat case, namely dispute trees [Dung *et al.*, 2006; Dung *et al.*, 2007] and dispute derivations [Dung *et al.*, 2006; Dung *et al.*, 2007; Toni, 2013]. In this section we also illustrate how this machinery can be adapted to provide a foundation for inter-agent ABA dialogues [Fan and Toni, 2014b]. In Section 6 we overview various uses of (flat) ABA to provide explanations of reasoning outputs [Schulz and Toni, 2016; Fan and Toni, 2014a; Fan *et al.*, 2013; Zhong *et al.*, 2014; Fan *et al.*, 2014; Fan and Toni, 2015c]. In Section 7 we overview various existing approaches to accommodating preferences in (flat) ABA [Kowalski and Toni, 1996; Toni, 2008a; Toni, 2008b; Thang and Luong, 2013; Fan *et al.*, 2013] or extending ABA to accommodate reasoning with preferences [Wakaki, 2014; Čyras and Toni, 2016a; Čyras and Toni, 2016b]. Finally, in Section 8 we conclude, emphasising, in particular, omissions and future work.

This chapter complements other earlier surveys of ABA [Dung *et al.*, 2009; Toni, 2012; Toni, 2014]. In particular, all earlier surveys focused exclusively on flat ABA frameworks. These are powerful knowledge representation mechanisms, as, for example, they fully capture logic programming (see Section 3) and default logic [Reiter, 1980] (see [Bondarenko *et al.*, 1997]), both widely used formalisms for non-monotonic reasoning and knowledge representation and reasoning, as well as, for instance, some forms of decision-making (see Section 7). However, non-flat frameworks allow to capture additional forms of reasoning, including the kind of non-monotonic reasoning encapsulated by autoepistemic logic (see Section 3), as well as circumscription [McCarthy, 1980], amongst others (see [Bondarenko *et al.*, 1997]). For example, in non-flat ABA one can represent beliefs as assumptions that can be deduced via rules from other assumptions.

Moreover, differently from earlier surveys, this chapter summarises uses of ABA for non-monotonic reasoning (Section 3) and defeasible reasoning (Section 7) as well as the explanatory power of ABA (Section 6) afforded by its computational machinery. At the same time, this chapter ignores other aspects of ABA, emphasised instead in the earlier surveys, such as the equivalence between different presentations of ABA in the literature, e.g. alternative views of arguments (as trees [Dung *et al.*, 2009] rather than as forward [Bondarenko *et al.*, 1997] or backward [Dung *et al.*, 2006] deductions).

This chapter is related to a number of other chapters in this handbook. In particular, it takes for granted notions from abstract argumentation, as overviewed in the chapter on *Abstract Argumentation Frameworks and Their Semantics* in this handbook. Moreover, the chapter on *Argumentation Based on Logic Programming* presents an approach to structured argumentation grounded in logic programming, but different from the logic programming instance of ABA, and the chapter on *Argumentation, Nonmonotonic Reasoning and Logic* overviews relationships between argumentation and non-monotonic reasoning more in general; the chapter on *Computational Problems in Formal Argumentation and Their Complexity* deals with computational complexity issues, that we neglect in this chapter for ABA (but we briefly consider in Section 8); finally, the chapter on *Foundations of Implementations for Formal Argumentation* overviews implementations of argumentation, including ones for ABA, that we ignore in this chapter (but again we briefly mention in Section 8).

2 ABA frameworks and semantics

In this section we introduce ABA frameworks [Bondarenko *et al.*, 1993; Bondarenko *et al.*, 1997; Dung *et al.*, 2009; Toni, 2014] and their standard semantics of admissible, preferred, complete, well-founded (called *grounded* in the specific case of flat ABA frameworks), stable and ideal extensions [Bondarenko *et al.*, 1993; Bondarenko *et al.*, 1997; Dung *et al.*, 2007] as sets of assumptions. All the semantics considered have counterparts in logic programming, in the sense that they correspond to semantics of logic programs in the logic programming instance of ABA (see Section 3).

Definition 2.1 *An* ABA framework *is a tuple* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ *where*

- $\langle \mathcal{L}, \mathcal{R} \rangle$ *is a* deductive system*, with* $\mathcal{L}$ *a* language *(a set of sentences) and* $\mathcal{R}$ *a set of* (inference) rules*, each with a* head *and a* body*, where the head is a sentence in* $\mathcal{L}$*, and the body consists of* $m \geq 0$ *sentences in* $\mathcal{L}$*;*
- $\mathcal{A} \subseteq \mathcal{L}$ *is a (non-empty) set, with elements referred to as* assumptions*;*
- $\overline{}$ *is a total mapping from* $\mathcal{A}$ *into* $\mathcal{L}$*;* $\overline{a}$ *is referred to as the* contrary *of* a*, for* $a \in \mathcal{A}$*.*

Rules in $\mathcal{R}$ can be written in different formats, e.g. a rule with head σ_0 and body $\sigma_1, \ldots, \sigma_m$ may be written as

$$\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_m \qquad \text{or} \qquad \frac{\sigma_1, \ldots, \sigma_m}{\sigma_0}.$$

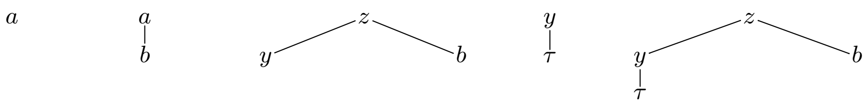
Note that $\leftarrow$ is not to be interpreted as logical implication, when used to represent rules in ABA as above. In the remainder of this paper, we will use these two syntactic conventions for writing rules interchangeably. Moreover, unless specified otherwise, we will assume as given a generic ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$. Note also that sentences have a contrary if, and only if, they are assumptions. This contrary is not to be confused with negation, which may or may not occur in $\mathcal{L}$.

Rules in ABA frameworks can be chained to form *deductions*. These can be defined in several ways, notably in a forward [Bondarenko *et al.*, 1997], a backward [Dung *et al.*, 2006] or a tree-style manner [Dung *et al.*, 2009]. We use here the latter style, as follows:

Definition 2.2 *A* deduction for $\sigma \in \mathcal{L}$ supported by $S \subseteq \mathcal{L}$ and $R \subseteq \mathcal{R}$*, denoted* $S \overset{R}{\vdash} \sigma$ *(or simply* $S \vdash \sigma$*), is a (finite) tree with*

- *nodes labelled by sentences in* $\mathcal{L}$ *or by* τ*,*[1]
- *the root labelled by* σ*,*
- *leaves either* τ *or sentences in* S*,*
- *non-leaves* σ' *with, as children, the elements of the body of some rule in* $\mathcal{R}$ *with head* σ'*, and* R *the set of all such rules.*

Example 2.3 *Consider an ABA framework* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ *with* $\mathcal{R} = \{x \leftarrow c,\ z \leftarrow y, b,\ \ y \leftarrow \ \ ,\ \ a \leftarrow b\}$ *and* $\mathcal{A} = \{a, b, c\}$*.*[2] *The following are examples of deductions, denoted as indicated (first with the supporting rules and then without):*



$\{a\} \overset{\{\}}{\vdash} a$ $\{b\} \overset{\{a \leftarrow b\}}{\vdash} a$ $\{y, b\} \overset{\{z \leftarrow y, b\}}{\vdash} z$ $\{\} \overset{\{y \leftarrow\}}{\vdash} y$ $\{b\} \overset{\{z \leftarrow y, b,\ y \leftarrow\}}{\vdash} z$

$\{a\} \vdash a$ $\{b\} \vdash a$ $\{y, b\} \vdash z$ $\{\} \vdash y$ $\{b\} \vdash z$

Note that deductions for assumptions have a non-empty rule support only if they occur as head of rules, and sentences occurring as head of rules with an empty body are always supported by an empty set of sentences (and a singleton set of rules).

Semantics of ABA frameworks are defined in terms of sets of assumptions meeting desirable requirements. One such requirement is being *closed* under deduction, defined as follows:

Definition 2.4 *The* closure *of a set of sentences* $S \subseteq \mathcal{L}$ *is*

$$Cl(S) = \{\sigma \in \mathcal{A} \mid \exists\ S' \overset{R}{\vdash} \sigma,\ S' \subseteq S,\ R \subseteq \mathcal{R}\}.$$

A set of assumptions $A \subseteq \mathcal{A}$ *is* closed *iff* $A = Cl(A)$*.*

In Example 2.3, $\{a, b\}$ is closed whereas $\{b\}$ is not.

[1] $\tau \notin \mathcal{L}$ represents "true" and stands for the empty body of rules. In other words, each rule with empty body can be interpreted as a rule with body τ for the purpose of presenting deductions as trees.

[2] Throughout, we often omit to specify the language $\mathcal{L}$, as it is implicit from the rules and assumptions. Also, if the contraries of assumptions are not explicitly defined, then they are assumed to be different from each other and any other explicitly mentioned sentences.

Note that, in some ABA frameworks, sets of assumptions are guaranteed to be closed. These ABA frameworks are referred to as *flat* and, as we will see later, exhibit additional properties than generic ABA frameworks.

Definition 2.5 *An ABA framework* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ *is* flat *iff for every* $A \subseteq \mathcal{A}$*,* A *is closed.*

The ABA framework in Example 2.3 is not flat, whereas the following is an example of a flat ABA framework.

Example 2.6 *An ABA framework with* $\mathcal{R} = \{r \leftarrow b, c, \quad q \leftarrow \quad , \quad p \leftarrow q, a\}$ *and* $\mathcal{A} = \{a, b, c\}$ *is guaranteed to be flat. Here, as in all flat ABA frameworks, deductions for assumptions can only be supported by an empty set of rules, e.g. there is a single deduction for* a*:*

$$\{a\} \overset{\{\}}{\vdash} a \ .$$

It is easy to see that if no assumption is the head of a rule, then an ABA framework is flat [Dung *et al.*, 2006]. However, an ABA framework can be flat even if some assumptions are heads of rules. For instance, in an ABA framework with $\mathcal{R} = \{a \leftarrow x\}$ and $\mathcal{A} = \{a\}$, the assumption a appears as the head of the rule $a \leftarrow x$, but since x is not deducible from any set of assumptions, all sets of assumptions in this ABA framework are guaranteed to be closed, and so the framework is flat. Note, however, that "dummy" rules such as $a \leftarrow x$ above, whose body is not deducible from any set of assumptions, could without loss of generality be deleted from ABA frameworks, as they generate no conclusions. On the other hand, the ABA framework in Example 2.3 has no such "dummy" rules and is not flat (as, indeed, $\{b\}$ is not closed).

The remaining desirable requirements met by sets of assumptions, as semantics for ABA frameworks, are given in terms of a notion of *attack* between sets of assumptions, defined as follows:

Definition 2.7 *A set of assumptions* $A \subseteq \mathcal{A}$ attacks *a set of assumptions* $B \subseteq \mathcal{A}$ *iff there are* $A' \subseteq A$ *and* $b \in B$ *such that* $A' \vdash \overline{b}$*.*

The following definitions of semantics for ABA are adapted from [Bondarenko *et al.*, 1993; Bondarenko *et al.*, 1997; Dung *et al.*, 2007].

Definition 2.8 *A set of assumptions (or* extension*) is* conflict-free *iff it does not attack itself. A set of assumptions/extension* $A \subseteq \mathcal{A}$ *is*

- admissible *iff it is closed, conflict-free and, for every* $B \subseteq \mathcal{A}$*, if* B *is closed and attacks* A*, then* A *attacks* B*;*

- preferred *iff it is maximally (w.r.t.* $\subseteq$*) admissible;*

- complete *iff it is admissible and contains all assumptions it* defends*, where A defends a iff for every $B \subseteq \mathcal{A}$, if B is closed and attacks $\{a\}$, then A attacks B;*
- stable *iff it is closed, conflict-free and, for every $a \notin A$, A attacks $\{a\}$;*
- well-founded *iff it is the intersection of all complete extensions;*
- ideal *iff A is maximal (w.r.t. $\subseteq$) such that*
  - *(i) it is admissible, and*
  - *(ii) for all preferred extensions $P \subseteq \mathcal{A}$, $A \subseteq P$.*

Note that ideal sets of assumptions were originally defined, in [Dung *et al.*, 2007], in the context of flat ABA frameworks only. The original definition naturally generalises to general, possibly non-flat, ABA frameworks as given above. Note also that, in the case of flat ABA frameworks, the term *grounded* is conventionally used instead of *well-founded* (e.g. in [Dung *et al.*, 2007]): we will adopt this convention too later in the chapter.

Example 2.9 *Consider a non-flat ABA framework with rules $\mathcal{R} = \{x \leftarrow c,\ z \leftarrow b,\ \ a \leftarrow b\}$, $\mathcal{A} = \{a, b, c\}$ and $\overline{a} = x$, $\overline{b} = y$, $\overline{c} = z$. Then, $\{c\}$ is closed and conflict-free. It is attacked by $\{b\}$, which cannot be counter-attacked but is not closed and thus can be disregarded; it is also attacked by the closed $\{a, b\}$, which is counter-attacked by $\{c\}$. Thus, $\{c\}$ is admissible, as well as preferred and complete. $\{\}$ is also admissible and complete, and thus well-founded, but not preferred. $\{b\}$ is not admissible, because it is not closed. Moreover, the closed $\{a, b\}$ is admissible because it is conflict-free and $\{b\}$ counter-attacks the closed $\{c\}$ which attacks $\{a, b\}$. Finally, $\{a, b\}$ is preferred and complete, and thus $\{\}$ is ideal.*

Note that a set of assumptions/extension can be seen as characterising the set of all sentences in the given ABA framework for which deductions exist supported by (subsets of) the extension:

Definition 2.10 *The* consequences *of an extension $A \subseteq \mathcal{A}$ is*

$$Cn(A) = \{\sigma \in \mathcal{L} \mid \exists\ A' \vdash \sigma,\ A' \subseteq A\}.$$

As an illustration, in Example 2.9, $Cn(\{c\}) = \{c, x\}$.

In the remainder of the paper, when a sentence belongs to the consequences of an admissible / preferred / stable / complete / well-founded / ideal extension we will say that it is admissible / preferred / stable / complete / well-founded / ideal, respectively. Thus, in Example 2.9, x is admissible.

The following properties on relationships amongst extensions according to various semantics hold for generic (possibly non-flat) ABA frameworks:

Theorem 2.11 *Let $A \subseteq \mathcal{A}$ be a set of assumptions.*

(i) If A is stable, then it is preferred.

(ii) If A is admissible, then there is some $P \subseteq \mathcal{A}$ such that P is preferred and $A \subseteq P$.

(iii) If A is stable, then it is complete.

(iv) If A is ideal and $S \subseteq \mathcal{A}$ is the intersection of all preferred extensions, then $A \subseteq S$.

(v) If A is the intersection of all preferred extensions and admissible, then it is ideal.

(vi) If A is ideal, then for each set of assumptions B attacking A there exists no admissible set of assumptions $B' \subseteq \mathcal{A}$ such that $B' \supseteq B$.

(vii) If A is well-founded, then for every $S \subseteq \mathcal{A}$, if S is stable, then $A \subseteq S$.

Proof.

(i) See proof of Theorem 4.6 in [Bondarenko *et al.*, 1997].

(ii) See proof of Theorem 4.9 in [Bondarenko *et al.*, 1997].

(iii) See proof of Theorem 5.5 in [Bondarenko *et al.*, 1997].

(iv) By definition, $A \subseteq P$ for every preferred $P \subseteq \mathcal{A}$, so $A \subseteq S$.

(v) The intersection of all preferred extensions A is a $\subseteq$-maximal set of assumptions that is contained in every preferred extension, so if A is in addition admissible, then it is by definition ideal.

(vi) Assume A is ideal and let B attack A. By contradiction, assume there exists an admissible $B' \supseteq B$. Then, by (ii) above, there is a preferred set of assumptions P such that $B' \subseteq P$. By definition of ideal extension, $A \subseteq P$, hence P is not conflict-free, contradicting its admissibility.

(vii) By definition, the well-founded extension is contained in every complete extension. Also, by (iii) above, every stable extension is complete. Therefore, the well-founded extension must be contained in every stable extension.

■

Note that item (v) was given and proven in [Dung *et al.*, 2007] (as Theorem 2.1(iv)) in the case of abstract argumentation frameworks [Dung, 1995].

The following properties on existence of extensions according to various semantics hold for generic (possibly non-flat) ABA frameworks.

Theorem 2.12

(i) If there is an admissible extension, then there is at least one preferred extension.

(ii) If the empty set of assumptions is closed, then there is at least one preferred extension.

(iii) If the empty set of assumptions is closed, then there exists an ideal extension.

Proof.

(i) Directly from Theorem 2.11(ii) (see also comments after the proof of Theorem 4.9 in [Bondarenko *et al.*, 1997]).

(ii) Directly from (i) above, as the empty set, if closed, is necessarily admissible (see also comments after the proof of Theorem 4.9 in [Bondarenko *et al.*, 1997]).

(iii) If $\{\}$ is closed, then it is admissible. So by (i) above, there is a preferred extension. Hence, the intersection S of preferred extensions exists too. Given that $\{\}$ is admissible, there must then be a $\subseteq$-maximal admissible subset of S, i.e. an ideal extension.

■

For a simple example of a (necessarily non-flat) ABA framework in which the empty set is not closed, consider $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ with $\mathcal{R} = \{a \leftarrow \;,\;\; x \leftarrow a\}$, $\mathcal{A} = \{a\}$ and $\overline{a} = x$: here, $\{\} \vdash a$, so that $\{\}$ is not closed; note also that no set is admissible, because any admissible set needs to be a closed superset of the empty set, and since there are deductions $\{\} \vdash a$ as well as $\{\} \vdash x$, where x is the contrary of a, no closed superset of $\{\}$ is conflict-free.

Flat ABA frameworks fulfil the following property, often referred to as the *Fundamental Lemma* (see e.g. [Dung, 1995; Bondarenko *et al.*, 1997]):

Theorem 2.13 *Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be a* flat *ABA framework, and let $A \subseteq \mathcal{A}$ be an admissible set of assumptions that defends assumptions $a, a' \in \mathcal{A}$. Then $A \cup \{a\}$ is admissible and defends a'.*

Proof. See proof of Theorem 5.7 in [Bondarenko *et al.*, 1997]. ■

Note that non-flat ABA frameworks need not in general fulfil the Fundamental Lemma: consider $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ with $\mathcal{R} = \{c \leftarrow a, b\}$, $\mathcal{A} = \{a, b, c\}$ and $\overline{a} = x$, $\overline{b} = y$, $\overline{c} = z$; it is non-flat, because $\{a, b\} \vdash c$; observe that both $\{a\}$ and $\{b\}$ are closed and unattacked, so, for instance, $\{a\}$ is admissible and defends b; however, $\{a, b\}$ is not closed, and so not admissible.

Flat ABA frameworks also fulfil additional properties concerning relationships between semantics, as follows:

Theorem 2.14 *Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be a flat ABA framework, and let $A \subseteq \mathcal{A}$ be a set of assumptions.*

(i) If A is preferred, then it is complete.

(ii) If A is grounded, then it is minimally (w.r.t. $\subseteq$) complete.

(iii) If A is grounded, then for every $P \subseteq \mathcal{A}$, if P is preferred, then $A \subseteq P$.

(iv) If A is ideal, then it is complete.

(v) If A is ideal and $G \subseteq \mathcal{A}$ is grounded, then $A \supseteq G$.

(vi) If A is maximally (w.r.t. $\subseteq$) complete, then it is preferred.

(vii) If A is admissible, then it is ideal iff for each set of assumptions B attacking A there exists no admissible set of assumptions $B' \subseteq \mathcal{A}$ such that $B' \supseteq B$.

Proof.

(i) Directly from Theorem 5.7 in [Bondarenko *et al.*, 1997], see Corollary 5.8 in [Bondarenko *et al.*, 1997].

(ii) See proof of Theorem 6.2 in [Bondarenko *et al.*, 1997].

(iii) See proof of Theorem 6.4 in [Bondarenko *et al.*, 1997].

(iv) Let I be ideal and suppose it defends $a \in \mathcal{A}$. Due to flatness, $I \cup \{a\}$ is admissible, and hence contained in every preferred extension. So $a \in I$ by $\subseteq$-maximality of I.

(v) Directly from (iv) and (ii) above.

(vi) If A was $\subseteq$-maximally complete but not preferred, then, by Theorem 1(ii), there would be some preferred yet not complete P such that $A \subsetneq P \subseteq \mathcal{A}$, contrary to (i) above.

(vii) See Theorem 3.3 in [Dung *et al.*, 2007].

■

Note that items (iv) and (v) were given and proven in [Dung *et al.*, 2007] (as items (ii) and (iii) respectively in Theorem 2.1), in the case of abstract argumentation frameworks. Also, (vii) was given and proven as Lemma 4(a) in [Dunne, 2009].

The following examples show that the properties in Theorem 2.14 may not hold, in general, for non-flat ABA frameworks.

Example 2.15 *Consider an ABA framework* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ *with* $\mathcal{R} = \{d \leftarrow c\}$, $\mathcal{A} = \{a, b, c, d\}$ *and* $\overline{a} = p$, $\overline{b} = a$, $\overline{c} = b$, $\overline{d} = d$. *Then* $\{a\}$ *is preferred and ideal, but not complete, as it defends c. (Cf. Theorem 2.14(i), (iv).) Note that* $\{a, c\}$ *is not admissible, as it is not closed whereas* $\{a, c, d\}$ *is closed, but not admissible as not conflict-free.*

In this example, there is no complete extension and thus no well-founded extension, and thus the ideal extension is not a superset of the well-founded extension.

Example 2.16 *Consider an ABA framework* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ *with* $\mathcal{R} = \{p \leftarrow a, \; p \leftarrow b, \; c \leftarrow \;\}$, $\mathcal{A} = \{a, b, c, d\}$ *and* $\overline{a} = b$, $\overline{b} = a$, $\overline{c} = d$, $\overline{d} = p$. *Here, the complete extensions are* $\{a, c\}$ *and* $\{b, c\}$, *and thus* $\{c\}$ *is well-founded, but it is not (minimally) complete, as it does not defend itself against (the attacking)* $\{d\}$. *(Cf. Theorem 2.14(ii).) This also shows that even if there is an admissible extension, there need not be an ideal extension.*

Example 2.17 *Consider an ABA framework* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ *with* $\mathcal{R} = \{d \leftarrow c, \; f \leftarrow e, \; p \leftarrow d, \; p \leftarrow e\}$, $\mathcal{A} = \{a, b, c, d, e, f\}$ *and* $\overline{a} = f$, $\overline{b} = a$, $\overline{c} = b$, $\overline{d} = p$, $\overline{e} = q$, $\overline{f} = a$. *Then* $\{e, f, b\}$ *is the only complete extension, and thus the well-founded extension. Moreover* $\{a\}$ *and* $\{e, f, b\}$ *are (the only) preferred extensions, and* $\{e, f, b\} \nsubseteq \{a\}$. *Therefore, there exists a preferred extension that does not contain the well-founded extension. (Cf. Theorem 2.14(iii).)*

Example 2.18 *Consider an ABA framework* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ *with* $\mathcal{R} = \{q \leftarrow a, \; r \leftarrow b, \; c \leftarrow q, r, \; z \leftarrow a, \; z \leftarrow b, \; z \leftarrow c\}$, $\mathcal{A} = \{a, b, c\}$ *and* $\overline{a} = c$, $\overline{b} = c$, $\overline{c} = z$. *Here, every* $A \subseteq \mathcal{A}$ *containing c is not conflict-free, so not admissible. Also,* $\{a, b\}$ *is not closed, so not admissible. However,* $\{a\}$ *is admissible, but not complete, as it defends b. Likewise* $\{b\}$ *is admissible, but not complete. Indeed, both* $\{a\}$ *and* $\{b\}$ *are preferred, yet not complete. Therefore,* $\{\}$ *is* $\subseteq$*-maximally complete, yet not preferred. (Cf. Theorem 2.14(vi).)*

Example 2.19 *Consider an ABA framework* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ *with* $\mathcal{R} = \{z \leftarrow c, \; c \leftarrow a, b\}$, $\mathcal{A} = \{a, b, c\}$ *and* $\overline{a} = x$, $\overline{b} = y$, $\overline{c} = z$. *Then* $\{a\}$ *is admissible (and preferred) and unattacked. Observe that* $\{a, x\}$ *is not closed, and* $\{a, c\}$, $\{x, c\}$, $\{a, x, c\}$ *are not conflict-free. So* $\{b\}$ *is preferred, yet* $\{a\} \nsubseteq \{b\}$, *so that A is not ideal. (Cf. Theorem 2.14(vii).)*

Flat ABA frameworks fulfil additional properties concerning existence of extensions w.r.t. various semantics, as follows:

Theorem 2.20 *Let* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ *be a* flat *ABA framework.*

(i) There is at least one preferred extension.

(ii) There is a unique ideal extension.

(iii) There is at least one complete extension.

(iv) There is a unique grounded extension and it is the least fixed point of Def, where, for $A \subseteq \mathcal{A}$, Def(A) $= \{a \in \mathcal{A} \mid A$ defends $a\}$.

Proof.

(i) Directly from the second item of Theorem 2.12, as, in the case of flat ABA frameworks, the empty set (like any other set of assumptions) is guaranteed to be closed (see also [Bondarenko *et al.*, 1997]).

(ii) Follows from Theorem 2.12(iii).

(iii) Directly from (i) above and Theorem 2.14(i).

(iv) See proof of Theorem 6.2 in [Bondarenko *et al.*, 1997].

■

Note that (ii) above was given and proven in [Dung *et al.*, 2007] in the case of abstract argumentation frameworks.

The following examples show that the properties in Theorem 2.20 may not hold, in general, for non-flat ABA frameworks.

Example 2.21 *Consider an ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ with $\mathcal{R} = \{a \leftarrow \}$, $\mathcal{A} = \{a\}$ and $\overline{a} = a$. Here, $\{\}$ is not closed and $\{a\}$ is not conflict-free. Thus, no set of assumptions is admissible. Hence, there is no preferred, complete, ideal or well-founded extension.*

Finally, consider an example which shows that, differently from flat ABA frameworks, in general, an ideal extension need not be unique for non-flat ABA frameworks.

Example 2.22 *Consider an ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ with assumptions $\mathcal{A} = \{a, a', b, b', c, d\}$, rules $\mathcal{R} = \{c \leftarrow a, a', \quad \overline{c} \leftarrow d, \quad \overline{d} \leftarrow a, b, \quad \overline{d} \leftarrow a', b', \quad a' \leftarrow a, b, c, \quad a \leftarrow a', b, c, \quad a' \leftarrow a, b', c, \quad a \leftarrow a', b', c\}$, and contraries $\overline{b} = b'$, $\overline{b'} = b$.[3] Here, $\{\}$ is closed, so admissible. There are two preferred extensions: $\{a, a', b, c\}$ and $\{a, a', b', c\}$. Their intersection $\{a, a', c\}$ is not admissible, because it cannot defend against (the closed attacking) $\{d\}$. Likewise, $\{a, c\}$ and $\{a', c\}$ are not admissible. Also, $\{a, a'\}$ is not closed. However, both $\{a\}$ and $\{a'\}$ are admissible, and hence ideal extensions.*

Note that additional properties hold for (generic and/or flat) ABA frameworks of restricted kinds, for instance, where "cycles" are not allowed (e.g. *stratified* and *order-consistent* ABA frameworks, see [Bondarenko *et al.*, 1997] for

[3] For readability, with an abuse of notation we may sometimes assume that the contraries (in this case, $\overline{c}$ and $\overline{d}$) of assumptions (in this case, c and d) are actually symbols in the language (different from other explicitly mentioned sentences).

details). Moreover, additional properties hold for other special classes of ABA frameworks, in addition to flat ABA frameworks, namely *normal* [Bondarenko *et al.*, 1997] and *simple* [Dimopoulos *et al.*, 2002] ABA frameworks (see [Bondarenko *et al.*, 1997; Dimopoulos *et al.*, 2002] for details).

3 ABA and non-monotonic reasoning

In this section we illustrate two instances of ABA for Non-Monotonic Reasoning, namely Autoepistemic Logic (AEL) [Moore, 1985] and Logic Programming (LP). The formal definitions of these instances, as well as correspondence results between the semantics for ABA as given in Definition 2.8 and their original semantics, can be found in [Bondarenko *et al.*, 1997; Schulz and Toni, 2015]

For illustration, as well as a running example throughout the chapter, we will use the following extract from the Nationwide[4] building society's 2016 policy for UK/EU Breakdown Assistance:

> COVERED FOR: UK/EU Breakdown Assistance for account holder(s) in any private car they are travelling in
>
> NOT COVERED FOR: private cars not registered to the account holder(s) unless the account holder(s) are in the vehicle at the time of the breakdown

We consider a person, Mary (denoted simply as m), who is an account holder travelling in a friend's car (denoted as c) when the car breaks down somewhere in the EU. In the remainder of this section we show how the application of the policy above to Mary's case can be represented in the AEL and LP instances of ABA, as given in general in [Bondarenko *et al.*, 1997]. In giving the concrete instantiations below we will use the following abbreviations: ah stands for "account holder"; tr stands for "travelling"; pr stands for "private vehicle"; cov stands for "covered"; reg stands for "registered"; cov' stands for "there is an exception to being not covered".

[4] `www.nationwide.co.uk`

3.1 Breakdown Assistance policy in the AEL instance of ABA

The application of the Breakdown Assistance policy to Mary's case can be represented in the AEL instance of ABA as follows:

$\mathcal{L} =$ a modal language containing a modal operator L (where $L\sigma$ stands for "σ is believed") as well as atoms $ah(m),\ tr(m,c),\ pr(c),\ cov(m,c), reg(c,m),\ cov'(m,c), in(m,c)$

$\mathcal{R} =$ a complete set of inference rules of classical logic for $\mathcal{L}$ together with the following inference rules (all with an empty body):

$$\overline{ah(m) \wedge tr(m,c) \wedge pr(c) \wedge \neg L \neg cov(m,c) \rightarrow cov(m,c)}$$

$$\overline{\neg reg(c,m) \wedge \neg L cov'(m,c) \rightarrow \neg cov(m,c)}$$

$$\overline{in(m,c) \rightarrow cov'(m,c)} \quad \overline{ah(m)} \quad \overline{tr(m,c)}$$

$$\overline{pr(c)} \quad \overline{\neg reg(c,m)} \quad \overline{in(m,c)}$$

$\mathcal{A} =$ $\{L\sigma, \neg L\sigma \mid \sigma \in \mathcal{L}\}$

$\overline{L\sigma} = \neg L\sigma$ and $\overline{\neg L\sigma} = \sigma$ for any $\sigma \in \mathcal{L}$

Note that, in this ABA framework, $\mathcal{R}$ includes domain-independent rules, e.g.

$$\frac{\sigma_1 \wedge \sigma_2}{\sigma_1} \text{ for any } \sigma_1, \sigma_2 \in \mathcal{L},$$

as well as domain-specific rules, e.g.

$$\overline{in(m,c)}.$$

Note also that this ABA framework (as well as any other AEL instance of ABA) is not flat [Bondarenko *et al.*, 1997], as, for instance, the set of assumptions $\{Lcov(m), \neg Lcov(m)\}$ is not closed, because it is classically inconsistent. Nonetheless, for this instance, the empty set of assumptions is closed.

Given this representation in ABA, the problem of determining whether Mary should be covered or not amounts to determining whether $cov(m)$ is stable (following the conventional AEL approach of determining whether $cov(m)$ belongs to a consistent stable expansion [Moore, 1985] of the theory consisting of the heads of the domain-specific part of $\mathcal{R}$), or preferred, or well-founded etc. (by adopting any of the other ABA semantics). In this particular example, all ABA semantics agree that Mary should be covered, by assuming $\neg L \neg cov(m,c)$, in agreement with the original semantics of AEL, as predicted by the general correspondence Theorem 3.18 in [Bondarenko *et al.*, 1997] and the fact that, in this example, all ABA semantics agree with the semantics of stable extensions. As an illustration, $\{\neg L \neg cov(m,c)\}$ is admissible, since it is conflict-free, closed and the (closed) set of assumptions $\{\neg Lcov'(m,c)\}$ attacking it, as well as all its (closed) supersets, are attacked by the (closed) empty set of assumptions.

Note that, in the AEL instance of ABA, beliefs, of the form $L\,\sigma$ or $\neg L\,\sigma$, are assumptions that may occur as heads of rules. For example, the earlier AEL instance of ABA may be extended so that $\mathcal{R}$ includes also

$$\overline{\qquad\qquad}\\ L\,ah(m)$$

to represent that Mary is believed to be an account holder. This kind of knowledge cannot be directly represented in flat ABA.

3.2 Breakdown Assistance policy in the LP instance of ABA

The application of the Breakdown Assistance policy to Mary's case can be represented in the LP instance of ABA as follows:

$$\begin{array}{rl}
\mathcal{R} = & \{cov(m,c) \leftarrow ah(m), tr(m,c), pr(c), not\ \neg cov(m,c),\\
& \neg cov(m,c) \leftarrow \neg reg(c,m), not\ cov'(m,c),\\
& cov'(m,c) \leftarrow in(m,c),\\
& ah(m) \leftarrow,\quad tr(m,c) \leftarrow,\quad pr(c) \leftarrow,\quad \neg reg(c,m) \leftarrow,\quad in(m,c) \leftarrow\}\\
\mathcal{A} = & \{not\ p(t_1,t_2),\ not\ q(t) \mid p \in \{cov, tr, \neg cov, \neg reg, cov', in\},\\
& \qquad\qquad\qquad\qquad\qquad q \in \{ah, pr\},\\
& \qquad\qquad\qquad\qquad\qquad t_1, t_2, t \in \{m, c\}\}\\
\mathcal{L} = & \mathcal{A} \cup \{x \mid not\ x \in \mathcal{A}\}\\
\overline{not\,x} = & x \text{ for any } not\ x \in \mathcal{A}
\end{array}$$

So, $\mathcal{L}$ is the Herbrand base of (the logic program) $\mathcal{R}$ together with all negation as failure (NAF) literals that can be built from this Herbrand base, and $\mathcal{A}$ is the set of all these NAF literals. Note that in this illustration we treat $\neg cov$ and $\neg reg$ as predicate symbols.

Given this representation in ABA, the problem of determining whether Mary should be covered or not amounts to determining, for instance, whether $cov(m,c)$ is stable (following the stable model semantics [Gelfond and Lifschitz, 1988] for $\mathcal{R}$, seen as a logic program, by virtue of the correspondence Theorem 3.13 in [Bondarenko *et al.*, 1997]), or admissible/preferred (following the preferred extension semantics [Dung, 1991] for $\mathcal{R}$, by virtue of the correspondence Theorem 4.5 in [Bondarenko *et al.*, 1997]), or grounded (following the well-founded model semantics [Gelder *et al.*, 1991] for $\mathcal{R}$, by virtue of the correspondence Theorem 3.13 in [Bondarenko *et al.*, 1997]), or ideal (following the scenario semantics [Alferes *et al.*, 1993] for $\mathcal{R}$ In this particular example, all ABA semantics agree that Mary should be covered, by assuming $not\ cov(m,c)$. As an illustration, $\{not\ cov(m,c)\}$ is admissible, since it is conflict-free and the set of assumptions $\{not\ cov'(m,c)\}$ attacking it, as well as all its supersets, are attacked by the empty set of assumptions.

4 ABA versus abstract argumentation

In this section we focus on the relationship between flat ABA frameworks and Abstract Argumentation (AA) frameworks [Dung, 1995]. In particular, flat ABA is an instance of AA, under all semantics considered in this paper and, conversely, AA is an instance of (flat) ABA.

Flat ABA frameworks are instances of AA frameworks where arguments are deductions supported by sets of assumptions and attacks are defined by appropriately lifting the notion of attack between sets of assumptions to a notion of attack between arguments [Dung *et al.*, 2007; Toni, 2012].

Definition 4.1 *Let $\mathcal{ABA} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be a flat ABA framework.*

- *An* argument *for $\sigma \in \mathcal{L}$ supported by $A \subseteq \mathcal{A}$ and $R \subseteq \mathcal{R}$, denoted $A \vdash^{R}_{arg} \sigma$ (or simply $A \vdash_{arg} \sigma$), is such that there is a deduction $A \vdash^{R} \sigma$.*
- *An argument $A \vdash_{arg} \sigma$* attacks *an argument $B \vdash_{arg} \pi$ iff there is $b \in B$ such that $\sigma = \overline{b}$.*

Then $\mathcal{AA}(\mathcal{ABA}) = (Args, attack)$ is the corresponding AA framework *of $\mathcal{ABA}$ with $Args$ the set of all arguments (as in the first bullet) and $attack$ the set of all pairs $(\mathtt{a}, \mathtt{b})$ such that $\mathtt{a}, \mathtt{b} \in Args$ and $\mathtt{a}$ attacks $\mathtt{b}$ (as in the second bullet).*

Note that $Args$ contains an argument for every assumption in $\mathcal{A}$ as illustrated by the following example.

Example 4.2 *Consider an ABA framework $\mathcal{ABA}$ with rules and assumptions as in Example 2.6 and $\overline{a} = r$, $\overline{b} = q$, $\overline{c} = p$. Then $\mathcal{AA}(\mathcal{ABA})$ is $(Args, attack)$ with $Args = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{p}, \mathtt{q}, \mathtt{r}\}$ where $\mathtt{a} = \{a\} \vdash_{arg} a$, $\mathtt{b} = \{b\} \vdash_{arg} b$, $\mathtt{c} = \{c\} \vdash_{arg} c$, $\mathtt{p} = \{a\} \vdash_{arg} p$, $\mathtt{q} = \{\} \vdash_{arg} q$, $\mathtt{r} = \{b, c\} \vdash_{arg} r$, and $attack = \{(\mathtt{p}, \mathtt{c}), (\mathtt{p}, \mathtt{r}), (\mathtt{q}, \mathtt{b}), (\mathtt{q}, \mathtt{r}), (\mathtt{r}, \mathtt{a}), (\mathtt{r}, \mathtt{p})\}$.*

The semantics of an AA framework corresponding to a flat ABA framework can be determined using the standard AA semantics [Dung, 1995; Dung *et al.*, 2007]. For all ABA semantics considered in this paper, the semantics of a flat ABA framework corresponds to the semantics of its corresponding AA framework, as follows:

Theorem 4.3 *Let $\mathcal{ABA} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be a flat ABA framework and let $\mathcal{AA}(\mathcal{ABA})$ be its corresponding AA framework.*

(i) If a set of assumptions $A \subseteq \mathcal{A}$ is admissible / preferred / stable / complete / grounded / ideal in $\mathcal{ABA}$, then the union of all arguments supported by any $A' \subseteq A$ is admissible / preferred / stable / complete / grounded / ideal, respectively, in $\mathcal{AA}(\mathcal{ABA})$.

(ii) The union of all sets of assumptions supporting the arguments in an admissible / preferred / stable / complete / grounded / ideal set of arguments in $\mathcal{AA}(\mathcal{ABA})$ is admissible / preferred / stable / complete / grounded / ideal, respectively, in $\mathcal{ABA}$.

Proof. See the proof of Theorem 2.2 in [Dung *et al.*, 2007] for admissible, grounded and ideal extensions, the proof of Theorem 1 in [Toni, 2012]for stable extensions[5], and the proof of Theorem 6.1 and 6.3 [Caminada *et al.*, 2015] for complete and preferred extensions respectively. ■

Note that for the preferred, stable, complete, grounded, and ideal semantics the correspondence between the extensions of a flat ABA framework and the extensions of the corresponding AA framework is one-to-one. For the admissible semantics, instead, the correspondence is one-to-many, i.e. the union of all sets of assumptions supporting the arguments in an admissible extension may be the same for various admissible extensions of the corresponding AA framework, as illustrated in the following example.

Example 4.4 *The ABA framework from Example 4.2 has two admissible extensions: $\{\}$ and $\{a\}$. In contrast, the corresponding AA framework has five admissible extensions: $\mathtt{A}_1 = \{\}$, $\mathtt{A}_2 = \{\mathtt{q}\}$, $\mathtt{A}_3 = \{\mathtt{p}\}$, $\mathtt{A}_4 = \{\mathtt{p},\mathtt{q}\}$, $\mathtt{A}_5 = \{\mathtt{p},\mathtt{a}\}$, $\mathtt{A}_6 = \{\mathtt{q},\mathtt{a}\}$, $\mathtt{A}_7 = \{\mathtt{p},\mathtt{q},\mathtt{a}\}$. However, the union of all sets of assumptions supporting the arguments in $\mathtt{A}_1$ and $\mathtt{A}_2$ is $\{\}$, so both correspond to the first admissible extension of the ABA framework. Similarly, the union of all sets of assumptions supporting arguments in the other admissible extensions ($\mathtt{A}_3 - - \mathtt{A}_7$) of the AA framework is $\{a\}$, so they all correspond to the second admissible extension of the ABA framework.*

Theorem 4.3 shows that, under the semantics considered therein, flat ABA frameworks are an instance of AA frameworks and the semantics of ABA can alternatively be defined in terms of extensions as sets of arguments, as in [Dung *et al.*, 2007], rather than in terms of extensions as sets of assumptions, as in [Bondarenko *et al.*, 1993; Bondarenko *et al.*, 1997]. This implies, for example, that existing machinery for computing extensions of AA frameworks can be used to compute extensions of ABA frameworks whose corresponding AA frameworks are finite. Conversely, as shown below, AA frameworks are an instance of flat ABA frameworks, that is any AA framework can be translated into a corresponding flat ABA framework such that their respective extensions correspond [Toni, 2012]. This implies, in particular, that existing machinery for determining whether sentences are admissible / preferred / complete / grounded / ideal in flat ABA (see Section 5) can be used to determine whether arguments in an AA framework belong to an admissible / preferred / complete / grounded / ideal extension. The ABA framework corresponding to an AA

[5]The proof of Theorem 1 in [Toni, 2012] actually considers a different notion of stable extension, but can naturally be modified to prove the result indicated here.

framework has the arguments in the AA framework as (the only) assumptions and appropriate notions of contraries of these assumptions and rules to encode the attacks between the arguments in the original AA framework, as follows:

Definition 4.5 *Let $\mathcal{AA} = (Args, attack)$ be an AA framework. The* corresponding ABA framework *of $\mathcal{AA}$ is $\mathcal{ABA}(\mathcal{AA}) = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ with*

- $\mathcal{A} = Args$;
- $\mathcal{L} = \mathcal{A} \cup \{\mathtt{a}^c \mid \mathtt{a} \in \mathcal{A}\}$;
- *for all* $\mathtt{a} \in \mathcal{A}$: $\overline{\mathtt{a}} = \mathtt{a}^c$;
- $\mathcal{R} = \{\mathtt{b}^c \leftarrow \mathtt{a} \mid (\mathtt{a}, \mathtt{b}) \in attack\}$.

Note that clearly the corresponding ABA framework of any AA framework is flat since assumptions never occur in the head of a rule, by construction.

Since the set of arguments in a given AA framework coincides with the set of assumptions of the corresponding ABA framework, there is a straightforward one-to-one correspondence between all semantics of the AA and ABA framework.

Theorem 4.6 *Let $\mathcal{AA} = (Args, attack)$ be an AA framework and let $\mathcal{ABA}(\mathcal{AA})$ be its corresponding ABA framework.*

(i) If $\mathtt{A} \subseteq Args$ is admissible / preferred / stable / complete / grounded / ideal in $\mathcal{AA}$, then $\mathtt{A}$ is admissible / preferred / stable / complete / grounded / ideal, respectively, in $\mathcal{ABA}(\mathcal{AA})$.

(ii) If $A \subseteq \mathcal{A}$ is admissible / preferred / stable / complete / grounded / ideal set of arguments in $\mathcal{ABA}(\mathcal{AA})$, then A is admissible / preferred / stable / complete / grounded / ideal, respectively, in $\mathcal{AA}$.

Proof. See proof of Theorem 2 in [Toni, 2012] for admissible. As noted in [Toni, 2012], the proof for other semantics is similar. ■

Example 4.7 *Consider the AA framework $\mathcal{AA}$ with $Args = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and $attack = \{(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{a}), (\mathtt{b}, \mathtt{c})\}$. The corresponding ABA framework is $\mathcal{ABA}(\mathcal{AA})$ with $\mathcal{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $\overline{\mathtt{a}} = \mathtt{a}^c$, $\overline{\mathtt{b}} = \mathtt{b}^c$, $\overline{\mathtt{c}} = \mathtt{c}^c$, and $\mathcal{R} = \{\mathtt{b}^c \leftarrow \mathtt{a}, \quad \mathtt{a}^c \leftarrow \mathtt{b}, \mathtt{c}^c \leftarrow \mathtt{b}\}$. The admissible extensions of $\mathcal{AA}$ are $\{\}$, $\{\mathtt{a}\}$, $\{\mathtt{b}\}$, and $\{\mathtt{a}, \mathtt{c}\}$, which are exactly the admissible extensions of $\mathcal{ABA}(\mathcal{AA})$. Correspondence as dictated by Theorem 4.6 hold for the other semantics considered therein too.*

5 Dispute trees, dispute derivations and ABA dialogues

In this section we overview the main existing computational machinery for flat ABA frameworks, allowing to determine whether sentences are admissible (and therefore preferred, by Theorem 2.11 (ii), and complete, by Theorem 2.11 (ii) and Theorem 2.14 (i)), grounded, or ideal.[6] This machinery is based on the computation of *dispute trees* (overviewed in Section 5.1), using *dispute derivations* (illustrated in Section 5.2) that, in particular, can be executed amongst agents to form *ABA dialogues* (illustrated in Section 5.3).

5.1 Dispute trees

Dispute trees [Dung *et al.*, 2006; Dung *et al.*, 2007] provide an abstraction of the problem of determining whether arguments in AA frameworks belong to an admissible / grounded / ideal extension. Since flat ABA frameworks correspond to special instances of AA frameworks (see Section 4), dispute trees can be used to determine whether sentences are admissible / grounded / ideal, respectively, as well as identifying assumptions in admissible / grounded / ideal extensions for ABA, respectively, supporting arguments for these sentences. Dispute trees can be defined abstractly for any abstract argumentation framework as follows:

Definition 5.1 *Let* $(Args, attack)$ *be any abstract argumentation framework. A* dispute tree *for* $\mathtt{a} \in Args$ *is a tree* $\mathcal{T}$ *such that:*

(i) *every node of* $\mathcal{T}$ *is of the form* $[\mathtt{L}:\mathtt{x}]$, *with* $\mathtt{L} \in \{\mathtt{P}, \mathtt{O}\}$, $\mathtt{x} \in Args$*: the node is* labelled *by argument* $\mathtt{x}$ *and assigned the* status *of either* proponent ($\mathtt{P}$) *or* opponent ($\mathtt{O}$)*;*

(ii) *the root of* $\mathcal{T}$ *is a* $\mathtt{P}$ *node labelled by* $\mathtt{a}$*;*

(iii) *for every* $\mathtt{P}$ *node* n, *labelled by some* $\mathtt{b} \in Args$, *and for every* $\mathtt{c} \in Args$ *such that* $\mathtt{c}$ *attacks* $\mathtt{b}$, *there exists a child of* n, *which is an* $\mathtt{O}$ *node labelled by* $\mathtt{c}$*;*

(iv) *for every* $\mathtt{O}$ *node* n, *labelled by some* $\mathtt{b} \in Args$, *there exists exactly one child of* n *which is a* $\mathtt{P}$ *node labelled by some* $\mathtt{c} \in Args$ *such that* $\mathtt{c}$ *attacks* $\mathtt{b}$*;*

(v) *there are no other nodes in* $\mathcal{T}$ *except those given by 1–4.*

The defence set *of a dispute tree* $\mathcal{T}$, *denoted by* $\mathcal{D}(\mathcal{T})$, *is the set of all arguments labelling* $\mathtt{P}$ *nodes in* $\mathcal{T}$*.*

[6] In general, this machinery cannot be used to determine whether a sentence is stable, as this requires the computation of a full extension, as discussed in [Dung *et al.*, 2002]. However, for restricted types of flat ABA frameworks whose preferred extensions are guaranteed to be stable, determining whether a sentence is admissible amounts to determining whether it is stable, too.

Example 5.2 *Given the AA framework with* $Args = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}, \mathtt{g}\}$ *and* $attack = \{(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{c}), (\mathtt{d}, \mathtt{e}), (\mathtt{d}, \mathtt{f}), (\mathtt{e}, \mathtt{d}), (\mathtt{e}, \mathtt{f}), (\mathtt{f}, \mathtt{g}), (\mathtt{g}, \mathtt{f})\}$*, consider the trees in the figure below. The tree on the left is not a dispute tree since an opponent node is a leaf node, thus violating condition (iv) in Definition 5.1. In contrast, the trees in the middle and on the right satisfy all conditions and are thus dispute trees for* `c` *and* `d`*, respectively.*

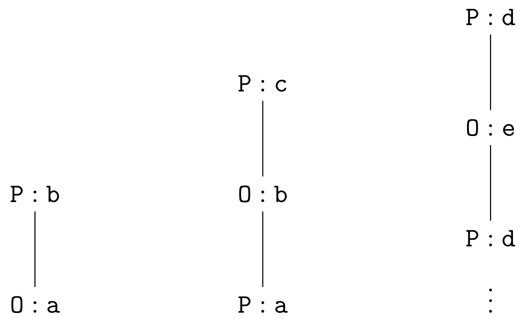


Figure 1. Only the middle and right of the three trees are dispute trees.

In order to help determine membership of arguments in admissible / grounded / ideal extensions of AA frameworks, dispute trees need to fulfil special requirements, as follows:

Definition 5.3 *Let* $(Args, attack)$ *be any abstract argumentation framework. A dispute tree* $\mathcal{T}$ *(for some argument in Args) is*

- admissible *iff no argument in* $\mathcal{T}$ *labels both* `P` *and* `O` *nodes;*
- grounded *iff it is finite;*
- ideal *iff for no argument* `a` *in* $\mathcal{T}$ *labelling an* `O` *node there exists an admissible dispute tree for* `a`.

Example 5.4 *Consider again the AA framework from Example 5.2. The dispute tree shown in the middle of Figure 1 is admissible since no argument labels both a proponent and an opponent node, as well as grounded since it is finite. Furthermore, the dispute tree is ideal since its only opponent node is labelled with* `b` *and there are no dispute trees for* `b`*, and thus there are no admissible dispute trees for* `b`*. The dispute tree for* `d` *on the right of Figure 1 is admissible, but not grounded since it is infinite. It is furthermore not ideal since there is an admissible dispute tree for* `e` *(obtained by exchanging* `d` *and* `e` *in the dispute tree for* `d` *on the right of Figure 1).*

The left of Figure 2 gives an example of a dispute tree which is ideal but not grounded. The opponent nodes of this dispute tree are all labelled by argument `f`.

Since the only dispute tree for f *is the one displayed in the middle of Figure 2, which is not an admissible dispute tree since argument* d *(as well as* e*) labels both an opponent and a proponent node, the dispute tree for* g *on the left of Figure 2 is ideal. Note that there are other admissible dispute trees for* g *which are not ideal. For example the one on the right of Figure 2 is not ideal since there exists an admissible dispute tree for* e*.*

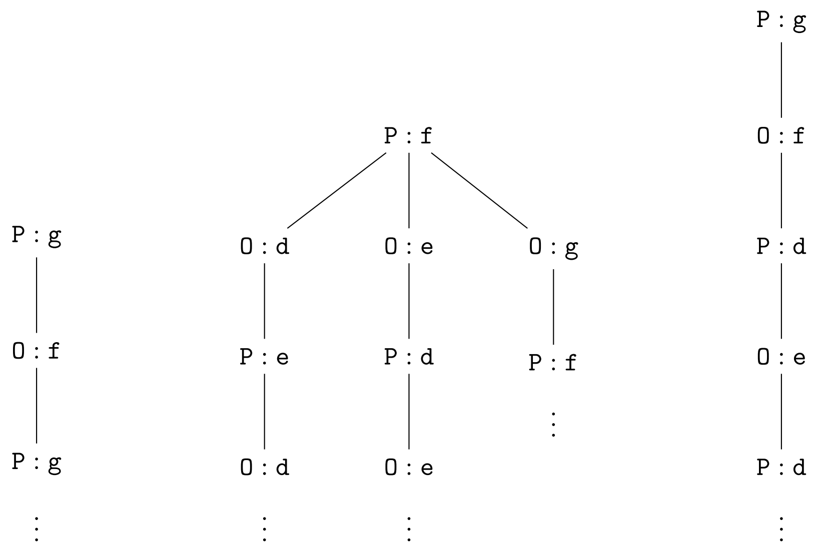


Figure 2. Three dispute trees constructed from the AA framework in Example 5.2.

Theorem 5.5 *Let* $(Args, attack)$ *be any abstract argumentation framework.*

(i) If $\mathcal{T}$ *is an admissible dispute tree for an argument* a *then the defence set of* $\mathcal{T}$ *is admissible.*

If $\mathtt{a} \in \mathtt{A}$ *for some admissible set of arguments* $\mathtt{A} \subseteq Args$ *then there exists an admissible dispute tree for* a *with defence set* $\mathtt{A}'$ *such that* $\mathtt{A}' \subseteq \mathtt{A}$ *and* $\mathtt{A}'$ *is admissible.*

(ii) If $\mathcal{T}$ *is an ideal dispute tree for an argument* a *then the defence set* A *of* $\mathcal{T}$ *is such that* A *is admissible and* $\mathtt{A} \subseteq I$ *with* I *the ideal extension of* $(Args, attack)$*.*

If $\mathtt{a} \in I$ *with* I *the ideal extension of* $(Args, attack)$*, then there exists an ideal dispute tree for* a *with defence set* A *and* $\mathtt{A} \subseteq I$*.*

(iii) If $\mathcal{T}$ *is a grounded dispute tree for an argument* a *then the defence set* A *of* $\mathcal{T}$ *is such that* A *is admissible and* $\mathtt{A} \subseteq G$ *with* G *the grounded extension of* $(Args, attack)$*.*

If $\mathtt{a} \in G$ *with* G *the grounded extension of* $(Args, attack)$*, then there exists a grounded dispute tree for* $\mathtt{a}$ *with defence set* $\mathtt{A}$ *and* $\mathtt{A} \subseteq G$*.*

Proof.

(i) See proof of Theorem 3.2 in [Dung *et al.*, 2007].

(ii) See proof of Theorem 3.4 in [Dung *et al.*, 2007].

(iii) Follows directly from Theorem 3.7 in [Kakas and Toni, 1999].

■

Example 5.6 *As discussed in Example 5.4, the dispute tree in the middle of Figure 1 is admissible and grounded. As stated in Theorem 5.5 the defence set,* $\{\mathtt{a}, \mathtt{c}\}$*, is admissible and is a subset of the grounded extension of the AA framework from Example 5.2, in fact in this case it coincides with the grounded extension. The ideal extension of the AA framework is* $\{\mathtt{a}, \mathtt{c}, \mathtt{g}\}$ *and we saw that there exists an ideal dispute tree for* $\mathtt{g}$ *(on the left of Figure 2) whose defence set is* $\{\mathtt{g}\}$*, which is a subset of the ideal extension.*

In order to determine whether a sentence is admissible / grounded / ideal, given a flat ABA framework, a dispute tree for an argument for that sentence can be used, by virtue of the correspondence results overviewed in Section 4 and Theorem 5.5 above. For example, given the ABA framework in Section 3.2, the dispute tree in Figure 3 for argument $\{not\,\neg cov(m,c)\} \vdash_{arg} cov(m,c)$ can be used to determine that $cov(m,c)$ is admissible, grounded and ideal. Indeed, this is a dispute tree since the leaf node cannot be attacked and no other opponent node can attack the root. Moreover, it is trivially admissible and, since it is finite, it is grounded. Finally, it is ideal as no admissible dispute tree for its only opponent node exists (as $\{\} \vdash_{arg} cov'(m,c)$ cannot be attacked).

$$\mathtt{P} : \{not\,\neg cov(m,c)\} \vdash_{arg} cov(m,c)$$

$$\mathtt{O} : \{not\; cov'(m,c)\} \vdash_{arg} \neg cov(m,c)$$

$$\mathtt{P} : \{\} \vdash_{arg} cov'(m,c)$$

Figure 3. A dispute tree for $\{not\,\neg cov(m,c)\} \vdash_{arg} cov(m,c)$ for the flat ABA framework in Section 3.2.

5.2 Dispute derivations

Dispute derivations [Dung *et al.*, 2006; Dung *et al.*, 2007; Toni, 2013; Craven and Toni, 2016] are algorithms for determining whether a given sentence, in the language of a flat ABA framework, is admissible, grounded or ideal. Different kinds of dispute derivations can be defined for the different semantics, as in [Dung *et al.*, 2006; Dung *et al.*, 2007], or the same template of dispute derivations can be instantiated differently for the different semantics, as in [Toni, 2013; Craven and Toni, 2016] and, for the LP instance of ABA, in [Kakas and Toni, 1999]. Dispute derivations for determining whether a sentence is admissible can also be used to determine whether the sentence is complete or preferred [Toni, 2013]. All given notions of dispute derivations are defined as games between (fictional) *proponent* (`P`) and *opponent* (`O`) players, as for dispute trees. All given notions are sound and, for restricted types of flat ABA frameworks (referred to as *p-acyclic* [Dung *et al.*, 2006]), complete [Dung *et al.*, 2006; Dung *et al.*, 2007; Toni, 2013]. The most recently defined types of dispute derivations are complete in general [Craven and Toni, 2016], for the admissible and grounded semantics. Different types of dispute derivations also differ in the data structures they deploy as well as their outputs:

- the dispute derivations of [Dung *et al.*, 2006; Dung *et al.*, 2007] deploy sets of assumptions and output admissible sets of assumptions in all cases, and, in the case of grounded/ideal semantics, these sets of assumptions are contained in the grounded/ideal extension, respectively;

- the dispute derivations of [Toni, 2013; Craven and Toni, 2016] deploy a mixture of sets of assumptions and sets of *potential arguments*, i.e. deductions supported by any sets of sentences (rather than assumptions) and with sentences in the support possibly marked as "seen", and output admissible sets of assumptions in all cases, as for the previous types of dispute derivations, as well as dialectical structures from which admissible / grounded / ideal dispute trees can be obtained.

We illustrate dispute derivations for the LP instance of ABA representation of the Breakdown Assistance policy, in Section 3.2, and refer to the original papers for formal definitions and results. In the illustration, we focus on the dispute derivations of [Toni, 2013], since they are generalisations of the earlier dispute derivations of [Dung *et al.*, 2006; Dung *et al.*, 2007] but still in the same spirit. Instead, the dispute derivations of [Craven and Toni, 2016] are based on a different conceptual model for ABA, where arguments and sets of arguments are defined as graphs instead (see [Craven and Toni, 2016] for details).

The (flat) ABA framework of Section 3.2 can be used to determine whether Mary should be covered, by determining whether $cov(m, c)$ is admissible (and thus, for this particular ABA framework, grounded, ideal etc.), i.e. if it belongs to an admissible extension. This can be determined in turn by means of a dispute derivation for $cov(m, c)$. This dispute derivation starts with a *potential* *argument* by P:[7]

$$\{\} \vdash^p_{\{cov(m,c)\}} cov(m,c),$$

namely a deduction $\{cov(m,c)\} \vdash cov(m,c)$ with no sentence in the support $\{cov(m,c)\}$ marked as "seen" (and the sentence $cov(m,c)$ in the support still "unseen"). In the first step of the derivation, then, P needs to "expand" its potential argument, while O is watching and can only put forward new potential arguments when P has sufficiently expanded its own potential arguments so as to have identified assumptions in their "unseen" support that O can attack (automatically rendering them "seen"). In this simple illustration, P will necessarily expand the initial potential argument to

$$\{\} \vdash^p_{\{ah(m),tr(m,c),pr(c),not\,\neg cov(m,c)\}} cov(m,c)$$

and identify the assumption $not\,\neg cov(m,c)$ as an element of the *defence set* of the dispute tree that the dispute derivation will output (if successful). At this stage O may opt to *eagerly* attack this assumption or *patiently* wait for P to carry on "expanding" its potential argument until it becomes an *actual argument*. This choice for O (and, in an analogous situation, for P) is dictated by the *selection function*, a parameter in the definition of dispute derivations. Whichever this selection function, at some later stage in the derivation the initial potential argument by P will become the *actual argument*

$$\{not\,\neg cov(m,c)\} \vdash^p_{\{\}} cov(m,c) \qquad (\mathtt{P}_{cov})$$

attacked by a potential argument by O

$$\{not\,cov'(m,c)\} \vdash^p_U \neg cov(m,c) \qquad (\mathtt{O}_{\neg cov}(U))$$

where, depending on the selection function, U may be as follows:

- $U = \{\neg reg(c,m)\}$, or
- $U = \{\}$.

In both cases, at some earlier stage, P will have chosen $not\,cov'(m,c)$, in the "unseen" support of a potential argument by O, as a *culprit*, causing that assumption to be marked as "seen" from that stage onwards. Note that O's potential argument $\mathtt{O}_{\neg cov}(U)$, whichever U, is necessarily obtained by "expanding" the potential argument

$$\{\} \vdash^p_{\{\neg cov(m,c)\}} \neg cov(m,c)$$

[7] In general, a potential argument is of the form $A \vdash^p_S \sigma$, for $A \subseteq \mathcal{A}$, $S \subseteq \mathcal{L}$, and $\sigma \in \mathcal{L}$, where the superscript p stands for "*potential*". Given $A \vdash^p_S \sigma$, there is a deduction for σ supported by $A \cup S$ (and some set of rules), with S the set of "unseen" sentences in this support and A the set of "seen" assumptions, as illustrated later.

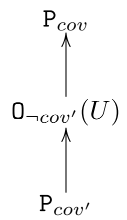
put forward earlier by O to attack P's defence set element $not\,\neg cov(m, c)$. Also, when P "sees" $not\,cov'(m, c)$ and chooses it as a culprit in $\mathtt{O}_{\neg cov}(U)$, it creates a potential argument

$$\{\} \vdash^{p}_{\{cov'(m,c)\}} cov'(m, c)$$

which is later "expanded" to

$$\{\} \vdash^{p}_{\{\}} cov'(m, c). \qquad (\mathtt{P}_{cov'})$$

Since O cannot possibly attack this argument, the derivation terminates successfully, returning, as output, the defence set $\{not\,\neg cov(m, c)\}$ as well as the dialectical structure



from which the dispute tree in Figure 3 is obtained.

In general, the defence set and the set of culprits are used to perform various kinds of *filtering* to save computation (to prevent players from attacking assumptions they have already attacked) as well as to guarantee that the computed defence set is conflict-free. Different semantics require different combinations of these filtering mechanisms. Moreover, the ideal semantics requires additional subcomputation to guarantee that the dispute tree is indeed ideal (namely that there exists no admissible dispute tree for the argument held at any opponent node).

5.3 ABA dialogues

ABA dialogues, as given in [Fan and Toni, 2014b; Fan and Toni, 2012a; Fan and Toni, 2011a], can be viewed as a distributed computation of dispute trees amongst agents, holding different ABA frameworks, but with the same underlying language $\mathcal{L}$.[8] An ABA dialogue is a sequence of *utterances*. The *content* of utterances may be a rule, an assumption, a contrary, or a claim whose "acceptability" (under admissible / grounded / ideal semantics) needs to be ascertained. The dialogue model can be used to support several dialogue types, e.g. information seeking and persuasion [Fan and Toni, 2011c; Fan and Toni, 2012a; Fan and Toni, 2012c; Fan *et al.*, 2014].

[8] Here, as in [Gaertner and Toni, 2008], we (equivalently) define the contrary of an assumption as a total mapping from an assumption to a (non-empty) set of sentences, instead of a mapping from an assumption to a sentence as in the original ABA. This lends itself better to a dialogical setting, as agents may hold different sentences as contrary to the same assumption.

Syntactically, given two agents a_i and a_j, let $\mathcal{ID}$ be a (non-empty, possibly infinite) set that is totally ordered, with the ordering given by $<$, and contains a special element ID_0 which is the least element w.r.t. $<$. Then, utterances are denoted as tuples:

$$\langle a_i, a_j, T, C, \text{ID} \rangle,$$

where

- a_i is the agent making this utterance;
- a_j is the recipient;
- C (the *content*) is of one of the following forms:
  - $claim(\chi)$ for some $\chi \in \mathcal{L}$ (a *claim*),
  - $rl(\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_m)$ for some $\sigma_0, \ldots, \sigma_m \in \mathcal{L}$ with $m \geq 0$ (a *rule*),
  - $asm(\alpha)$ for some $\alpha \in \mathcal{L}$ (an *assumption*),
  - $ctr(\alpha, \sigma)$ for some $\alpha, \sigma \in \mathcal{L}$ (a *contrary*),
  - a *pass* sentence π, such that $\pi \notin \mathcal{L}$.
- $\text{ID} \in \mathcal{ID} \setminus \{\text{ID}_0\}$ (the *identifier*).
- $T \in \mathcal{ID}$ (the *target*); we impose that $T < \text{ID}$.

Through a dialogue δ, the participating agents construct a joint *ABA framework $\mathcal{F}_\delta$ drawn from δ*. This $\mathcal{F}_\delta$ contains all information that the two agents have uttered in the dialogue and gives the context for examining the "acceptability" of the claim of the dialogue. Conceptually, a dialogue is "successful" if its claim is "acceptable" in $\mathcal{F}_\delta$. Note that the claim of a dialogue may be a belief, and acceptability thereof an indication that the agents may legitimately uphold the belief, or a course of actions, and acceptability thereof an indication that the agents may legitimately choose to adhere to it. Indeed, "acceptability" has so far shown to be an important criterion for assessing the outcome of various types of dialogues [Fan and Toni, 2011c; Fan and Toni, 2012a; Fan and Toni, 2012c; Fan *et al.*, 2014], and thus "successful" dialogues can be seen as building blocks of a widely deployable framework for distributed interactions in multi-agent systems.

Rather than checking "success" retrospectively, this can be guaranteed constructively by means of *legal-move functions* (see [Fan and Toni, 2011a; Fan and Toni, 2014b] for details) guaranteed to generate "successful" dialogues if a limited form of retrospective checking by means of *outcome functions* succeeds [Fan and Toni, 2011a; Fan and Toni, 2014b]. Dialogue goals, e.g. information-seeking, inquiry or persuasion, can be modelled with *strategy-move functions* [Fan and Toni, 2012a]. Given a dialogue, a legal-move function returns a set of allowed utterances that can be uttered to extend the dialogue. Legal-move functions can thus be viewed as dialogue protocols. Outcome functions are mappings from dialogues to true / false. Given a dialogue, an outcome function returns true if a certain property holds for that dialogue. From utterances allowed by legal-move functions, strategy-move functions further select the ones advancing dialogues towards their goals.

We illustrate ABA dialogues for information seeking, persuasion and inquiry for the flat ABA framework in Section 3.2 again, and refer to the original papers for formal definitions and results.

Informally, information seeking dialogues are dialogues with the inquirer agent seeking some specific information from the inquiree agent. In an information seeking dialogue, the inquirer agent does nothing but posing its query, whereas the inquiree agent puts forward information it possesses in answering the query. With the breakdown assistance policy example, suppose that the inquirer agent a_1 asks the inquiree agent a_2 about the existence of argument for the sentence $cov'(m, c)$, as follows:

$$\langle a_1, a_2, 0, claim(cov'(m, c)), 1\rangle$$
$$\langle a_2, a_1, 1, rl(cov'(m, c) \leftarrow in(m, c)), 2\rangle$$
$$\langle a_2, a_1, 2, rl(in(m, c) \leftarrow), 3\rangle$$

We can see that with a_1 and a_2 each using suitable strategy-move functions [Fan and Toni, 2012a], a_1 puts forward $cov'(m, c)$ as the claim of this dialogue and a_2 puts forward utterances 2 and 3 establishing the argument (in the ABA framework $\mathcal{F}_\delta$ drawn from the dialogue) for $cov'(m, c)$ supported by the empty set of assumptions and the two rules:

$$cov'(m, c) \leftarrow in(m, c) \text{ and } in(m, c) \leftarrow.$$

Persuasion dialogues are dialogues between two agents posing "incompatible" views towards some topic with the persuader trying to "prove" the topic and the persuadee trying to "disprove" it. Illustrating with the running example, we may have (for a_1 the persuader and a_2 the persuadee):

$$\langle a_1, a_2, 0, claim(not\ cov'(m, c)), 1\rangle$$
$$\langle a_1, a_2, 1, asm(not\ cov'(m, c)), 2\rangle$$
$$\langle a_2, a_1, 2, ctr(not\ cov'(m, c), cov'(m, c)), 3\rangle$$
$$\langle a_2, a_1, 3, rl(cov'(m, c) \leftarrow in(m, c)), 4\rangle$$
$$\langle a_2, a_1, 4, rl(in(m, c) \leftarrow), 5\rangle$$

Here, a_1 tries to establish the acceptability of $not\ cov'(m, c)$ by claiming it as an assumption, thus forming the argument $\{not\ cov'(m, c)\} \vdash not\ cov'(m, c)$, whereas a_2 puts forward the attacking argument $\{\} \vdash cov'(m, c)$ with utterances 3, 4 and 5. The presented persuasion behaviours of both agents are formally defined with strategy-move functions in [Fan and Toni, 2012c].

Inquiry dialogues are about two agents jointly "proving" or "disproving" the acceptability of some claim. Both agents put forward information supporting or attacking the claim. Again illustrated with the breakdown assistance policy example, we may have:

$\langle a_1, a_2, 0, claim(cov(m,c)), 1\rangle$
$\langle a_1, a_2, 1, rl(cov(m,c) \leftarrow ah(m), tr(m,c), pr(c), not\ \neg cov(m,c)), 2\rangle$
$\langle a_1, a_2, 2, rl(ah(m) \leftarrow), 3\rangle$
$\langle a_1, a_2, 2, rl(tr(m,c) \leftarrow), 4\rangle$
$\langle a_1, a_2, 2, rl(pr(c) \leftarrow), 5\rangle$
$\langle a_1, a_2, 2, asm(not\ \neg cov(m,c)), 6\rangle$
$\langle a_2, a_1, 6, ctr(not\ \neg cov(m,c), \neg cov(m,c)), 7\rangle$
$\langle a_2, a_1, 7, rl(\neg cov(m,c) \leftarrow \neg reg(c,m), not\ cov'(m,c)), 8\rangle$
$\langle a_2, a_1, 8, rl(\neg reg(c,m) \leftarrow), 9\rangle$
$\langle a_2, a_1, 8, asm(not\ cov'(m,c)), 10\rangle$
$\langle a_2, a_1, 10, ctr(not\ cov'(m,c), cov'(m,c)), 11\rangle$
$\langle a_2, a_1, 11, rl(cov'(m,c) \leftarrow in(m,c)), 12\rangle$
$\langle a_2, a_1, 12, rl(in(m,c) \leftarrow), 13\rangle$

With utterances 1-6, the argument $\{not\ \neg cov(m,c)\} \vdash cov(m,c)$ is formed. Utterances 7-10 form an attacking argument $\{not\ cov'(m,c)\} \vdash \neg cov(m,c)$, which is attacked by $\{\} \vdash cov'(m,c)$. The inquiry behaviour of agents is formally defined in [Fan and Toni, 2012a].

6 ABA and explanation

It is widely acknowledged that there is a strong interplay between argumentation and explanation, as for example discussed in [Seselja and Straßer, 2013]. In this section we overview existing proposals [Fan and Toni, 2015c; Schulz and Toni, 2016] using dispute trees in ABA (see Section 5) to provide (argumentative) explanations for why sentences should be concluded. Dispute trees for (flat) ABA can also serve as the basis for explanations in other settings, including various forms of decision-making [Fan and Toni, 2014a; Fan *et al.*, 2014; Zhong *et al.*, 2014; Fan *et al.*, 2013] and case-based reasoning [Čyras *et al.*, 2016] (see the original papers for details). In particular, natural language explanations can be drawn automatically from the dispute trees (see [Zhong *et al.*, 2014; Mocanu *et al.*, 2016] for details).

6.1 Dispute trees as explanations in flat ABA

We have seen (in Section 5) that dispute trees can be used to determine whether a sentence is admissible / grounded / ideal (and, as a consequence, preferred / complete). These dispute trees can also provide a computational counterpart for providing explanations for these sentences (being consequences of admissible / grounded / ideal / preferred / complete extensions, respectively). For example, the dispute tree in Figure 3 can be seen as providing an explanation for $cov(m,c)$, in the spirit of [Newton-Smith, 1981]:

> *... if I am asked to explain why I hold some general belief that p, I answer by giving my justification for the claim that p is true.*

Hence, if a belief q does not contribute to the justification of p, q should not be in the explanation of p. Dispute trees are explanations for (the argument in their root supporting) a sentence in that everything in them contribute to justifying the sentence. This informal notion can be formalised in terms of a notion of *related admissibility* of ABA arguments [Fan and Toni, 2015c],[9] in turn defined using a notion of *r-defence* [Fan and Toni, 2015c], given as follows:

Definition 6.1 *Given an ABA framework* $\mathcal{ABA} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$*, let* $\mathcal{AA}(\mathcal{ABA}) = (Args, attack)$ *be the corresponding AA framework of* $\mathcal{ABA}$*.*

- *Given* $\mathtt{a}, \mathtt{b} \in Args$*,* $\mathtt{a}$ r-defends $\mathtt{b}$ *iff:*
  - *(i)* $\mathtt{a} = \mathtt{b}$*, or*
  - *(ii) there exists* $\mathtt{c} \in Args$ *such that* $\mathtt{a}$ *attacks* $\mathtt{c}$ *and* $\mathtt{c}$ *attacks* $\mathtt{b}$*, or*
  - *(iii) there exists* $\mathtt{c} \in Args$ *such that* $\mathtt{a}$ *r-defends* $\mathtt{c}$ *and* $\mathtt{c}$ *r-defends* $\mathtt{b}$.
- *Given* $\mathtt{a} \in Args$ *and* $\sigma \in \mathcal{L}$*,* $\mathtt{a}$ r-defends σ *iff there exists* $\mathtt{b} \in Args$ *such that* $\mathtt{b}$ *supports* σ *and* $\mathtt{a}$ *r-defends* $\mathtt{b}$*.*

As an illustration, given the ABA framework in Section 3.2:

$\{\} \vdash_{arg} cov'(m,c)$ r-defends $\{\} \vdash_{arg} cov'(m,c)$,
$\{\} \vdash_{arg} cov'(m,c)$ r-defends $\{not\ \neg cov(m,c)\} \vdash_{arg} cov(m,c)$,
$\{\} \vdash_{arg} cov'(m,c)$ r-defends $cov'(m,c)$,
$\{not\ \neg cov(m,c)\} \vdash_{arg} cov(m,c)$ r-defends $cov(m,c)$,
$\{\} \vdash_{arg} cov'(m,c)$ r-defends $cov(m,c)$.

The notion of related admissibility is obtained by combining the r-defence relation and standard admissibility as follows:

Definition 6.2 *Given an ABA framework* $\mathcal{ABA}$*, let* $\mathcal{AA}(\mathcal{ABA}) = (Args, attack)$ *be the corresponding AA framework of* $\mathcal{ABA}$*. A set of arguments* $\mathtt{A} \subseteq Args$ *is* related admissible *iff:*

(i) $\mathtt{A}$ *is admissible, and*

(ii) there exists a topic *sentence* σ *(of* $\mathtt{A}$*) such that* σ *is supported by some argument in* $\mathtt{A}$ *and for all* $\mathtt{b} \in \mathtt{A}$*,* $\mathtt{b}$ *defends* σ*.*

Intuitively, for a related admissible set of arguments $\mathtt{A}$ with topic sentence σ, no argument in $\mathtt{A}$ is "unrelated" to σ as all arguments in $\mathtt{A}$ r-defend σ.

As an illustration, given the ABA framework in Section 3.2,

$$\{\{\} \vdash_{arg} cov'(m,c)\}$$

[9]The notions defined in this section can be defined trivially for any AA framework too, as in [Fan and Toni, 2015c]. The notions for AA frameworks corresponding to ABA frameworks, given below, are an instantiation of the notions for any AA frameworks.

is related admissible, with topic sentence $cov'(m, c)$, and

$$\{\{\} \vdash_{arg} cov'(m, c), \{not\ \neg cov(m, c)\} \vdash_{arg} cov(m, c)\}$$

is related admissible, with topic sentence $cov(m, c)$. Instead,

$$\{\{\} \vdash_{arg} cov'(m, c), \{not\ cov'(m, c)\} \vdash_{arg} \neg cov(m, c)\}$$

is not related admissible as it is not admissible; and

$$\{\{\} \vdash_{arg} ah(m), \{\} \vdash_{arg} pr(c)\}$$

is not related admissible as there does not exists a topic sentence σ such that it is defended by both $\{\} \vdash_{arg} ah(m)$ and $\{\} \vdash_{arg} pr(c)$.

Dispute trees correspond to explanations in that their defence sets are related admissible:

Theorem 6.3 *Given an ABA framework $\mathcal{ABA} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, let $\mathcal{AA}(\mathcal{ABA}) = (Args, attack)$ be the corresponding AA framework of $\mathcal{ABA}$. Let $\sigma \in \mathcal{L}$.*

(i) Let $\mathtt{a} = A \vdash_{arg} \sigma \in Args$ and $\mathcal{T}$ be a dispute tree for $\mathtt{a}$. If $\mathcal{T}$ is admissible / grounded / ideal, then $\mathcal{D}(\mathcal{T})$ is related admissible.

(ii) If $\mathtt{A} \subseteq Args$ is related admissible, with topic sentence σ, then there is an admissible dispute tree $\mathcal{T}$ such that $\mathtt{A}' = \mathcal{D}(\mathcal{T})$ and $\mathtt{A}' \subseteq \mathtt{A}$.

Proof.

(i) By definition 6.1, all arguments labelling $\mathtt{P}$ nodes ($\mathcal{D}(\mathcal{T})$) in a dispute tree r-defend the argument labelling the root note. By Theorem 5.5, all arguments labelling $\mathtt{P}$ nodes in an admissible / grounded / ideal dispute tree are admissible. Thus, by Definition 6.2, $\mathcal{D}(\mathcal{T})$ is related admissible.

(ii) If $\mathtt{A}$ is related admissible, by Definition 6.2, $\mathtt{A}$ is also admissible. By Theorem 5.5, there exists an admissible dispute tree $\mathcal{T}$ such that $\mathtt{A}' = \mathcal{D}(\mathcal{T})$ and $\mathtt{A}' \subseteq \mathtt{A}$.

■

6.2 Explanations for answer set programming

We have seen in Section 3.2 that a logic program can be encoded as an (equivalent) ABA framework such that the semantics of the ABA framework coincide with the semantics of the underlying logic program [Bondarenko *et al.*, 1997], for a wide range of semantics including the stable model (or *answer set*) semantics [Schulz and Toni, 2016; Schulz and Toni, 2015; Caminada and Schulz, 2015]. Logic programs under the answer set semantics (or *answer set programming*) can be applied in a wide range of scenarios [Baral and Uyan, 2001;

Lifschitz, 2002; Eiter *et al.*, 2008; Delgrande *et al.*, 2009; Ricca *et al.*, 2010; Gebser *et al.*, 2011b; Boenn *et al.*, 2011; Erdem, 2011; Ricca *et al.*, 2012; Terracina *et al.*, 2013], thanks also to the availability of efficient solvers for the computation of answer sets [Leone *et al.*, 2006; Gebser *et al.*, 2011a; Alviano *et al.*, 2015; Calimeri *et al.*, 2016]. These however do not provide any explanation of the answer sets computed. In particular, given one such answer set, there is no indication as to why a literal is or is not part of an answer set: this would instead be beneficial in human-computer interaction scenarios where logic programming is used for example to support human decision making.

As seen in Section 6.1, dispute trees do not only provide a way of determining whether or not a sentence is, for instance, admissible, but also an explanation as to *why* this is so.

Given that answer sets of a logic program correspond to stable extensions of the ABA framework encoding this logic program [Bondarenko *et al.*, 1997] and that if an answer set is guaranteed to exist then it is preferred (See Theorem 2.11 (i)), dispute trees can be used to determine, for a computed answer set and sentence in it, an explanation (in the form of a dispute tree) for why this is so. However, for the purpose of extracting explanations for literals in terms of other literals (rather than arguments, see [Schulz and Toni, 2016]), it is useful to single out, from the set of rules supporting ABA arguments, the rules with an empty body (referred to as *facts* in LP):

Definition 6.4 *Given a flat ABA framework* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$*, we say that* $(A, F) \vdash_{arg} \sigma$ *is a* fact-based-argument *for* $\sigma \in \mathcal{L}$ *supported by* $A \subseteq \mathcal{A}$ *and* $F \subseteq \{\pi \leftarrow \mid \pi \leftarrow \in \mathcal{R}\}$*, if there is an argument* $A \vdash^{R}_{arg} \sigma$ *such that* $F = R \cap \{\pi \leftarrow \mid \pi \leftarrow \in \mathcal{R}\}$*.*

A generalisation of dispute trees, which we call *explanation trees* [Schulz and Toni, 2016], where nodes are labelled by fact-based-arguments[10] can be used to explain why a literal is contained in a given answer set.

As an example, consider the ABA framework in Section 3.2, and the logic program amounting to its rules. This logic program has only one answer set: $\{ah(m), tr(m,c), pr(c), \neg reg(c,m), in(m,c), cov(m,c), cov'(m,c)\}$.

The explanation tree in Figure 4 justifies why Mary is covered, i.e. why $cov(m,c)$ is contained in the answer set. It expresses that there is evidence that Mary is covered (given by the argument with conclusion $cov(m,c)$ in the root proponent node) since Mary is the account holder and she is travelling in a car which is a private vehicle (facts supporting the argument), and since it can be assumed that there is no evidence that Mary is not covered ($not\, \neg cov(m,c)$ is an assumption). Even though there is evidence against this assumption, i.e. there is evidence that that Mary is not covered (given by the argument with conclusion $\neg cov(m,c)$ in the opponent node) because she is not registered on

[10]For better readability we will omit the symbol $\leftarrow$ for all facts in the set F.

$$\texttt{P} : (\{not\, \neg cov(m,c)\}, \{ah(m), tr(m,c), pr(c)\}) \vdash_{arg} cov(m,c)$$

$$|$$

$$\texttt{O} : (\{not\, cov'(m,c)\}, \{\neg reg(c,m)\}) \vdash_{arg} \neg cov(m,c)$$

$$|$$

$$\texttt{P} : (\{\}, \{in(m,c)\}) \vdash_{arg} cov'(m,c)$$

Figure 4. An explanation tree justifying why Mary is covered in the running example

the car, this evidence can be disregarded since Mary was in the car at the time of the breakdown (given by the proponent argument with conclusion $cov'(m,c)$, which attacks the assumptions $not\, cov'(m,c)$ of the opponent node). Note that this explanation tree is the same as the dispute tree in Figure 3 except that it uses fact-based-arguments.

In contrast to dispute trees which are used to justify only the containment of an argument *in* an extension, explanation trees can also explain why a literal is *not in* an answer set. In that case, explanation trees have an opponent node as their root, as illustrated by the explanation tree below which justifies why it is not the case that Mary is not covered (why $\neg cov(m,c)$ is not part of the answer set)

$$\texttt{O} : (\{not\, cov'(m,c)\}, \{\neg reg(c,m)\}) \vdash_{arg} \neg cov(m,c)$$

$$|$$

$$\texttt{P} : (\{\}, \{in(m,c)\}) \vdash_{arg} cov'(m,c)$$

Note that this explanation tree is a sub-tree of the previous explanation tree in Figure 4 justifying why $cov(m,c)$ is contained in the answer set.

Since explanation trees whose root node is a proponent node are dispute trees and since arguments which are in a stable extension are also in an admissible extension (Theorem 2.11 (i)), it follows from the relationhip between admissible extensions and admissible dispute trees given in Theorem 5.5 (i) that explanation trees starting with proponent nodes are admissible dispute trees. Thus, for literals contained in the answer set, explanation trees illustrate that the literal is supported by an admissible subset of this answer set.

Explanation trees whose root node is an opponent node have an explanation tree for a literal contained in the answer set as its direct sub-tree. Thus, this direct sub-tree is an admissible dispute tree. This means that literals not contained in the answer set are justified by illustrating that they are attacked by an admissible subset of the answer set.

In summary, explanation trees provide justifications of literals with respect to an answer set in terms of *admissible subsets* of this answer set [Schulz and Toni, 2016].

7 ABA and reasoning with preferences

Argumentation and preferences come a long way, see e.g. [Simari and Loui, 1992]. In general, preferences can be used to express, for instance, agents' degrees of belief, imperatives (moral, legal, etc.), aims, wishes. There are numerous methods in knowledge representation and reasoning to account for preference information, see e.g. [Prakken and Sartor, 1999; Kakas and Moraitis, 2003; Delgrande *et al.*, 2004; Brewka *et al.*, 2010; Domshlak *et al.*, 2011], and, in particular, several argumentation formalisms handling preferences, see e.g. [Bench-Capon, 2003; Modgil, 2009; Modgil and Prakken, 2014; García and Simari, 2014; Besnard and Hunter, 2014; Amgoud and Vesic, 2014; Baroni *et al.*, 2011], where preferences help to discriminate amongst information such as extensions, arguments, assumptions, rules, decisions and goals [Wakaki, 2014; Besnard and Hunter, 2014; Čyras and Toni, 2016a; Modgil and Prakken, 2014; Fan *et al.*, 2013]. There are various ways to deal with preferences in ABA too [Kowalski and Toni, 1996; Toni, 2008b; Thang and Luong, 2013; Fan *et al.*, 2013; Wakaki, 2014; Čyras and Toni, 2016a; Čyras and Toni, 2016b]. In this section we illustrate (by way of examples) these latter approaches. At a high-level, they can be divided in two groups: *meta level* approaches ([Wakaki, 2014; Čyras and Toni, 2016a; Čyras and Toni, 2016b], see Section 7.1), which, roughly, account for preferences at the semantic level, and *object level* approaches ([Kowalski and Toni, 1996; Toni, 2008b; Thang and Luong, 2013; Fan *et al.*, 2013], see Section 7.2), which, roughly, encode preferences within the existing ABA components (e.g. rules and assumptions) and avoid the need to modify the semantics of ABA frameworks.

Note that the examples chosen for the illustrations in this section have been selected for their simplicity, to give a high-level idea of the various approaches overviewed, and may not convey the full sophistication and usefulness of these approaches: the interested reader can find details as well as formal results in the original papers.

7.1 Handling preferences in ABA at the meta-level

[Wakaki, 2014] follows the ideas of prioritized logic programming [Sakama and Inoue, 2000] and equips ABA with explicit preferences by introducing a binary preference relation $\leqslant$ over the language $\mathcal{L}$. (For $a, b \in \mathcal{L}$, $a \leqslant b$ expresses that 'a is less or equally preferred than b'.) This ordering $\leqslant$ is then used to compute, by comparing consequences of extensions, a preference ordering $\sqsubseteq$ over extensions so as to select the most "preferable" extensions (i.e. the $\sqsubseteq$-maximal ones) of the underlying ABA framework. Such meta-level preference treatment can be well illustrated via scenarios of decision making with preferences, as in the following example.

Example 7.1 *Mary needs to decide what insurance policy to buy. Following the approach of [Fan* et al.*, 2013], information relevant to the decision making is represented via two tables,* $\texttt{T}_{\texttt{DA}}$ *and* $\texttt{T}_{\texttt{GA}}$*, as illustrated in Table 1, where*

- $\texttt{T}_{\texttt{DA}}$ *describes relations between* decision candidates *(Policy 1 (*P_1*), Policy 2 (*P_2*)) and* attributes *(£50, £70, no exceptions (*no_ex*));*
- $\texttt{T}_{\texttt{GA}}$ *describes relations between* goals *(*cheap *and full coverage (*full*)) and* attributes*.*

	£50	£70	no_ex
P_1	0	1	1
P_2	1	0	0

	£50	£70	no_ex
cheap	1	0	0
full	0	0	1

Table 1. $\texttt{T}_{\texttt{DA}}$(left) and $\texttt{T}_{\texttt{GA}}$(right), for Example 7.1.

Intuitively, each decision candidate has *certain attributes (*P_1 *has £70 and no_ex;* P_2 *has £50); and each goal can be* met *by certain attributes (*cheap *is met by £50;* full *is met by no_ex).*

In addition, suppose that the goal full *is preferred over* cheap*. In p_ABA, we can represent this information as a framework* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}, \leqslant \rangle$*, with the underlying ABA framework* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ *with*

$$\begin{aligned}
\mathcal{R} = \{ & \pounds 70 \leftarrow P_1, \quad no_ex \leftarrow P_1, \quad \pounds 50 \leftarrow P_2, \quad cheap \leftarrow \pounds 50, \\
& full \leftarrow no_ex\ , \quad \overline{P_2} \leftarrow P_1, \quad \overline{P_1} \leftarrow P_2 \}, \\
\mathcal{A} = \{ & P_1, P_2 \}, \ \ and \\
& cheap \leqslant full\ , \ \ cheap \leqslant cheap\ , \ \ full \leqslant full.
\end{aligned}$$

$\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ *has two preferred / stable extensions* $\{P_1\}$ *and* $\{P_2\}$*, with conclusions* $Cn(\{P_1\}) = \{P_1, \pounds 70,\ no\_ex,\ full\}$ *and* $Cn(\{P_2\}) = \{P_2, \pounds 50,\ cheap\}$*. We then find* $\{P_2\} \sqsubseteq \{P_1\}$ *and* $\{P_1\} \not\sqsubseteq \{P_2\}$*, so that* $\{P_1\}$ *is a* $\sqsubseteq$*-maximal extension, and is hence selected as the "preferable" one. Buying Policy 1 is thus deemed the better decision to take.*

Preferences in ABA can also be utilized to modify the attack relation between sets of assumptions, akin to approaches to argumentation with preferences such as [Bench-Capon, 2003; Modgil and Prakken, 2014; Amgoud and Vesic, 2014; Besnard and Hunter, 2014]. For instance, ABA$^+$ [Čyras and Toni, 2016a; Čyras and Toni, 2016b] equips ABA with a binary preference relation $\leqslant$ over assumptions, and incorporates preferences directly into the attack relation so as to *reverse* attacks that stem from sets containing assumptions less preferred than the one whose contrary is deduced, as illustrated next.

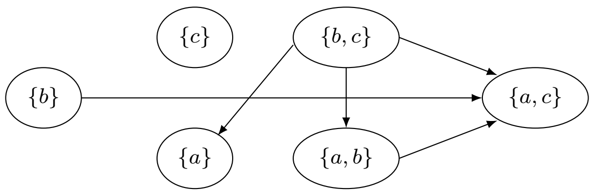
Example 7.2 *Suppose that Mary has decided to buy Policy 1, as suggested in Example 7.1. However, Mary has also found some information on the Internet* *about the policy: source C says that under certain circumstances (c), the policy applies only to citizens of certain specified countries; source B says that sometimes (say, assuming c), the policy applies only to UK residents ($UK \leftarrow b, c$); source A says that sometimes (assuming c) the policy applies only to non-UK residents ($non\_UK \leftarrow a, c$). Mary trusts the source A the least (i.e. $a < b$, $a < c$). What is Mary justified believing in about the applicability of the policy, given certain circumstances?*

We can formalize this in ABA$^+$ as follows: consider $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}, \leqslant \rangle$ with

$$\mathcal{A} = \{a, b, c\},$$
$$\mathcal{R} = \{non\_UK \leftarrow a, c, \quad UK \leftarrow b, c\},$$
$$\overline{a} = UK, \quad \overline{b} = non\_UK,$$
$$a < b, \quad a < c,$$

where the assumptions stand for the possibility to trust the sources and preferences indicate their relative credibility, rules are drawn given that information from sources A and B is applicable under certain circumstances (c), also given that sources A and B are in conflict.

The underlying ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ admits both $\{a, c\}$ and $\{b, c\}$ as stable / preferred extensions. In ABA$^+$, attacks from $\{a, c\}$ to (any set of assumptions containing) b are reversed, due to the a's lower credibility in comparison with b. Hence, $\{b, c\}$ is a unique stable / preferred extension, arguably the desirable outcome. This can be seen clearly given the graph depicted below, omitting, for readability, assumption sets $\{\}$ and $\{a, b, c\}$, as well as attacks to and from them:



7.2 Handling preferences in ABA at the object-level

Instead of equipping ABA frameworks with explicit preference relations as in Section 7.1, and then modifying the semantics of ABA (by either comparing extensions or modifying the attack relation), preferences can be encoded within the existing components (rules, assumptions and contraries) without modifying the semantics.

For instance, [Kowalski and Toni, 1996; Toni, 2008b] deal with preferences between rules by adding conditions (i.e. assumptions) to the body of rules expressing that the rules are not attacked by other higher preference rules, by appropriately defining contraries of these assumptions. For illustration, consider the following example:

Example 7.3 *In our breakdown policy example of Section 3, the rules in the ABA instance for LP of section 3.2 can be modified by adding assumptions as follows:*

$$cov(m,c) \leftarrow ah(m), tr(m,c), pr(c), not\ \neg cov(m,c), a_{cov(m,c)},$$
$$\neg cov(m,c) \leftarrow \neg reg(c,m), not\ cov'(m,c), a_{\neg cov(m,c)}.$$

If a preference of the second rule over the first one is to be expressed, then one could assign contraries

$$\overline{a_{cov(m,c)}} = \neg cov(m,c), \quad \overline{a_{\neg cov(m,c)}} = a^c,$$

where a^c is new to $\mathcal{L}$.

More generally (as in [Toni, 2008b]), one can assume a naming *function assigning distinguished names to elements (e.g. rules) of a given domain, and given preferences over the elements of the domain, consider a language that includes sentences expressing those preferences. For example, the two rules above can be given names r and r' respectively, and the language $\mathcal{L}$ would contain a "preference sentence" $r < r'$ expressing that the second rule is preferred over the first one. Then, when mapping the domain into an ABA framework, a rule*

$$\neg cov(m,c) \leftarrow r < r', a_{\neg cov(m,c)},$$

could be added, so as to account for preferences, which could be stated e.g. via a rule $r < r' \leftarrow$. This way, ABA can also account for dynamic preferences *(see e.g. [Prakken and Sartor, 1999]), i.e. preferences that are themselves deducible using rules, possibly from other assumptions.*

Yet another way to deal with preferences in ABA on the object level is used in [Thang and Luong, 2013] when translating Brewka's preferred subtheories [Brewka, 1989] into ABA. To capture the interplay between classically inconsistent sentences and partial preference information among them, [Thang and Luong, 2013] introduce assumptions for representing sentences in the domain language as well as for determining their acceptance status in the construction of preferred subtheories, and further introduce rules for: deriving sentences from their corresponding assumptions; deriving contraries of the least preferred elements of minimally inconsistent subsets; enforcing (non-)acceptance of an assumption iff the statuses more preferred assumptions are determined. This is illustrated next.

Example 7.4 *Let us rewrite the rules from Example 7.2 as*

$$\alpha, \gamma \rightarrow \neg UK, \quad \beta, \gamma \rightarrow UK$$

(where $\rightarrow$ is material implication) to constitute the facts (world knowledge), and let $T = \{\alpha, \beta, \gamma\}$ be the theory representing the defeasible knowledge, with preferences $\alpha < \beta$ and $\alpha < \gamma$. This partial order $<$ admits two extensions to

total orders, namely $\alpha < \beta < \gamma$ and $\alpha < \gamma < \beta$, both of which result in the same preferred subtheory of T, namely $\{\beta, \gamma\}$.

The domain can be mapped into an ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ with (for readability treating contraries of assumptions as symbols in the language)

$$\begin{aligned}
\mathcal{A} =& \{a_\alpha, a_\beta, a_\gamma\} \cup \{b_\alpha, b_\beta, b_\gamma\},\\
\mathcal{R} =& \{\alpha \leftarrow a_\alpha, \quad \beta \leftarrow a_\beta, \quad \gamma \leftarrow a_\gamma\} \cup \{\overline{a_\alpha} \leftarrow a_\beta, \overline{b_\beta}, a_\gamma, \overline{b_\gamma}\} \cup\\
& \{\overline{a_\alpha} \leftarrow b_\alpha, \quad \overline{a_\beta} \leftarrow b_\beta, \quad \overline{a_\gamma} \leftarrow b_\gamma\} \cup\\
& \{\overline{b_\beta} \leftarrow, \quad \overline{b_\gamma} \leftarrow, \quad \overline{b_\alpha} \leftarrow a_\beta, \overline{b_\beta}, a_\gamma, \overline{b_\gamma},\\
& \overline{b_\alpha} \leftarrow \overline{a_\beta}, \overline{b_\beta}, a_\gamma, \overline{b_\gamma}, \quad \overline{b_\alpha} \leftarrow a_\beta, \overline{b_\beta}, \overline{a_\gamma}, \overline{b_\gamma}, \quad \overline{b_\alpha} \leftarrow \overline{a_\beta}, \overline{b_\beta}, \overline{a_\gamma}, \overline{b_\gamma}\}.
\end{aligned}$$

This $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ has a unique stable extension $\{a_\beta, a_\gamma\}$, corresponding to the unique preferred subtheory of T.

Another example of preferences dealt with in ABA within the object-level is to support decision making with preferences over goals. Differently from the other approaches overviewed in this section, this method is specific to decision making settings, and uses preferences over sentences (the goals) within decision criteria (e.g. various kinds of "dominance", see [Fan *et al.*, 2013]) for choosing "best" decisions. This can be illustrated in the context of the same decision making setting of Example 7.1:

Example 7.5 *Given the two tables, $\texttt{T}_{\texttt{DA}}$ and $\texttt{T}_{\texttt{GA}}$, in Table 1, as well as the preference full $>$ cheap, the problem of identifying the "best" decisions, namely those "meeting the more preferred goals that no other decisions meet", can be represented in ABA with*

$$\begin{aligned}
\mathcal{R} = \quad & \{ \quad has(P_1, \pounds 70) \leftarrow \ , \quad has(P_1,\ no_ex) \leftarrow \ , \quad has(P_2, 50) \leftarrow \ ,\\
& \quad satBy(cheap, \pounds 50) \leftarrow \ , \quad satBy(full) \leftarrow no_ex,\\
& \quad prefer(full, cheap) \leftarrow \ \} \cup\\
& \{ \quad met(X,Y) \leftarrow has(X,Z), satBy(Y,Z) \mid X \in \{P_1, P_2\},\\
& \quad Y \in \{cheap, full\},\ Z \in \{\pounds 50, \pounds 70,\ no_ex\ \}\ \} \cup\\
& \{ \quad sel(X) \leftarrow met(X,Y), noBetterThan(X,Y) \mid X \in \{P_1, P_2\},\\
& \quad Y \in \{cheap, full\}\ \} \cup\\
& \{ \quad better(X,Y) \leftarrow met(X',Y'), prefer(Y',Y), X \neq X' \mid\\
& \quad X, X' \in \{P_1, P_2\},\ Y, Y' \in \{cheap, full\}\ \}\\
\mathcal{A} = \quad & \{ \quad noBetterThan(X,Y) \mid X \in \{P_1, P_2\},\ Y \in \{cheap, full\}\ \}\\
\overline{not\ x} = \quad & better(X,Y) \quad \text{for any} \quad x = noBetterThan(X,Y) \in \mathcal{A}
\end{aligned}$$

Then

$$\{\{noBetterThan(P_1, full)\} \vdash_{arg} sel(P_1)\}$$

is admissible whereas

$$\{\{noBetterThan(P_2, cheap)\} \vdash_{arg} sel(P_2)\}$$

is not, as the latter is attacked by $\{\} \vdash_{arg} better(P_2, cheap)$*. Indeed, Policy 1 is the "best" decision in this simple setting.*

8 Conclusion

This chapter overviews research, spanning over more than two decades (from [Bondarenko *et al.*, 1993] onwards), on Assumption-Based Argumentation (ABA), a framework for structured argumentation motivated by and emerging from non-monotonic reasoning. We have focused on the semantic foundations of ABA, in the general case as well as for the special case of flat ABA frameworks, while also providing an overview of the computational machinery (flat) ABA is equipped with as well as its uses for explaining argumentative conclusions. Finally, we have overviewed, with the aid of examples, uses and generalisations of ABA to support reasoning with preferences.

This chapter is meant as a taster of ABA rather than a comprehensive technical presentation, and complements other earlier overviews [Dung *et al.*, 2009; Toni, 2012; Toni, 2014]. In particular, it focuses on the case of general (possibly non-flat) frameworks rather than flat frameworks as in the earlier overviews, and provides a taster of explanation and the treatment of preferences in ABA.

We omitted to mention several aspects of ABA. For instance, there are several other instances of ABA for non-monotonic reasoning (see [Bondarenko *et al.*, 1997]), and ABA has also been shown to admit Adaptive Logic and ASPIC+ without preferences as instances [Heyninck and Straßer, 2016]. Other ABA semantics have been presented in the literature, e.g. the semi-stable semantics [Caminada *et al.*, 2015]. Moreover, formulation of (some) ABA semantics in terms of labellings, in the spirit of those proposed for abstract argumentation [Caminada and Gabbay, 2009], have been proposed [Schulz and Toni, 2014; Schulz and Toni, 2015; Schulz and Toni, 2017]. Further, the computational complexity of several reasoning problems in several instances of ABA is known [Dimopoulos *et al.*, 2002; Dunne, 2009], and several systems for (flat) ABA are publicly available (see `robertcraven.org/proarg/` and `www-abaplus.doc.ic.ac.uk`). Recent work also shows that (sets of) arguments in ABA can be re-interpreted as graphs, with conceptual and computational advantages [Craven and Toni, 2016]. We have seen in Section 7 that ABA has been extended to accommodate reasoning with preferences: other extensions of ABA also exist, notably the probabilistic ABA of [Dung and Thang, 2010]. Finally, we have not delved into applications of ABA: these are overviewed in earlier surveys [Dung *et al.*, 2009; Toni, 2012; Toni, 2014] or other papers [Gao *et al.*, 2016; Fan and Toni, 2016]. In particular, [Gao *et al.*, 2016] uses related admissibility in ABA (see Section 6.1) to coordinate and resolve conflicts amongst agents, while also guaranteeing that privacy is preserved, in some sense, whereas [Fan and Toni, 2016] reinterprets the problem of determining solutions in games in normal form in ABA, using ABA dialogues (as summarised in Section 5.3) to determine these solutions in a distributed fashion, without agents fully disclosing their preferences.

There are several open issues in ABA as well as several directions for future work. We have seen, in Section 6.2, that explanations as to why sentences are not "acceptable" may be useful [Schulz and Toni, 2016]. The concept of "not-explanations" can be defined, in general, in abstract argumentation [Fan and Toni, 2015b]: it would be useful to define this notion also for ABA. Other forms of explanations have been defined, notably for explaining inconsistencies in LP [Schulz *et al.*, 2015]: it would be interesting to define a notion of explanation for the lack of (e.g. stable) extensions in generic ABA. Some preliminary work [Zhong *et al.*, 2014; Mocanu *et al.*, 2016] indicates that natural language explanations can be naturally drawn from dispute trees computed by dispute derivations: it would be interesting to develop this work further and test the usefulness of the generated explanations in practice. Further, in multi-agent settings, it would be interesting to further study strategic behaviour of agents using ABA as their language of interaction [Fan and Toni, 2012c; Fan and Toni, 2015a; Gao *et al.*, 2016; Fan and Toni, 2016]. From a computational viewpoint, (flat) ABA is equipped with several (sound and complete) algorithms for determining the "acceptability" of sentences (and compute extensions "supporting" them): it would be interesting to see how these algorithms can be generalised to the case of any, possibly non-flat, ABA frameworks and/or deployed when preferences are given, e.g. in the spirit of Gorgias (see `gorgiasb.tuc.gr/index.html`) and dealt with at the meta-level (as in Section 7.1). Moreover, it would be interesting to identify (sound and complete) computational machinery for determining extensions of ABA, without having to resort to implementations of abstract argumentation by using the mapping described in Section 4.

BIBLIOGRAPHY

[Alferes *et al.*, 1993] José Júlio Alferes, Phan Minh Dung, and Luís Moniz Pereira. Scenario semantics of extended logic programs. In A. Nerode and L. Pereira, editors, *Proceedings of the 2nd International Workshop on Logic Programming and Nonmonotonic Reasoning (LPNMR'93)*, pages 334–348. MIT Press, 1993.

[Alviano *et al.*, 2015] Mario Alviano, Carmine Dodaro, Nicola Leone, and Francesco Ricca. Advances in WASP. In *Proceedings of the 13th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'15)*, pages 40–54, 2015.

[Amgoud and Vesic, 2014] Leila Amgoud and Srdjan Vesic. Rich preference-based argumentation frameworks. *International Journal of Approximate Reasoning*, 55(2):585–606, 2014.

[Baral and Uyan, 2001] Chitta Baral and Cenk Uyan. Declarative specification and solution of combinatorial auctions using logic programming. In *Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'01)*, pages 186–199, 2001.

[Baroni *et al.*, 2011] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. AFRA: argumentation framework with recursive attacks. *International Journal of Approximate Reasoning*, 52(1):19–37, 2011.

[Bench-Capon, 2003] Trevor J. M. Bench-Capon. Persuasion in practical argument using value based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[Besnard and Hunter, 2014] Philippe Besnard and Anthony Hunter. Constructing argument graphs with deductive arguments: A tutorial. *Argument & Computation*, 5(1):5–30, 2014.

[Besnard *et al.*, 2014] Philippe Besnard, Alejandro Javier García, Anthony Hunter, Sanjay Modgil, Henry Prakken, Guillermo Ricardo Simari, and Francesca Toni. Introduction to structured argumentation. *Argument & Computation*, 5(1):1–4, 2014.

[Boenn *et al.*, 2011] Georg Boenn, Martin Brain, Marina De Vos, and John P. Fitch. Automatic music composition using answer set programming. *Theory and Practice of Logic Programming*, 11(2-3):397–427, 2011.

[Bondarenko *et al.*, 1993] Andrei Bondarenko, Francesca Toni, and Robert A. Kowalski. An assumption-based framework for mon-monotonic reasoning. In L. M. Pereira and A. Nerode, editors, *Proceedings of the 2nd International Workshop on Logic Programming and Non-monotonic Reasoning (LPNMR'93)*, pages 171–189, Lisbon, Portugal, June 1993. MIT Press.

[Bondarenko *et al.*, 1997] Andrei Bondarenko, Phan Minh Dung, Robert A. Kowalski, and Francesca Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1-2):63–101, 1997.

[Brewka *et al.*, 1997] Gerhard Brewka, Jürgen Dix, and Kurt Konolige. *Nonmonotonic Reasoning: An Overview*, volume 73 of *CSLI Lecture Notes*. CSLI Publications, Stanford, CA, 1997.

[Brewka *et al.*, 2010] Gerhard Brewka, Mirosław Truszczyński, and Stefan Woltran. Representing preferences among sets. In Maria Fox and David Poole, editors, *24th AAAI Conference on Artificial Intelligence (AAAI'10)*, pages 273–278, Atlanta, Georgia, 2010. AAAI Press.

[Brewka, 1989] Gerhard Brewka. Preferred subtheories: An extended logical framework for default reasoning. In N. S. Sridharan, editor, *11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1043–1048, Detroit, 1989. Morgan Kaufmann.

[Calimeri *et al.*, 2016] Francesco Calimeri, Martin Gebser, Marco Maratea, and Francesco Ricca. Design and results of the fifth answer set programming competition. *Artificial Intelligence*, 231:151–181, 2016.

[Caminada and Gabbay, 2009] Martin Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009.

[Caminada and Schulz, 2015] Martin Caminada and Claudia Schulz. On the equivalence between assumption-based argumentation and logic programming. In *Proceedings of the 1st International Workshop on Argumentation and Logic Programming (ArgLP'15)*, 2015.

[Caminada *et al.*, 2015] Martin Caminada, Samy Sá, João Alcântara, and Wolfgang Dvoák. On the difference between assumption-based argumentation and abstract argumentation. *The IfCoLog Journal of Logics and their Applications*, 2(1):16–34, 2015.

[Craven and Toni, 2016] Robert Craven and Francesca Toni. Argument graphs and assumption-based argumentation. *Artificial Intelligence*, 233:1–59, 2016.

[Čyras and Toni, 2016a] Kristijonas Čyras and Francesca Toni. ABA+: Assumption-based argumentation with preferences. In Chitta Baral, James P. Delgrande, and Frank Wolter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 15th International Conference, (KR'16)*, pages 553–556. AAAI Press, 2016.

[Čyras and Toni, 2016b] Kristijonas Čyras and Francesca Toni. Properties of ABA+ for non-monotonic reasoning. In *Proceedings of the 16th International Workshop on Non-Monotonic Reasoning (NMR'16), CoRR*, volume abs/1603.08714, 2016.

[Čyras *et al.*, 2016] Kristijonas Čyras, Ken Satoh, and Francesca Toni. Abstract argumentation for case-based reasoning. In Chitta Baral, James P. Delgrande, and Frank Wolter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 15th International Conference (KR'16)*, pages 549–552. AAAI Press, 2016.

[Delgrande *et al.*, 2004] James P. Delgrande, Torsten H. Schaub, Hans Tompits, and Kewen Wang. A classification and survey of preference handling approaches in nonmonotonic reasoning. *Computational Intelligence*, 20(2):308–334, 2004.

[Delgrande *et al.*, 2009] James P. Delgrande, Torsten Grote, and Aaron Hunter. A general approach to the verification of cryptographic protocols using answer set programming. In

Proceedings of the 10th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'09), pages 355–367, 2009.

[Dimopoulos *et al.*, 2002] Yannis Dimopoulos, Bernhard Nebel, and Francesca Toni. On the computational complexity of assumption-based argumentation for default reasoning. *Artificial Intelligence*, 141(1/2):57–78, 2002.

[Domshlak *et al.*, 2011] Carmel Domshlak, Eyke Hüllermeier, Souhila Kaci, and Henri Prade. Preferences in AI: an overview. *Artificial Intelligence*, 175(7-8):1037–1052, 2011.

[Dung and Thang, 2010] Phan Minh Dung and Phan Minh Thang. Towards (probabilistic) argumentation for jury-based dispute resolution. In *Proceedings of the 2010 Conference on Computational Models of Argument (COMMA'10)*, pages 171–182. IOS Press, 2010.

[Dung *et al.*, 2002] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Argumentation-based proof procedures for credulous and sceptical non-monotonic reasoning. In A.C. Kakas and F. Sadri, editors, *Computational Logic: Logic Programming and Beyond - Essays in Honour of Robert A. Kowalski*, volume 2408 of *Lecture Notes in Artificial Intelligence*, pages 289–310. Springer Verlag, 2002.

[Dung *et al.*, 2006] Phan Minh Dung, Robert A. Kowalski, and Francesca Toni. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence*, 170:114–159, 2006.

[Dung *et al.*, 2007] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(10–15):642–674, 2007.

[Dung *et al.*, 2009] Phan Minh Dung, Robert A. Kowalski, and Francesca Toni. Assumption-based argumentation. In Iyad Rahwan and Guillermo R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 25–44. Springer, 2009.

[Dung, 1991] Phan Minh Dung. Negations as hypotheses: An abductive foundation for logic programming. In Koichi Furukawa, editor, *Logic Programming, Proceedings of the 8th International Conference (ICLP'91)*, pages 3–17. MIT Press, 1991.

[Dung, 1995] Phan Minh Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

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

[Eiter *et al.*, 2008] Thomas Eiter, Giovambattista Ianni, Thomas Lukasiewicz, Roman Schindlauer, and Hans Tompits. Combining answer set programming with description logics for the semantic web. *Artificial Intelligence*, 172(12-13):1495–1539, 2008.

[Erdem, 2011] Esra Erdem. Applications of answer set programming in phylogenetic systematics. In *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning - Essays Dedicated to Michael Gelfond on the Occasion of His 65th Birthday*, pages 415–431, 2011.

[Fan and Toni, 2011a] Xiuyi Fan and Francesca Toni. Assumption-based argumentation dialogues. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 198–203, 2011.

[Fan and Toni, 2011b] Xiuyi Fan and Francesca Toni. Conflict resolution with argumentation dialogues. In *Proceedings of the 2011 International Conference on Autonomous Agents & Multiagent Systems (AAMAS'11)*, pages 1095–1096, 2011.

[Fan and Toni, 2011c] Xiuyi Fan and Francesca Toni. A first step towards argumentation dialogues for discovery. In *Theory and Applications of Formal Argumentation - 1st International Workshop, (TAFA'11)*, pages 263–279, 2011.

[Fan and Toni, 2012a] Xiuyi Fan and Francesca Toni. Agent strategies for ABA-based information-seeking and inquiry dialogues. In *20th European Conference on Artificial Intelligence (ECAI'12)*, pages 324–329, 2012.

[Fan and Toni, 2012b] Xiuyi Fan and Francesca Toni. Argumentation dialogues for two-agent conflict resolution. In *Proceedings of the 2012 Conference on Computational Models of Argument (COMMA'12)*, 2012.

[Fan and Toni, 2012c] Xiuyi Fan and Francesca Toni. Mechanism design for argumentation-based persuasion. In *Proceedings of the 2012 Conference on Computational Models of Argument (COMMA'12)*, pages 322–333, 2012.

[Fan and Toni, 2014a] Xiuyi Fan and Francesca Toni. Decision making with assumption-based argumentation. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Theory* *and Applications of Formal Argumentation - 2nd International Workshop (TAFA'13)*, volume 8306 of *Lecture Notes in Computer Science*, pages 127–142. Springer, 2014.

[Fan and Toni, 2014b] Xiuyi Fan and Francesca Toni. A general framework for sound assumption-based argumentation dialogues. *Artificial Intelligence*, 216:20–54, 2014.

[Fan and Toni, 2015a] Xiuyi Fan and Francesca Toni. Mechanism design for argumentation-based information-seeking and inquiry. In Qingliang Chen, Paolo Torroni, Serena Villata, Jane Yung-jen Hsu, and Andrea Omicini, editors, *Principles and Practice of Multi-Agent Systems - 18th International Conference (PRIMA'15)*, volume 9387 of *Lecture Notes in Computer Science*, pages 519–527. Springer, 2015.

[Fan and Toni, 2015b] Xiuyi Fan and Francesca Toni. On computing explanations for non-acceptable arguments. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Theory and Applications of Formal Argumentation - 3rd International Workshop (TAFA'15)*, volume 9524 of *Lecture Notes in Computer Science*, pages 112–127. Springer, 2015.

[Fan and Toni, 2015c] Xiuyi Fan and Francesca Toni. On computing explanations in argumentation. In Blai Bonet and Sven Koenig, editors, *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, pages 1496–1502. AAAI Press, 2015.

[Fan and Toni, 2016] Xiuyi Fan and Francesca Toni. On the interplay between games, argumentation and dialogues. In Catholijn M. Jonker, Stacy Marsella, John Thangarajah, and Karl Tuyls, editors, *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS'16)*, pages 260–268. ACM, 2016.

[Fan *et al.*, 2013] Xiuyi Fan, Robert Craven, Ramsay Singer, Francesca Toni, and Matthew Williams. Assumption-based argumentation for decision-making with preferences: A medical case study. In João Leite, Tran Cao Son, Paolo Torroni, Leon van der Torre, and Stefan Woltran, editors, *Computational Logic in Multi-Agent Systems: 14th International Workshop (CLIMA'13)*, volume 8143 of *Lecture Notes in Computer Science*, pages 374–390. Springer, 2013.

[Fan *et al.*, 2014] Xiuyi Fan, Francesca Toni, Andrei Mocanu, and Matthew Williams. Dialogical two-agent decision making with assumption-based argumentation. In *Proceedings of the 2014 International Conference on Autonomous Agents & Multiagent Systems (AAMAS'14)*, pages 533–540, 2014.

[Gaertner and Toni, 2008] Dorian Gaertner and Francesca Toni. Hybrid argumentation and its properties. In *Proceedings of the 2008 Conference on Computational Models of Argument (COMMA'08)*, pages 183–195, 2008.

[Gao *et al.*, 2016] Yang Gao, Francesca Toni, Hao Wang, and Fanjiang Xu. Argumentation-based multi-agent decision making with privacy preserved. In Catholijn M. Jonker, Stacy Marsella, John Thangarajah, and Karl Tuyls, editors, *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS'16)*, pages 1153–1161. ACM, 2016.

[García and Simari, 2014] Alejandro Javier García and Guillermo Ricardo Simari. Defeasible logic programming: DeLP-servers, contextual queries, and explanations for answers. *Argument & Computation*, 5(1):63–88, 2014.

[Gebser *et al.*, 2011a] Martin Gebser, Benjamin Kaufmann, Roland Kaminski, Max Ostrowski, Torsten Schaub, and Marius Thomas Schneider. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107–124, 2011.

[Gebser *et al.*, 2011b] Martin Gebser, Torsten Schaub, Sven Thiele, and Philippe Veber. Detecting inconsistencies in large biological networks with answer set programming. *Theory and Practice of Logic Programming*, 11(2-3):323–360, 2011.

[Gelder *et al.*, 1991] Allen Van Gelder, Kenneth A. Ross, and John S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, 1991.

[Gelfond and Lifschitz, 1988] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In R. Kowalski and K. A. Bowen, editors, *Logic Programming, Proceedings of the 5th International Conference (ICLP'88)*, pages 1070–1080. MIT Press, 1988.

[Heyninck and Straßer, 2016] Jesse Heyninck and Christian Straßer. Relations between assumption-based approaches in nonmonotonic logic and formal argumentation. *CoRR*, abs/1604.00162, 2016.

[Kakas and Moraitis, 2003] Antonis C. Kakas and Pavlos Moraitis. Argumentation based decision making for autonomous agents. In *Proceedings of the 2003 International Conference* *on Autonomous Agents & Multiagent Systems (AAMAS'03)*, pages 883–890, Melbourne, 2003. ACM Press.

[Kakas and Toni, 1999] Antonis C. Kakas and Francesca Toni. Computing argumentation in logic programming. *Journal of Logic and Computation*, 9(4):515–562, 1999.

[Kowalski and Toni, 1996] Robert A. Kowalski and Francesca Toni. Abstract argumentation. *Artificial Intelligence and Law*, 4(3–4):275–296, 1996.

[Leone *et al.*, 2006] Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic*, 7(3):499–562, 2006.

[Lifschitz, 2002] Vladimir Lifschitz. Answer set programming and plan generation. *Artificial Intelligence*, 138(1-2):39–54, 2002.

[McCarthy, 1980] John McCarthy. Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1-2):27–39, 1980.

[Mocanu *et al.*, 2016] Andrei Mocanu, Xiuyi Fan, Francesca Toni, Matthew Williams, and Jiarong Chen. Online argumentation-based platform for recommending medical literature. In I. Hatzilygeroudis, V. Palade, and J. Prentzas, editors, *Combinations of Intelligent Methods and Applications*, volume 46 of *Smart Innovation, Systems and Technologies*, pages 97–115. Springer, 2016.

[Modgil and Prakken, 2014] Sanjay Modgil and Henry Prakken. The ASPIC+ framework for structured argumentation: A tutorial. *Argument & Computation*, 5(1):31–62, 2014.

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

[Moore, 1985] Robert Moore. Semantical considerations on non-monotonic logic. *Artificial Intelligence*, 25(1):75–94, 1985.

[Newton-Smith, 1981] William H. Newton-Smith. *The Rationality of Science*. Routledge, 1981.

[Prakken and Sartor, 1999] Henry Prakken and Giovanni Sartor. A system for defeasible argumentation, with defeasible priorities. In Michael Wooldridge and Manuela Veloso, editors, *Artificial Intelligence Today*, volume 1600 of *Lecture Notes in Computer Science*, pages 365–379. Springer, 1999.

[Reiter, 1980] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

[Ricca *et al.*, 2010] Francesco Ricca, Antonella Dimasi, Giovanni Grasso, Salvatore Maria Ielpa, Salvatore Iiritano, Marco Manna, and Nicola Leone. A logic-based system for e-tourism. *Fundamenta Informaticae*, 105(1-2):35–55, 2010.

[Ricca *et al.*, 2012] Francesco Ricca, Giovanni Grasso, Mario Alviano, Marco Manna, Vincenzino Lio, Salvatore Iiritano, and Nicola Leone. Team-building with answer set programming in the Gioia-Tauro Seaport. *Theory and Practice of Logic Programming*, 12(3):361–381, 2012.

[Sakama and Inoue, 2000] Chiaki Sakama and Katsumi Inoue. Prioritized logic programming and its application to commonsense reasoning. *Artificial Intelligence*, 123(1-2):185–222, 2000.

[Schulz and Toni, 2014] Claudia Schulz and Francesca Toni. Complete assumption labellings. In *Proceedings of the 2014 Conference on Computational Models of Argument (COMMA'14)*, pages 405–412, 2014.

[Schulz and Toni, 2015] Claudia Schulz and Francesca Toni. Logic programming in assumption-based argumentation revisited - semantics and graphical representation. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, pages 1569–1575, 2015.

[Schulz and Toni, 2016] Claudia Schulz and Francesca Toni. Justifying answer sets using argumentation. *Theory and Practice of Logic Programming*, 16(1):59–110, 2016.

[Schulz and Toni, 2017] Claudia Schulz and Francesca Toni. Labellings for assumption-based and abstract argumentation. *International Journal of Approximate Reasoning*, 84:110–149, 2017.

[Schulz *et al.*, 2015] Claudia Schulz, Ken Satoh, and Francesca Toni. Characterising and explaining inconsistency in logic programs. In Francesco Calimeri, Giovambattista Ianni, and Miroslaw Truszczynski, editors, *Proceedings of the 13th International Conference on*

Logic Programming and Nonmonotonic Reasoning (LPNMR'15), volume 9345 of *Lecture Notes in Computer Science*, pages 467–479. Springer, 2015.

[Seselja and Straßer, 2013] Dunja Seselja and Christian Straßer. Abstract argumentation and explanation applied to scientific debates. *Synthese*, 190(12):2195–2217, 2013.

[Simari and Loui, 1992] Guillermo Ricardo Simari and Ronald Prescott Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53(2-3):125–157, 1992.

[Terracina *et al.*, 2013] Giorgio Terracina, Alessandra Martello, and Nicola Leone. Logic-based techniques for data cleaning: An application to the Italian national healthcare system. In *Proceedings of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'13)*, pages 524–529, 2013.

[Thang and Luong, 2013] Phan Minh Thang and H.T. Luong. Translating preferred subtheories into structured argumentation. *Journal of Logic and Computation*, 2013.

[Toni, 2008a] Francesca Toni. Assumption-based argumentation for closed and consistent defeasible reasoning. In Ken Satoh, Akihiro Inokuchi, Katashi Nagao, and Takahiro Kawamura, editors, *New Frontiers in Artificial Intelligence, JSAI 2007 Conference and Workshops, Miyazaki, Japan, June 18-22, 2007, Revised Selected Papers*, volume 4914 of *Lecture Notes in Computer Science*, pages 390–402. Springer, 2008.

[Toni, 2008b] Francesca Toni. Assumption-based argumentation for epistemic and practical reasoning. In Pompeu Casanovas, Giovanni Sartor, Nuria Casellas, and Rossella Rubino, editors, *Computable Models of the Law, Languages, Dialogues, Games, Ontologies*, volume 4884 of *Lecture Notes in Computer Science*, pages 185–202. Springer, 2008.

[Toni, 2012] Francesca Toni. Reasoning on the web with assumption-based argumentation. In Thomas Eiter and Thomas Krennwallner, editors, *Reasoning Web. Semantic Technologies for Advanced Query Answering. 8th International Summer School. Proceedings.*, volume 7487 of *LNCS*, pages 370–386. Springer, 2012.

[Toni, 2013] Francesca Toni. A generalised framework for dispute derivations in assumption-based argumentation. *Artificial Intelligence*, 195:1–43, 2013.

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

[Wakaki, 2014] Toshiko Wakaki. Assumption-based argumentation equipped with preferences. In Hoa Khanh Dam, Jeremy V. Pitt, Yang Xu, Guido Governatori, and Takayuki Ito, editors, *Principles and Practice of Multi-Agent Systems - 17th International Conference (PRIMA'14)*, volume 8861 of *Lecture Notes in Computer Science*, pages 116–132. Springer, 2014.

[Zhong *et al.*, 2014] Qiaoting Zhong, Xiuyi Fan, Francesca Toni, and Xudong Luo. Explaining best decisions via argumentation. In *Proceedings of the European Conference on Social Intelligence (ECSI'14)*, pages 224–237, 2014.

Kristijonas Čyras
Imperial College London, UK
Email: k.cyras@imperial.ac.uk

Xiuyi Fan
Nanyang Technological University, Singapore
Email: xyfan@ntu.edu.sg

Claudia Schulz
Imperial College London, UK
Email: claudia.schulz@imperial.ac.uk

Francesca Toni
Imperial College London, UK
Email: ft@imperial.ac.uk