Some design guidelines for practical argumentation systems

Phan Minh DUNG [a] Francesca TONI [b,1] Paolo MANCARELLA [c]
[a] *AIT, Bangkok, Thailand*
[b] *Department of Computing, Imperial College London, UK*
[c] *Dipartimento di Informatica, Università di Pisa, Italy*

Abstract. We give some design guidelines for argumentation systems. These guidelines are meant to indicate essential features of argumentation when used to support "practical reasoning". We express the guidelines in terms of postulates. We use a notion of redundancy to provide a formal counterpart of these postulates. We study the satisfaction of these postulates in two existing argumentation frameworks: assumption-based argumentation and argumentation in classical logic.

Keywords. Formal and informal models for argumentation

Introduction

Argumentation is widely recognised as an important mechanism to support "practical reasoning", e.g. in support of debate [6,3] and for legal reasoning [1]. As an example of the kind of "practical reasoning" we have in mind, consider the following situation: *You are presenting your case of why you should become the president of your country to your voters. These will form a mixed audience, in that some may be of basic education and simple professions, others will be educated or highly-educated. You will have to face one or more opponents, who would also like to become president and will attack your case. How should you conduct and participate in such a debate so as to have a chance to win it?* We believe that the answer lies in the following cardinal principles:

Principle 1 : Your arguments must be simple so that people of all backgrounds can follow them easily. It does not make sense to deliver some sophisticated arguments that only experts can understand, possibly after spending considerable effort. You should make clear the contrast between you and your opponents by attacking their arguments in a simple, transparent way, immediately obvious to the audience.

Principle 2 : Deliver your arguments in full to make your points explicitly, but avoid repetitions and irrelevant details that could be a distraction and a point of attack by your opponents.

Principle 3 : Do not disregard or dismiss any arguments by your opponents, unless you are certain that you can rebate them at any time. Do not disregard or dismiss any argument against the opponents, to avoid losing your edge and be perceived as the loser.

[1] Corresponding Author

We explore how these principles can be formalised as guidelines for *generic argumentation systems*, in the form of *postulates*. These systems are given in terms of arguments, viewed as defeasible proofs in some (unspecified) logic, and an attack relation, meant to be used to capture those defeasible proofs that can be deemed acceptable by parties engaged in practical reasoning. We do not prescribe any notion of acceptability of arguments. We allow for some defeasible proofs not to be deployed as arguments in practical reasoning tasks. Thus, not every defeasible proof is an argument in general.

Several well-known argumentation frameworks can be seen as instances of our generic argumentation systems. We analyse the fulfilment of these postulates in the context of some existing logic-based argumentation systems, focusing on assumption-based argumentation (ABA) [8] and argumentation in classical logic (ArCL) [3]. This analysis relies upon a formal understanding of the postulates in terms of a notion of redundancy of arguments, that we provide, in part, in the context of abstract argumentation [6].

The paper is organised as follows. In section 1 we give some background on abstract argumentation, ABA and ArCL. In section 2 we give our postulates for generic argumentation frameworks. In section 3 we reformulate (two of) these postulates in terms of a "less redundant" relation. In sections 4 and 5 we analyse these postulates in ArCL and ABA, respectively. In section 6 we conclude.

1. Background

Abstract argumentation [6]

An **abstract argumentation framework** is a pair $\langle Arg, att \rangle$ where Arg is a finite set, whose elements are referred to as *arguments*, and $att \subseteq Arg \times Arg$ is a binary relation over Arg. Given $\alpha, \beta \in Arg$, α **attacks** β iff $(\alpha, \beta) \in att$. Given sets $X, Y \subseteq Arg$ of arguments, X **attacks** Y iff there exists $x \in X$ and $y \in Y$ such that $(x, y) \in att$. A set of arguments is referred to as *extension*. An extension $X \subseteq Arg$ is

- **acceptable wrt** a set $Y \subseteq Arg$ of arguments iff for each β that attacks an argument in X, there exists $\alpha \in Y$ such that α attacks β;
- **admissible** iff X does not attack itself and X is acceptable wrt itself;
- **preferred** iff X is (subset) maximally admissible;
- **complete** iff X is admissible and X contains all arguments x such that $\{x\}$ is acceptable wrt X;
- **grounded** iff X is (subset) minimally complete;
- **ideal** iff X is admissible and it is contained in every preferred set of arguments.

For $AF = \langle Arg, att \rangle$, the **characteristic function** $\mathcal{F}_{AF}$ is such that $\mathcal{F}_{AF}(X)$ is the set of all acceptable arguments wrt X. Then X that does not attack itself is 1) an admissible extension iff $X \subseteq \mathcal{F}_{AF}(X)$, 2) a complete extension iff it is a fix-point of $\mathcal{F}_{AF}$, and 3) a grounded extension iff X is the least fix-point of $\mathcal{F}_{AF}$.

Assumption-based argumentation (ABA) [4,7,9,8]

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

- $(\mathcal{L}, \mathcal{R})$ is a *deductive system*, with $\mathcal{L}$ a language and $\mathcal{R}$ a set of inference rules,
- $\mathcal{A} \subseteq \mathcal{L}$, referred to as the set of *assumptions*,
- $\overline{}$ is a (total) mapping from $\mathcal{A}$ into $\mathcal{L}$, where $\overline{x}$ is referred to as the *contrary* of x.

We will assume that inference rules have the syntax $\frac{s_1,\ldots s_n}{s_0}$ (for $n \geq 0$) where $s_i \in \mathcal{L}$.

An *argument* in favour of a sentence c in $\mathcal{L}$ supported by a set of assumptions A is a (defeasible) proof of c from A and (some of) the rules in $\mathcal{R}$. We will provide a formal definition of argument in section 5 (see definition 5.2). For the purposes of defining semantics and computational mechanisms for ABA, the notation $\langle A, c\rangle$ is used, to stand for an argument for c supported by A. This notation can be seen as a shorthand for the notation $\langle A, P, c\rangle$, where P is the proof used to construct the argument. The (short) notation does not distinguish arguments with the same support and conclusion but using different proofs. E.g. $\langle\{a\}, p\rangle$ may represent an argument using inference rule $\frac{a}{p}$ as well as an argument using inference rules $\frac{q}{p}$, $\frac{a}{q}$ (see [9] for a discussion of this issue). The short notation suffices to define semantics and computational mechanisms for ABA, as the only form of defeasibility in ABA is given by assumptions.

All notions of extensions for abstract argumentation can be used in ABA, by using a notion of "attack" amongst arguments whereby $\langle X, x\rangle$ **attacks** $\langle Y, y\rangle$ iff $x = \overline{\alpha}$ for some $\alpha \in Y$. As shown in [9], theorem 2.2, there is a one-to-one correspondence between semantics in terms of extensions (sets of arguments), as presented here, and semantics in terms of sets of assumptions, as presented in the original definition of ABA in [4].

Argumentation based on classical logic (ArCL) [3]
In ArCL, given a (possibly inconsistent) set of (first-order) sentences Δ, an *argument* for a (first-order) sentence c is a pair $\langle S, c\rangle$ such that

(i) S is consistent
(ii) $S \vdash c$, where $\vdash$ is the classical consequence relation
(iii) S is a minimal subset of Δ fulfilling conditions (i), (ii).

An argument $\langle S_1, c_1\rangle$ is a **canonical undercut for** an argument $\langle S_2, c_2\rangle$ iff

(a) $c_1 = \neg(s_1 \wedge \ldots \wedge s_n)$, for $S_2 = \{s_1, \ldots, s_n\}$
(b) $s_1, \ldots, s_n$ is the canonical enumeration of S_2 (according to some ordering of the elements of Δ given a-priori, without loss of generality).

An *argument tree* for a sentence s is a tree whose nodes are arguments such that

(I) the root is an argument for s
(II) for no node $\langle S, c\rangle$, S is a subset of the union of the supports of the node's ancestors
(III) the children of a node N are all canonical undercuts for N obeying (II).

Nodes of a tree can be marked as undefeated (U) or defeated (D) as follows: for all nodes N, if there is a child of N marked U, then N is marked D; otherwise, N is marked U. Then, an argument tree is **warranted** iff the root of the tree is marked U.

As for ABA, we will see an argument $\langle S, c\rangle$ in ArCL as a shorthand for $\langle S, P, c\rangle$ where P is a proof (e.g. using natural deduction) for c using the sentences in S.

2. Postulates for practical argumentation

In this section we consider generic argumentation frameworks, equipped with notions of

- *legitimate arguments*, as defeasible proofs in some (possibly implicit) underlying logic; each legitimate argument consists of a support, a proof and a claim;
- *"deployed" arguments*, namely (legitimate) arguments that can be deployed in practical argumentation;

- *attack* between arguments, as a binary relation that may be primitive or derived (from primitive notions);
- "dialectical" *semantics* for accepting sets of legitimate arguments.

We will use the terminology *"illegitimate" argument* for any argument that is not legitimate. Illegitimate arguments may be anything, e.g. unsupported claims or supported claims with invalid or without proofs. They may be introduced by any parties involved in the exchange of arguments, e.g., a witness with an incoherent account of past events.

Several existing argumentation frameworks could be seen as providing notions of legitimate arguments in our generic sense, e.g. ABA, ArCL, DeLP [10], just to mention some. These various concrete frameworks differ in the underlying logic (e.g. any deductive system in ABA and standard notions of logical deduction in first-order logic for ArCL), in how their choice of deployed arguments (e.g. ABA requires that deployed arguments can be constructed backwards and ArCL requires minimality and consistency of their support), and in how they define the attack relation as a derived notion (e.g. from a primitive notion of contrary of assumptions in ABA). Finally, the concrete frameworks differ in their choice of dialectical semantics (e.g. ABA uses notions of extensions, ArCL uses a notion of warranted argument trees and DeLP uses a notion of warranted literals).

Our generic argumentation frameworks, for specific choices of "dialectical" semantics, with the set of deployed arguments coinciding with the set of legitimate arguments, can be seen as instances of abstract argumentation frameworks.

We give postulates, intended as design guidelines for generic argumentation systems *when these are used for practical reasoning* (as understood in this paper).

Postulate 2.1 (**Transparency**) Deployed arguments and attacks should be *transparent* in the sense of being computationally tractable, as follows:

1. the computational cost of verifying that deployed arguments are legitimate should be AT MOST polynomial (in the size of the arguments);
2. the computational cost of verifying that an argument attacks another should be AT MOST linear (in the size of the claim of the argument).

The size of an argument is the size of its support and proof. Intuitively, this postulate guarantees that arguments can be understood by any parties, independently of their level of sophistication (cf. principle 1 in the introduction). Note that the construction of deployed arguments could be highly complex, as for example in the case of legal proceedings. However, once these arguments are constructed, the verification that they are so, e.g. by other parties, should be tractable (i.e. polynomial). Proofs in propositional logic, using for example natural deduction or resolution, can be checked in polynomial time. So arguments as proofs in propositional logic can be deemed to fulfil this postulate.

Also, this postulate forces attacks to be directly inspectable (linear time). This is again in line with principle 1 in the introduction. As attacks need to be immediately recognisable, without any "reasoning" by the spectators in a debate. Attacks defined in terms of inconsistency (namely by sanctioning that the claim of the attacking argument is inconsistent with the support of the attacked argument) would not satisfy this postulate.

Postulate 2.2 (**Relevance**) The support of deployed arguments should be *relevant* to the claim of the arguments, to some degree.

Intuitively, this postulate amounts to forcing proponents and opponents of arguments to focus and avoid digressions not contributing to the important points they want to make and possibly opening up attacks from their counterparts (cf. principle 2 in the introduction). In the strongest sense this postulate 2.2 can be interpreted to mean that the support should be necessary to establish the claim, in the sense that the removal of any part of this support would render the arguments illegitimate. In a weaker sense this can be interpreted to mean that the argument is a defeasible proof of its claim from its support, without any obvious redundancy of any parts of the support.

The first two postulates focus on the inner workings of legitimate, deployed arguments and attacks. The third and final postulate instead considers the use of arguments in "debate", namely in the context of "dialectical" semantics.

Postulate 2.3 (**No dismissal**) No legitimate argument should be dismissed without reason. If, for any reason, some legitimate arguments are dismissed (not deployed), their dismissal should not change the semantics of the given argumentation framework.

This postulate amounts to avoiding leaving any stone unturned (cf. principle 3).

In the remainder we will refer to legitimate arguments simply as arguments. An argument with proof P from support S for a claim c will be represented as a triple $\langle S, P, c\rangle$ or simply as a pair $\langle S, c\rangle$, leaving the proof implicit, if clear from the context.

3. Redundancy

We can formally restate postulates 2.2 and 2.3 in terms of a notion of "redundancy" of arguments, defined in terms of a "less redundant" preference relation:

Postulate 3.1 If a relation $\prec$ represents a *less redundant* relation between arguments, then the following properties should be satisfied:

1. $\prec$ is transitive (i.e., for arguments α, β, γ, if $\alpha \prec \beta$ and $\beta \prec \gamma$, then $\alpha \prec \gamma$) reflexive (i.e. for any argument α, $\alpha \prec \alpha$) and antisymmetric (i.e. for arguments α, β, if $\alpha \prec \beta$ and $\beta \prec \alpha$, then $\alpha = \beta$)
2. given arguments α, β, γ, if $\alpha \prec \beta$ then
   (a) if β attacks γ then α attacks γ;
   (b) if γ attacks α then γ attacks β.
3. for each argument α there is an argument β such that
   (a) there is no other argument γ such that $\gamma \prec \beta$, and
   (b) either $\alpha = \beta$ or $\beta \prec \alpha$.

Intuitively, condition 2a says that α has a stronger conclusive force than β, and condition 2b says that α is less exposed to attacks than β (e.g. because its support is narrower). Condition 3 says that each argument has a "non-redundant" version (possibly itself). Depending on the context, different instances of the "less redundant" relation could be introduced. For example, one could define an argument α as "less redundant than" an argument β (with both arguments supporting the same claim) if the support of α is a subset of the support of β, namely, for $\alpha = \langle S_\alpha, c\rangle$ and $\beta = \langle S_\beta, c\rangle$: $\alpha \prec \beta$ iff $S_\alpha \subseteq S_\beta$ Below, we will refer to this notion of $\prec$ as $\prec_\subseteq$.

We will see, in section 5, another notion of $\prec$ where arguments with proof that can be arranged as trees are "less redundant" than arguments with proofs that cannot.

An example of a relation over pairs of arguments that is *not* a suitable notion of "less redundant" is $\prec$ such that $\alpha \prec \beta$ iff the cardinality of the support of α is strictly smaller than that of the support of β. Indeed, this notion will typically not fulfil condition 2b.

3.1. Redundancy and relevance postulate

Definition 3.1 Given a "less redundant" relation $\prec$, the support of an argument α is *relevant* if α is minimal wrt $\prec$.

For example, given $\prec_{\subseteq}$ given earlier, relevant arguments are arguments with a subset-minimal support. In the context of the presidential debate considered in the introduction, this subset-minimality-based definition of relevance does not seem to be useful. Indeed, its adoption may violate postulate 2.1 in that, for example, the cost of verifying that arguments as proofs in propositional logic have a subset-minimal support may be non-polynomial. Instead, if arguments are proofs built from Horn clauses, then subset-minimality is acceptable, in that the cost of verifying that arguments in Horn logic have a subset-minimal support is polynomial [2].

3.2. Redundancy and no dismissal postulate

Arguments can be deemed to be *redundant* if there exist other arguments that are "less redundant" than them.

Definition 3.2 Given a "less redundant" relation $\prec$, an argument α is *redundant* (wrt $\prec$) if THERE EXISTS AN ARGUMENT $\beta \neq \alpha$ SUCH THAT $\beta \prec \alpha$. The set NR of all *non-redundant* arguments (wrt $\prec$) is such that for each argument α there is $\beta \in NR$ such that $\beta \prec \alpha$.

Theorem 3.1 below states that redundant arguments can be dismissed without affecting (some) "dialectical" semantics. Thus, non-redundant arguments, in the sense of definition 3.2, can be seen as fulfilling postulate 2.3 (for these semantics).

Theorem 3.1 considers (some of) the semantics that have been studied for abstract argumentation frameworks [6,9]. Thus, for the purposes of this theorem, we treat (legitimate) arguments as abstract and assume an abstract argumentation framework $\langle Arg, att\rangle$ where att is the attack relation. The theorem is formulated using the following notion of "$\prec$-trimmed" versions of abstract argumentation framework:

Definition 3.3 Let $\langle Arg, att\rangle$ be an abstract argumentation framework. Let $\prec$ be a "less redundant" relation between arguments in Arg and $\mathcal{NR} \subseteq Arg$ the set of all non-redundant arguments (wrt $\prec$). Further, let $att_{\mathcal{NR}}$ be the restriction of att on $\mathcal{NR}$, i.e. $att_{\mathcal{NR}} = att \cap \mathcal{NR} \times \mathcal{NR}$. The argumentation framework $\langle \mathcal{NR}, att_{\mathcal{NR}}\rangle$ is referred to as the *$\prec$-trimmed version of* $\langle Arg, att\rangle$.

Theorem 3.1 follows directly from the following lemmas (see appendix A for all proofs):

Lemma 3.1 Let $\langle \mathcal{NR}, att_{\mathcal{NR}}\rangle$ be the *$\prec$-trimmed version of* $\langle Arg, att\rangle$. Then

1. If $\beta \in \mathcal{NR}$ is acceptable wrt $B \subseteq \mathcal{NR}$ in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$ then β is acceptable wrt B in $\langle Arg, att \rangle$
2. Each admissible set of arguments in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$ is also admissible in $\langle Arg, att \rangle$.
3. If $\alpha \in Arg$ is acceptable wrt $B \subseteq Arg$ in $\langle Arg, att \rangle$ then each $\beta \in Arg$ such that $\beta \prec \alpha$ is acceptable wrt B in $\langle Arg, att \rangle$.
4. If $\alpha \in Arg$ is acceptable wrt $B \subseteq \mathcal{NR}$ in $\langle Arg, att \rangle$ then each $\beta \in \mathcal{NR}$ such that $\beta \prec \alpha$ is acceptable wrt B in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$.

Lemma 3.2 Let $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$ be the *$\prec$-trimmed version of* $\langle Arg, att \rangle$. Also, let $\mathcal{C}$ and $\mathcal{C}_{\mathcal{NR}}$ be the sets of complete extensions of $\langle Arg, att \rangle$ and $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$ respectively. Further, let $AF = \langle Arg, att \rangle$. Then, $\mathcal{F}_{AF}(X)$ is a monotonic (wrt set inclusion) bijection from $\mathcal{C}_{\mathcal{NR}}$ onto $\mathcal{C}$ such that

1. For each $X \in \mathcal{C}_{\mathcal{NR}}$: $\mathcal{F}_{AF}(X) \cap \mathcal{NR} = X$.
2. For each $X \in \mathcal{C}$: $\mathcal{F}_{AF}(X \cap \mathcal{NR}) = X$.

Theorem 3.1 Let $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$ be the *$\prec$-trimmed version of* $AF = \langle Arg, att \rangle$. Then

1. Let X be a complete or preferred or grounded or ideal extension in $\langle Arg, att \rangle$. Then $X \cap \mathcal{NR}$ is a complete or preferred or grounded or ideal extension, respectively, in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$.
2. Let X be a complete or preferred or grounded or ideal extension in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$. Then $\mathcal{F}_{AF}(X)$ is a complete or preferred or grounded or ideal extension, respectively, in $\langle Arg, att \rangle$.

This theorem shows that removing or adding redundant arguments (for any notion of $\prec$) does not change the semantics (of complete, preferred, grounded, and ideal extensions for abstract argumentation) of the underlying argumentation framework. Hence the issue of whether to work with an argumentation framework with less or more redundant arguments is purely of computational efficiency nature.

Consider instead dismissing arguments that are self-attacking. The dismissal of these arguments would violate postulate 2.3 if, for example, the chosen dialectical semantics is that of admissible extensions. Indeed, consider an argumentation framework with arguments α, β such that α attacks β and α attacks itself. Then, $\{\beta\}$ is not an admissible extension in conventional abstract argumentation but would be an admissible extension if self-attacking arguments were dismissed.

4. Postulates for practical argumentation in ArCL

ArCL can be seen as an instance of our generic argumentation frameworks where legitimate arguments are of the (abbreviated) form $\langle S, c \rangle$ such that $S \vdash c$, deployed arguments are arguments in the sense of section 1, attacks are canonical undercuts, and the "dialectical" semantics is given by the notion of warranted trees.

According to this understanding of ArCL, this fulfils the relevance postulate 2.2 in the sense that it uses relevant arguments wrt $\prec_{\subseteq}$ (see section 3).

As far as the transparency postulate 2.1 is concerned, in ArCL:

1. checking consistency of arguments' support cannot be done in polynomial time;

2. both conditions (a) and (b) of the definition of attack (canonical undercut) can be checked in linear time (in the size of the claim of the attacking argument).

Thus, overall, ArCL does not fulfil this postulate.

Finally, ArCL can be deemed not to fulfil the no dismissal postulate 2.3, in the sense that, by disregarding legitimate arguments on the ground that their support is "not new" within a given argument tree (namely, this support is a subset of the support of previously used arguments), one obtains different warranted argument trees. For example, consider $\Delta = \{p, q, \neg p \vee \neg q\}$ (with canonical enumeration $1 : p$, $2 : q$, $3 : \neg p \vee \neg q$). The only argument tree for p has root $\langle \{p\}, p\rangle$ with a single child N=$\langle \{q, \neg p \vee \neg q\}, \neg p\rangle$. The argument tree is unwarranted. Here, the legitimate argument N'=$\langle \{p, q\}, \neg(\neg p \vee \neg q)\rangle$, attacking N, is not allowed as a child of N as its support is "not new". However, If N' had been considered, as a child of N, then the tree would have been warranted.

5. Postulates for practical argumentation in ABA

In this section, unless otherwise specified, we assume as given a generic $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}\rangle$.

Three kinds of arguments have been defined for ABA: forward arguments, as given in definition 5.1 below (adapted from [4,7]); backward arguments, as given in definition 5.2 below (adapted from [7,9]); and tree-arguments, as given in definition 5.3 below (adapted from [8]). As discussed in [8], backward arguments can be seen as constructing, in a top-down manner, tree-arguments. All these kinds of arguments can be seen as legitimate arguments in ABA, but the deployed arguments are backward arguments.

Definition 5.1 A *forward argument* for $c \in \mathcal{L}$ supported by $A \subseteq \mathcal{A}$ is a sequence $\beta_1, \ldots, \beta_m$, of sentences in $\mathcal{L}$, where $m > 0$ and $c = \beta_m$, such that, for all $i = 1, \ldots, m$,

- $\beta_i \in A$, or
- there exists $\frac{s_1,\ldots,s_n}{\beta_i} \in \mathcal{R}$ such that $s_1, \ldots, s_n \in \{\beta_1, \ldots, \beta_{i-1}\}$.

We use the notation $\langle A, [\beta_1, \ldots, \beta_m], c\rangle_f$ for a forward argument for c supported by A with proof $\beta_1, \ldots, \beta_m$. We use the shorthand $\langle A, c\rangle_f$ when the proof is irrelevant.

Definition 5.2 Given a selection function f [2], a *backward argument* for $c \in \mathcal{L}$ supported by $A \subseteq \mathcal{A}$ is as sequence of multi-sets $S_1, \ldots, S_m$, where $S_1 = \{c\}$, $S_m = A$, and for every $1 \leq i < m$, where σ is the sentence occurrence in S_i selected by f:

1. If $\sigma \notin \mathcal{A}$ then $S_{i+1} = S_i - \{\sigma\} \cup S$ for some $\frac{S}{\sigma} \in \mathcal{R}$.
2. If $\sigma \in \mathcal{A}$ then $S_{i+1} = S_i$.

We use the notation $\langle A, [S_1, \ldots, S_m], c\rangle$ for a backward argument for c supported by A with proof $S_1, \ldots, S_m$. We use the shorthand $\langle A, c\rangle$ when the proof is irrelevant.

Definition 5.3 A *tree-argument* for $c \in \mathcal{L}$ supported by $A \subseteq \mathcal{A}$ is a tree $\mathcal{T}$ with nodes labelled by sentences in $\mathcal{L}$ or by τ, [3] such that

[2] A selection function takes in input a sequence of multi-sets and returns as output a sentence occurring in the last multi-set in the sequence [7].

[3] The symbol τ intuitively stands for "true". It allows to distinguish between facts, namely inference rules with an empty set of premises, and assumptions.

- the root is labelled by c
- for every node N
  - $*$ if N is a leaf then N is labelled either by an assumption or by τ;
  - $*$ if N is not a leaf and l_N is the label of N, then there is an inference rule $l_N \leftarrow b_1, \ldots, b_m$ ($m \geq 0$) and
    either $m = 0$ and the child of N is τ
    or $m > 0$ and N has m children, labelled by $b_1, \ldots, b_m$ (respectively)
- A is the set of all assumptions labelling the leaves.

We use the notation $\langle A, \mathcal{T}, c\rangle_t$ for a tree-argument for c supported by A with proof $\mathcal{T}$. We use the shorthand $\langle A, c\rangle_t$ when the proof is irrelevant.

It is easy to see that

- there is a backward argument $\langle A, c\rangle$ iff there is a tree-argument $\langle A, c\rangle_t$

Namely, the notions of backward argument and tree-argument are equivalent. Moreover, by theorem 4.1 in [7]:

- for every backward argument $\langle A, c\rangle$ (or tree-argument $\langle A, c\rangle_t$) there is a forward argument $\langle A, c\rangle_f$
- for every forward argument $\langle A, c\rangle_f$ there is a backward argument $\langle A', c\rangle$ (and tree-argument $\langle A', c\rangle_t$) for some $A' \subseteq A$

In other words, forward arguments may have "redundancies" in their support. We can thus define a notion $\prec_{tree}$ of "less redundant" as follows:

Definition 5.4 Given forward arguments $\langle A_1, c\rangle_f$ and $\langle A_2, c\rangle_f$, $\langle A_1, c\rangle_f \prec_{tree} \langle A_2, c\rangle_f$ iff EITHER THERE EXIST TREE ARGUMENTS $\langle A_1, c\rangle_t$ AND $\langle A_2, c\rangle_t$ OR there exists a tree-argument $\langle A_1, c\rangle_t$ but there exists no tree-argument $\langle A_2, c\rangle_t$.

Lemma 5.1 $\prec_{tree}$ is a "less redundant" relation, in the sense of postulate 3.1.

Note that it may be the case that $\langle A_1, c\rangle_f \prec_{\subseteq} \langle A_2, c\rangle_f$ but $\langle A_1, c\rangle_f \not\prec_{tree} \langle A_2, c\rangle_f$. For example, let $\mathcal{R}$ be $\{\frac{a}{p}\}$ and $\mathcal{A} = \{a, b, e\}$. Then $\langle\{a, b\}, p\rangle_f \prec_{\subseteq} \langle\{a, b, e\}, p\rangle_f$ but $\langle\{a, b\}, p\rangle_f \not\prec_{tree} \langle\{a, b, e\}, p\rangle_f$. However, given a relevant (wrt $\prec_{\subseteq}$) argument $\langle A_1, c\rangle_f$, if $\langle A_1, c\rangle_f \prec_{\subseteq} \langle A_2, c\rangle_f$ then $\langle A_1, c\rangle_f \prec_{tree} \langle A_2, c\rangle_f$. In the earlier example, $\langle\{a\}, p\rangle_f$ is (the only) relevant argument (wrt $\prec_{\subseteq}$), $\langle\{a\}, p\rangle_f \prec_{\subseteq} \langle\{a, b, e\}, p\rangle_f$ and indeed $\langle\{a\}, p\rangle_f \prec_{tree} \langle\{a, b, e\}, p\rangle_f$.

The relevance postulate 2.2 and no dismissal postulate 2.3 hold for all instances of ABA, for the notion $\prec_{tree}$ of "less redundant". In particular, postulate 2.2 holds since:

Property 5.1 Backward arguments are ARGUMENTS WITH RELEVANT SUPPORT in the sense of definition 3.1, wrt $\prec_{tree}$.

Moreover, postulate 2.3 holds because, by focusing on backward arguments, ABA solely dismisses forward arguments that are redundant.

Property 5.2 The set of all non-redundant arguments wrt $\prec_{tree}$, in the sense of definition 3.2, is the set of all backward arguments.

Theorem 3.1 then holds for the $\prec_{tree}$-trimmed version of (the abstract argumentation framework corresponding to) any ABA framework. Theorem 4.2 in [7] is a corollary of lemma 3.1 used to prove our theorem 3.1.

We analyse postulate 2.1 in two of of the existing instances of ABA, studied in [4].

5.1. *ABA for logic programming*

A *(normal) logic program* [4] is a set of rules of the form $p \leftarrow l_1, \ldots, l_n$ where p is an atom, $l_1, \ldots, l_n$ are literals, and $n \geq 0$. Negative literals, $not\ q$, are the *negation as failure* of atoms, q. Logic programs P can be represented as ABA frameworks where

- $\mathcal{R} = \{\frac{B}{p} | p \leftarrow B \in P\}$
- $\mathcal{A} = \{not\ p | p$ is an atom in the Herbrand base of $P\}$
- $\overline{not\ p} = p$ for all $not\ p \in \mathcal{A}$

This instance of ABA fulfils the transparency postulate 2.1 since

1. the computational cost of verifying that deployed arguments (namely backward arguments) are legitimate arguments (namely proofs) is linear in the size of the argument (number of rules and assumptions used in the argument);
2. the computational cost of verifying that an argument attacks another is constant (a syntactical check on the atom that is the conclusion of the first argument) and thus linear.

5.2. *ABA for default logic*

A *default theory* [14] is a pair (T, D) where

- $T \subseteq \mathcal{L}_0$, where $\mathcal{L}_0$ is a first-order language
- D is a set of rules $\frac{s_0, M s_1, \ldots, M s_n}{s}$ where $s_0, \ldots, s_n,\ s \in \mathcal{L}_0$, and $n \geq 0$.

Default theories (T, D) can be represented as ABA frameworks $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ where, given some deductive system $(\mathcal{L}_0, \mathcal{R}_0)$ for classical first-order logic:

- $\mathcal{L}$=$\mathcal{L}_0 \cup \{M\alpha \,|\, \alpha \in \mathcal{L}_0\}$, $\mathcal{R} = \mathcal{R}_0 \cup D$
- $\mathcal{A} = \{M\ s | s \in \mathcal{L}_0\}$
- $\overline{M\ s} = \neg s$ for all $M\ s \in \mathcal{A}$

For example, $\mathcal{R}_0$ may be the set of inference rules for natural deduction, including (all instances of the schemes) $\frac{A,B}{A \wedge B}$ $(\wedge I)$, $\frac{A, A \rightarrow B}{B}$ $(\rightarrow E)$, etc, and all tautologies (as inference rules with empty premises).

This instance of ABA FULFILS the transparency postulate 2.1, since

1. the computational cost of verifying that deployed arguments (namely backward arguments) are legitimate arguments (namely PROOFS) is linear in the size of the argument (number of rules and assumptions used in the argument);
2. the computational cost of verifying that an argument attacks another is linear (since this verification is a syntactical test requiring to scan a sentence s, and is thus linear in the size of s).

[4] We focus here on propositional logic programs.

6. Conclusions

We have identified a number of postulates for generic argumentation systems. These postulates are meant to assess the suitability of argumentations systems to support practical reasoning, e.g. of the form needed in debate in front of audiences of mixed expertise. This suitability amounts to tranparency (of arguments and attacks), relevance (of the support of arguments) and no dismissal of arguments that may make a difference. We have analysed the fulfilment of these postulates in (two instances of) assumption-based argumentation (ABA) and in argumentation in classical logic (ArCL). Our analysis is solely in the context of the form of practical reasoning we envisage, and does not determine the usability of argumentation frameworks in other settings (e.g. for non-monotonic reasoning, or in support of decision-making, or as a mechanism for resolving inconsistencies).

Other authors have considered postulates for argumentation, notably [5]. However, their focus is on *rationality* postulates for rule-based argumentation systems with strict and defeasible rules, whereas our focus is on any argumentation system when used in support of practical reasoning.

For lack of space, we have omitted to consider other argumentation frameworks: we plan to do so in the future. As an example, it would be interesting to consider DeLP [10] (also an instance of our generic argumentation frameworks, as mentioned in section 2): we envisage that this will not fulfil transparency (because attacks in DeLP cannot be detected in constant time) and no dismissal (in the same sense that ArCL does not). Other argumentation frameworks we plan to study include Prakken and Sartor's [13] and Carneades [11].

Our list of postulates is not exhaustive. It would be interesting to consider other postulates, e.g. concerning the avoidance of obvious repetitions in debate and, as in [12], the relevance of all arguments put forward in the debate to the starting point of the debate.

References

[1] T. Bench-Capon, H. Prakken, and G. Sartor. Argumentation in legal reasoning. In I. Rahwan and G. Simari, editors, *Argumentation in AI: The Book*, pages 363–382. Springer, 2009.

[2] J. Bentahar and Z. Maamar. Complexity results for argumentation-based agent communication. In *Proc. of the 4th IEEE Int. Conf. on Innovations in Information Technology*, pages 506–510, 2008.

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

[4] A. Bondarenko, P.M. Dung, R.A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artif. Intell.*, 93(1-2):63–101, 1997.

[5] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.

[6] P.M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77:321–357, 1995.

[7] P.M. Dung, R.A. Kowalski, and F. Toni. Dialectic proof procedures for assumption-based, admissible argume ntation. *Artif. Intell.*, 170:114–159, 2006.

[8] P.M. Dung, R.A. Kowalski, and F. Toni. Assumption-based argumentation. In I. Rahwan and G. Simari, editors, *Argumentation in AI: The Book*, pages 199–218. Springer, 2009.

[9] P.M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artif. Intell., Special Issue on Argumentation in Artificial Intelligence*, 171(10–15):642–674, 2007.

[10] A. Garcia and G. Simari. Defeasible logic programming: An argumentative approach. *Journal of Theory and Practice of Logic Programming*, 4(1-2):95–138, 2004.

[11] T. F. Gordon, H. Prakken, and D. Walton. The Carneades model of argument and burden of proof. *Artif. Intell.*, 171(10-15):875–896, 2007.

[12] S. Parsons, P. McBurney, E. Sklar, and M. Wooldridge. On the relevance of utterances in formal inter-agent dialogues. In *AAMAS*, page 240, 2007.
[13] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7(1):25–75, 1997.
[14] R. Reiter. A logic for default reasoning. *Artif. Intell.*, 13(1-2):81–132, 1980.

A. Proofs

Proof of lemma 3.1

1. Let $\alpha \in Arg$ attack β. Hence there is $\alpha' \in \mathcal{NR}$ such that $\alpha' \prec \alpha$. Hence α' attacks β in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$. Therefore there is $\gamma \in B$ such that γ attacks α'. Hence γ attacks α.
2. Follows directly from item 1 of this lemma.
3. Let $\beta \prec \alpha$ and $\gamma \in Arg$ such that γ attacks β. Therefore γ attacks α. Hence there is $\sigma \in B$ attacking γ.
4. Follows directly from item 3 of this lemma.

Proof of lemma 3.2
The monotonicity of $\mathcal{F}_{AF}(X)$ wrt set inclusion is obvious.

1. We show that $\mathcal{F}_{AF}(X)$ is indeed a function from $\mathcal{C}_{\mathcal{NR}}$ into $\mathcal{C}$ such that $\mathcal{F}_{AF}(X) \cap \mathcal{NR} = X$ by showing that $\mathcal{F}_{AF}(X)$ is a complete extension in $\langle AF, att \rangle$ if X is a complete extension in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$.
   Let X be a complete extension in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$. Hence, by lemma 3.1, item 2, X is admissible in $\langle AF, att \rangle$. Further, by lemma 3.1, item 4, for each $\alpha \in \mathcal{F}_{AF}(X)$, each "non-redundant version of α" is acceptable wrt X in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$. Thus, as X is complete in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$, each "non-redundant version of α" belongs to X. Therefore, $(\mathcal{F}_{AF}(X) - X) \cap \mathcal{NR} = \emptyset$. Hence $\mathcal{F}_{AF}(X) \cap \mathcal{NR} = X$.
   To prove that $\mathcal{F}_{AF}(X)$ is a complete extension, let β be acceptable wrt $\mathcal{F}_{AF}(X)$ and let γ attack β. Hence, there is $\sigma \in \mathcal{F}_{AF}(X)$ attacking γ. Hence, any "non-redundant version of σ" is in X and attacking γ. Hence, X attacks γ. Thus, β is acceptable wrt X, and therefore $\beta \in \mathcal{F}_{AF}(X)$. As a consequence, $\mathcal{F}_{AF}(X)$ is complete.
2. Let C be a complete extension in $\langle AF, att \rangle$. We show that $C' = C \cap \mathcal{NR}$ is complete in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$. From lemma 3.1, item 3, and the fact that C is complete in $\langle AF, att \rangle$, each "non-redundant version of arguments" in C belongs to C.
   Let $\gamma \in \mathcal{NR}$ attack C'. Hence, there is $\alpha \in C$ attacking γ. Let α' be a "non-redundant version of α". Therefore, $\alpha' \in C'$. Hence, α' attacks γ. Hence C' is admissible. Each non-redundant argument acceptable wrt C' is acceptable wrt C and hence belongs to C and hence to C'. C' is therefore complete.
   As $C' \subseteq C$, it is clear that $\mathcal{F}_{AF}(C') \subseteq \mathcal{F}_{AF}(C) = C$. We show now that each argument acceptable wrt C is also acceptable wrt C'. Let β be an argument acceptable wrt C in $\langle AF, att \rangle$ and let σ be an argument attacking β. Hence, there is an argument $\delta \in C$ attacking σ. Hence, there is a "non-redundant version" $\delta' \in C$ of δ attacking σ. Hence C' attacks σ. Thus β is acceptable wrt C' in $\langle AF, att \rangle$. WE HAVE SHOWN THAT $\mathcal{F}_{AF}(C') \supseteq \mathcal{F}_{AF}(C) = C$, I.E. $\mathcal{F}_{AF}(C \cap \mathcal{NR}) = C$.

Proof of theorem 3.1
Let $\mathcal{C}$ and $\mathcal{C}_{\mathcal{NR}}$ be the sets of complete extensions of $\langle Arg, att \rangle$ and $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$, respectively.

From lemma 3.2, it follows immediately that for each $X \in \mathcal{C}_{\mathcal{NR}}$, $\mathcal{F}_{AF}(X)$ is minimal or maximal wrt set inclusion in $\mathcal{C}$ iff X is minimal or maximal respectively in $\mathcal{C}_{\mathcal{NR}}$. Hence X is grounded or preferred in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$ iff $\mathcal{F}_{AF}(X)$ is grounded or preferred in $\langle Arg, att \rangle$, respectively.

Similarly, X is contained in every preferred extension of $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$ iff $\mathcal{F}_{AF}(X)$ is contained in every preferred extension of $\langle Arg, att \rangle$. Hence X is ideal in $\langle \mathcal{NR}, att_{\mathcal{NR}} \rangle$ iff $\mathcal{F}_{AF}(X)$ is ideal in $\langle Arg, att \rangle$.