Available online at www.sciencedirect.com
ScienceDirect
Artificial Intelligence 171 (2007) 642–674
Artificial Intelligence
www.elsevier.com/locate/artint

Computing ideal sceptical argumentation

P.M. Dung [a], P. Mancarella [b], F. Toni [c,*]

[a] *Division of Computer Science, Asian Institute of Technology, PO Box 2754, Bangkok 10501, Thailand*
[b] *Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo, 3 I-56127 Pisa, Italy*
[c] *Department of Computing, Imperial College London, South Kensington Campus, London SW7 2AZ, UK*

Received 8 November 2006; received in revised form 24 April 2007; accepted 2 May 2007

Available online 10 May 2007

Abstract

We present two dialectic procedures for the sceptical *ideal* semantics for argumentation. The first procedure is defined in terms of dispute trees, for abstract argumentation frameworks. The second procedure is defined in dialectical terms, for assumption-based argumentation frameworks. The procedures are adapted from (variants of) corresponding procedures for computing the credulous admissible semantics for assumption-based argumentation, proposed in [P.M. Dung, R.A. Kowalski, F. Toni, Dialectic proof procedures for assumption-based, admissible argumentation, Artificial Intelligence 170 (2006) 114–159]. We prove that the first procedure is sound and complete, and the second procedure is sound in general and complete for a special but natural class of assumption-based argumentation frameworks, that we refer to as *p-acyclic*. We also prove that in the case of p-acyclic assumption-based argumentation frameworks (a variant of) the procedure of [P.M. Dung, R.A. Kowalski, F. Toni, Dialectic proof procedures for assumption-based, admissible argumentation, Artificial Intelligence 170 (2006) 114–159] for the admissible semantics is complete. Finally, we present a variant of the procedure of [P.M. Dung, R.A. Kowalski, F. Toni, Dialectic proof procedures for assumption-based, admissible argumentation, Artificial Intelligence 170 (2006) 114–159] that is sound for the sceptical grounded semantics.

Keywords: Abstract argumentation; Assumption-based argumentation; Ideal semantics; Proof procedure; Dispute

1. Introduction

Argumentation has proven to be a useful abstraction mechanism for understanding several AI problems, for example non-monotonic reasoning (e.g. see [3,7]), defeasible logic (e.g. see [13]) and several forms of reasoning needed to be performed by agents (e.g. see [15]).

Several formulations of argumentation have been proposed, including the frameworks of abstract argumentation [7] and assumption-based argumentation [3,6,8]. For these two frameworks, several semantics have been proposed defining what it means for a set of arguments to be deemed "acceptable" to a rational reasoner. All these semantics rely upon the semantics of admissible arguments [3,7]. This semantics is *credulous*, in that it sanctions a set as "acceptable" if it can successfully dispute every argument against it, without disputing itself. However, there might be conflicting admissible sets. In some applications, it is more appropriate to adopt a *sceptical* semantics, whereby, for

* Corresponding author.
E-mail addresses: dung@cs.ait.ac.th (P.M. Dung), paolo.mancarella@unipi.it (P. Mancarella), ft@doc.ic.ac.uk (F. Toni).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2007.05.003

example, only beliefs sanctioned by all (maximally) admissible sets of assumptions are held. For example, in the legal domain, different members of a jury could hold different admissible sets of assumptions but a guilty verdict must be the result of sceptical reasoning. Also, in a multi-agent setting, agents may have competing plans for achieving goals (where a plan can be interpreted as an argument for the goal it allows to achieve), and, when negotiating resources, they may decide to give away a resource only if that resource is not needed to support *any* of their plans. Furthermore, in the same setting, agents may decide to request an "expensive" resource from another agent only if that resource is useful to render all plans for achieving its goals executable.

Several sceptical semantics have been proposed for argumentation frameworks, notably the *grounded* semantics [7] and the semantics whereby beliefs held within all maximally admissible sets of arguments are drawn, referred to as the *sceptically preferred* semantics. The grounded semantics can be easily computed but is often overly sceptical. Procedures for the computation of the sceptically preferred semantic exist, e.g. the TPI procedure [21] for *coherent* argumentation frameworks [10], namely frameworks where all preferred sets of arguments are guaranteed to be *stable*, and the procedure of [5], for any argumentation framework, defined in non-dialectical terms. To the best of our knowledge, no dialectical procedure exists for checking whether a given belief can be deemed to hold under the sceptically preferred semantics for non-coherent cases.

In this paper we present two novel procedures for computing sceptical argumentation under the *ideal* semantics, originally proposed for extended logic programming in [1]. We adapt this semantics for abstract [7] and assumption-based [3] argumentation frameworks. The ideal semantics is sceptical, and has the advantage of being easily computable by a modification of the machinery presented in [8], but without being overly sceptical.

We define a procedure for the ideal semantics in abstract argumentation frameworks in terms of a form of *dispute trees* adapted from corresponding trees for computing the admissibility semantics in [8]. We prove that this procedure is sound and complete for all abstract argumentation frameworks. We define a procedure for the ideal semantics in assumption-based argumentation frameworks in terms of a form of *dispute derivations* adapted from corresponding derivations for computing the credulous admissibility semantics in [8]. These derivations use arguments which can be computed effectively by backward deductions in assumption-based frameworks. We prove that this procedure is sound for all assumption-based frameworks, and complete for a special class of assumption-based frameworks we define, with the property of being *p-acyclic*.

The paper is organised as follows. In Section 2 we review abstract and assumption-based argumentation frameworks, define the ideal semantics for both types of frameworks and relate it to other well-known semantics. We also provide a formal link between the two types of frameworks. In Section 3 we define dispute trees for abstract argumentation frameworks under the ideal semantics, by extending corresponding trees from the admissibility semantics proposed in [8]. In Section 4 we define dispute derivations for assumption-based argumentation under the ideal semantics, by extending corresponding derivations from the admissibility semantics proposed in [8]. We also provide two new variants of the derivations of [8], and prove that the first is sound for the sceptical grounded semantics, and the second is sound in general and complete for p-acyclic frameworks. In Section 5 we discuss some related work. Finally, in Section 6 we conclude.

This paper is an extended and improved version of the paper [9]: with respect to its predecessor, it presents a formal analysis of the ideal semantics also for abstract argumentation, a procedure (in terms of dispute trees) for the ideal semantics for abstract argumentation, a procedure in terms of dispute derivations for the grounded semantics, and contains proofs of all results.

2. Argumentation frameworks and semantics

In this section we briefly review the notions of abstract argumentation [7] and assumption-based argumentation [3,4,8,14,16], and present the ideal semantics for argumentation, adapted from [1].

2.1. Abstract argumentation

Definition 2.1. An *abstract argumentation framework* is a pair $(Arg, attacks)$ where Arg is a finite set, whose elements are referred to as *arguments*, and $attacks \subseteq Arg \times Arg$ is a binary relation over Arg. Given sets $X, Y \subseteq Arg$ of arguments, X *attacks* Y iff there exists $x \in X$ and $y \in Y$ such that $(x, y) \in attacks$.

Given an abstract argumentation framework, several notions of "acceptable" sets of arguments can be defined [3,7]. In this paper, we focus on the notions of admissible and ideal sets, defined below:

Definition 2.2. A set X of arguments is

- *admissible* iff X does not attack itself and X attacks every set of arguments Y such that Y attacks X;
- *preferred* iff X is maximally admissible;
- *complete* iff X is admissible and X contains all arguments x such that X attacks all attacks against x;
- *grounded* iff X is minimally complete;
- *ideal* iff X is admissible and it is contained in every preferred set of arguments.

Example 2.1. Consider the abstract framework $(Arg, attacks)$ where:

$$Arg = \{a, b, c, d\}$$
$$attacks = \{(a, a), (a, b), (b, a), (c, d), (d, c)\}$$

The *attacks* relation can be depicted as follows, where a pair (x, y) is represented by a directed arrow $x \rightarrow y$.

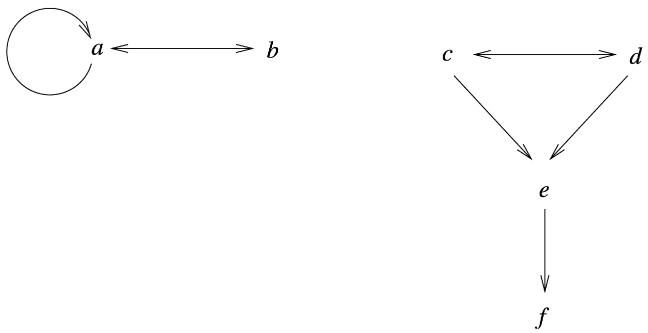
It is easy to see that:

- $\{\}$ is grounded;
- $\{b, d\}$ and $\{b, c\}$ are preferred;
- $\{b\}$ is the maximal ideal set.

Hence, the maximal ideal set is *less sceptical* than the grounded set. In the example, it coincides with the intersection of all preferred sets, but this does not hold in the general case. The following example shows that the maximal ideal set can be a proper subset of the intersection of all preferred sets. Consider the abstract framework $(Arg', attacks')$ which extends the above one as follows:

$$Arg' = Arg \cup \{e, f\}$$
$$attacks' = attacks \cup \{(c, e), (d, e), (e, f)\}$$

namely,



Then it is easy to see that:

- $\{b, d, f\}$ and $\{b, c, f\}$ are preferred;
- $\{b\}$ is the maximal ideal set, and $\{b\} \subset \{b, f\} = \{b, d, f\} \cap \{b, c, f\}$.

It is easy to see that the union of two ideal sets is still an ideal set.[1]

Lemma 2.1. *Let X and Y be two ideal sets of arguments. Then $X \cup Y$ is ideal.*

The following properties of the ideal semantics hold.

Theorem 2.1.

(i) *Every abstract argumentation framework admits a unique maximal ideal set of arguments.*
(ii) *The maximal ideal set of arguments is complete.*
(iii) *The maximal ideal set of arguments is a superset of the grounded set and is a subset of the intersection of all preferred sets.*
(iv) *If the intersection of all preferred sets of arguments is admissible, then it coincides with the maximal ideal set.*

Thus, the ideal semantics is sceptical, but less sceptical than the (overly sceptical) grounded semantics. It is, in general, more sceptical than the sceptical version of the preferred semantics, but, as we will prove in this paper, has the advantage of being easily computable by a modification of the machinery presented in [8].

2.2. Assumption-based argumentation frameworks

The abstract view of argumentation does not deal with the problem of actually finding arguments and attacks amongst them. Typically, arguments are built by connecting rules in the belief set of the proponent of arguments, and attacks arise from conflicts amongst such arguments. In assumption-based argumentation, arguments and attacks are not given as primitive. Instead, they are derived from the notions of inference rules, assumptions and contraries, as follows:

- arguments are obtained by reasoning backwards with a given set of inference rules (the "beliefs") from conclusions to assumptions, and
- attacks are defined in terms of a notion of "contrary" of assumptions.

Computationally, the use of assumption-based argumentation allows to exploit the fact that different arguments can share the same assumptions and thus avoid recomputation in many cases and the need to worry about sub-arguments of arguments.

Concretely, assumption-based argumentation frameworks are concrete instances of abstract argumentation frameworks where arguments in *Arg* are defined as *deductions* from *assumptions* in an underlying logic, viewed as a *deductive system*, and where *attacks* is defined in terms of a notion of *contrary*.

Definition 2.3. A *deductive system* is a pair $(\mathcal{L}, \mathcal{R})$ where

- $\mathcal{L}$ is a formal language consisting of countably many sentences, and
- $\mathcal{R}$ is a countable set of inference rules of the form
  $$\frac{\alpha_1, \ldots, \alpha_n}{\alpha}$$
  $\alpha \in \mathcal{L}$ is called the *conclusion* of the inference rule, $\alpha_1, \ldots, \alpha_n \in \mathcal{L}$ are called the *premises* of the inference rule and $n \geqslant 0$.

If $n = 0$, then the inference rule represents an axiom. A deductive system does not distinguish between domain-independent axioms/rules, which belong to the specification of the logic, and domain-dependent axioms/rules, which represent a background theory.

[1] All the proofs of the main results are given in Appendix A.

For notational convenience, we write $\alpha \leftarrow \alpha_1, \ldots, \alpha_n$ instead of $\frac{\alpha_1,\ldots,\alpha_n}{\alpha}$ and α instead of $\alpha \leftarrow$, throughout the paper.

Deductions can be understood as proof trees: the root of the tree is labelled by the conclusion of the deduction and the leaves are labelled by the premises supporting the deduction. For every non-terminal node in the tree, there is an inference rule whose conclusion matches the sentence labelling the node, and the children of the node are labelled by the premises of the inference rule. Following [8], we define deductions as sequences of frontiers $S_1, \ldots, S_m$ of the proof trees. Each frontier is represented by a multi-set, in which the same sentence can have several occurrences, if it is generated more than once as a premise of different inference steps.[2] In order to generate proof trees, a selection strategy is needed to identify which node to expand next. We formalise this selection strategy by means of a *selection function*, as in the formalisation of SLD resolution. A selection function, in this context, takes as input a sequence of multi-sets S_i and returns as output a sentence occurrence in S_i. We restrict the selection function so that if a sentence occurrence is selected in a multi-set in a sequence then it will not be selected again in any later multi-set in that sequence.

Definition 2.4. Given a selection function f, a (*backward*) *deduction* of a conclusion α based on (or supported by) a set of premises P is a sequence of multi-sets $S_1, \ldots, S_m$, where $S_1 = \{\alpha\}$, $S_m = P$, and for every $1 \leqslant i < m$, where σ is the sentence occurrence in S_i selected by f:

1. If σ is not in P then $S_{i+1} = S_i - \{\sigma\} \cup S$ for some inference rule of the form $\sigma \leftarrow S \in \mathcal{R}$.[3]
2. If σ is in P then $S_{i+1} = S_i$.

Each S_i is a step in the deduction.

Deductions are the basis for the construction of arguments in assumption-based argumentation frameworks, but to obtain an argument from a backward deduction we restrict the premises to ones that are acceptable as *assumptions*. Moreover, to specify when one argument attacks another, we need to determine when a sentence is the *contrary* of an assumption. Given a deductive system $(\mathcal{L}, \mathcal{R})$, these two notions—the notion of assumption and the notion of the contrary of an assumption—determine an assumption-based argumentation framework [8].

Definition 2.5. An *assumption-based argumentation framework* is a tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-} \rangle$ where

- $(\mathcal{L}, \mathcal{R})$ is a deductive system.
- $\mathcal{A} \subseteq \mathcal{L}$, $\mathcal{A} \neq \{\}$. $\mathcal{A}$ is the set of candidate *assumptions*.
- If $\alpha \in \mathcal{A}$, then there is no inference rule of the form $\alpha \leftarrow \alpha_1, \ldots, \alpha_n \in \mathcal{R}$.
- $^{-}$ is a (total) mapping from $\mathcal{A}$ into $\mathcal{L}$. $\bar{\alpha}$ is the *contrary* of α.

Note that, by the third bullet, following [8] we restrict ourselves to *flat* frameworks [3], whose assumptions do not occur as conclusions of inference rules.

Definition 2.6. An *argument* for a conclusion is a deduction of that conclusion whose premises are all assumptions (in $\mathcal{A}$).

Notation 2.1. In the remainder of this paper, we denote an argument for a conclusion α supported by a set of assumptions A simply as $A \vdash \alpha$.

Given an argument a of the form $A \vdash \alpha$ we say that a is *based upon* A.

The notation $A \vdash \alpha$ focuses attention on the set of assumptions A supporting an argument and its conclusion α. Instead, this notation ignores the internal structure of the argument, namely the inference rules used to generate it, as

[2] Multi-sets of sentences are equivalent to nodes labelled by sentences. The fact that a sentence can have several occurrences in a multi-set is equivalent to the fact that several nodes in a proof tree can be labelled by the same sentence.

[3] We use the same symbols for multi-set membership, union, intersection and subtraction as we use for ordinary sets.

well as the fact that there can be several distinct arguments that give rise to the same $A \vdash \alpha$ relationship. However, in our approach to argumentation, the set of assumptions supporting an argument and the conclusion of the argument encapsulate the essence of the argument, in that the only way to attack an argument is to attack one of its assumptions by supporting a conclusion that is the contrary of that assumption.

Definition 2.7.

- An *argument* $A \vdash \alpha$ *attacks an argument* $B \vdash \beta$ if and only if $A \vdash \alpha$ attacks an assumption in B;
- an *argument* $A \vdash \alpha$ *attacks an assumption* β if and only if α is the contrary $\overline{\beta}$ of β.

Let $ABF = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ be an assumption-based argumentation framework. Then, AF, the abstract framework *corresponding to ABF*, is $AF = (Arg, attacks)$ constructed as follows:

- each argument $a \in Arg$ is an argument $A \vdash \alpha$ as in Definition 2.6;
- $(a, b) \in attacks$ if and only if a attacks b as in Definition 2.7.

Notation 2.2. In the remainder of this paper, we write $AF \approx ABF$ whenever AF is the abstract framework corresponding to ABF. Similarly, we write $a \approx A \vdash \alpha$ whenever a is the argument in AF corresponding to the argument $A \vdash \alpha$ in ABF. Finally, given $a \approx A \vdash \alpha$ we will (improperly) refer to A as the set of assumptions *supporting a* and we will say that a is *supported by A*.

In the assumption-based approach to argumentation, the attack relationship between arguments depends solely on sets of assumptions. In some other approaches, however, such as that of Pollock [18] and Prakken and Sartor [19], an argument can attack another argument by contradicting its conclusion. [8,16] show how to reduce such "rebuttal" attacks to the "undermining" attacks of assumption-based argumentation frameworks.

Our focus on the assumptions of arguments motivates the following definition of attack, admissible and ideal semantics for assumption-based argumentation frameworks, where sets of arguments are replaced by sets of assumptions underlying arguments.

Definition 2.8.

- A *set of assumptions A attacks a set of assumptions B* if and only if there exists an argument $A' \vdash \bar{\alpha}$ such that $A' \subseteq A$ and $\alpha \in B$.
- A *set of assumptions A is admissible* if and only if A does not attack itself and A attacks every set of assumptions B that attacks A.
- A *set of assumptions A is preferred* if and only if it is maximally admissible.
- A *set of assumptions A is complete* if and only if it is admissible and contains all assumptions x such that A attacks all attacks against $\{x\}$.
- A *set of assumptions A is grounded* if and only if it is minimally complete.
- A *set of assumptions A is ideal* if and only if A is admissible and it is contained in every preferred set of assumptions.

Finally, we introduce the concepts of admissible and ideal belief, which will be useful in Section 4.

Definition 2.9. Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ be an assumption-based argumentation framework and let $\alpha \in \mathcal{L}$. Then α is an *admissible/grounded/ideal belief* if and only if there exists an argument $A \vdash \alpha$ such that A is a subset of an admissible/grounded/ideal set of assumptions.

The following theorem links the assumption-based approach to argumentation and the abstract approach, instantiated to assumption-based argumentation.

Theorem 2.2. *Let ABF be an assumption-based argumentation framework, and AF* $\approx$ *ABF.*

(i) *If a set of assumptions A is admissible/grounded/ideal wrt to ABF, then the union of all arguments supported by any subset of A is admissible/grounded/ideal wrt AF.*
(ii) *The union of all sets of assumptions supporting the arguments in an admissible/grounded/ideal set of arguments wrt AF is admissible/grounded/ideal wrt ABF.*

Example 2.2. Let *ABF* be an assumption-based framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ where

– $\mathcal{R}$ is the following set of rules:

$$\neg\alpha \leftarrow \alpha$$
$$\neg\alpha \leftarrow \beta$$
$$\neg\beta \leftarrow \alpha$$
$$\neg\gamma \leftarrow \delta$$
$$\neg\delta \leftarrow \gamma$$

– $\mathcal{A} = \{\alpha, \beta, \gamma, \delta\}$ and $\bar{\alpha} = \neg\alpha$, $\bar{\beta} = \neg\beta$, $\bar{\gamma} = \neg\gamma$, and $\bar{\delta} = \neg\delta$.

Some of the arguments in *ABF* are the following:

$\{\alpha\} \vdash \neg\alpha \qquad \{\beta\} \vdash \neg\alpha$

$\{\gamma\} \vdash \neg\delta \qquad \{\delta\} \vdash \neg\gamma$

It is worth noticing that, for instance, $\{\beta, \gamma\} \vdash \neg\alpha$ is *not* an argument, due to Definitions 2.4 and 2.6. Indeed, by Definition 2.4, there is no backward deduction for $\neg\alpha$ supported by $\{\beta, \gamma\}$, since γ is not in the premises of any inference rule for α or needed in a backward deduction of any premise of any such inference rule. So, by Definition 2.6, there exists no argument for $\neg\alpha$ based upon $\{\beta, \gamma\}$. It is also easy to see that:

- $\{\}$ is the grounded set of assumptions;
- $\{\beta, \delta\}$ and $\{\beta, \gamma\}$ are preferred sets of assumptions;
- $\{\beta\}$ is the maximal ideal set of assumptions.

3. Computing ideal sets of arguments for abstract argumentation

Ideal sets of arguments can be computed incrementally, in defence of a given, desired argument, by means of (special kinds of) *admissible dispute trees*, adapted from the dispute trees defined in [8] for computing admissible sets of arguments in the specific case of assumption-based argumentation frameworks. Below, we generalise the dispute trees of [8] to abstract argumentation frameworks, and then show how they can be used to compute admissible sets of arguments (Section 3.1, which is a direct adaptation of [8]) and to compute ideal sets of arguments (Section 3.2).

In general, dispute trees can be seen as a way of generating a winning strategy for a *proponent* to win a dispute against an *opponent*. The proponent starts by putting forward an initial argument, and then the proponent and the opponent alternate in attacking each other's previously presented arguments. The proponent wins if it has a counter-attack against every attacking argument by the opponent. Nodes in a dispute tree are labelled by arguments and are assigned the status of *proponent node* or *opponent node*, depending upon whether the argument at that node is put forward by the proponent or by the opponent. The root of the tree, at which the proponent puts forward an initial argument, is the starting point of the dispute. On every branch of a dispute tree, proponent and opponent alternate, but, for every proponent node, there is a (possibly empty) set of children, which are opponent nodes labelled by all the attacks against the proponent node, whereas, for every opponent node, there exists a single child, which is a proponent node, labelled by a single counter-attack against the opponent node.

Definition 3.1. A *dispute tree* for an initial argument a is a (possibly infinite) tree $\mathcal{T}$ such that

1. Every node of $\mathcal{T}$ is labelled by an argument and is assigned the status of *proponent* node or *opponent* node, but not both.

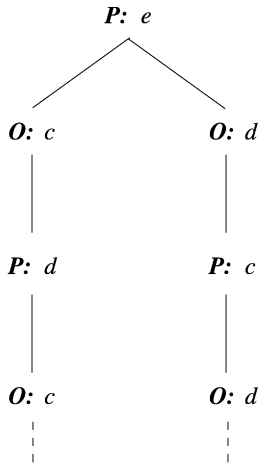


Fig. 1. A dispute tree labelled by e.

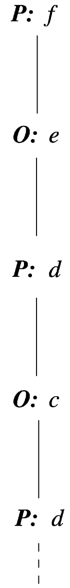


Fig. 2. A dispute tree labelled by f.

2. The root is a proponent node labelled by a.
3. For every proponent node N labelled by an argument b, and for every argument c that attacks b, there exists a child of N, which is an opponent node labelled by c.
4. For every opponent node N labelled by an argument b, there exists exactly one child of N which is a proponent node labelled by an argument which attacks b.
5. There are no other nodes in $\mathcal{T}$ except those given by 1–4 above.

The set of all arguments belonging to the proponent nodes in $\mathcal{T}$ is called the *defence set* of $\mathcal{T}$.

Note that, in 3 above, for every proponent node N labelled by an argument b, if there are no attacks against b, then N is a terminal node. Note also that a branch in a dispute tree may be finite or infinite. A finite branch represents a winning dispute that ends with an argument by the proponent that the opponent is unable to attack. An infinite branch represents a winning dispute in which the proponent counter-attacks every attack of the opponent, ad infinitum.

Example 3.1. Consider the abstract framework (Arg', $attacks'$) of Example 2.1. Fig. 1 shows the (infinite) dispute tree with root labelled by e (a node is represented by **X**: y where y is the argument labelling the node and **X** is either **P** or **O** representing the status of the node). The defence set of this tree is $\{e, c, d\}$. In Fig. 2 we show an (infinite) dispute tree with root labelled by f. The defence set of this tree is $\{f, d\}$.

Note that we could obtain finite trees from the (possibly infinite) dispute trees we define here by using some filtering mechanisms so that one does not have to re-defend what has already been defended before in a tree. As the focus of this paper is on developing computational tools for assumption-based argumentation, we will develop such mechanisms in Section 4.

3.1. Computing admissible sets of arguments

The definition of dispute tree incorporates the requirement that the proponent must counter-attack every attack, but it does not incorporate the further requirement that the proponent does not attack itself. This further requirement is incorporated in the definition of *admissible* dispute tree:

Definition 3.2. A dispute tree $\mathcal{T}$ is *admissible* if and only if no argument labels both a proponent and an opponent node.

Going back to Example 3.1, the dispute tree of Fig. 1 is not admissible, whereas the dispute tree of Fig. 2 is admissible.

A proof procedure that searches for admissible dispute trees does not always need to incorporate an explicit admissibility check. For example, finite dispute trees are guaranteed to be admissible even without such a check, as shown by the following theorem which is analogous to Theorem 5.2 of [8]. Indeed, the proof of the next theorem is an adaptation of the proof in [8].

Theorem 3.1. *Any dispute tree that has no infinitely long branches is an admissible dispute tree.*

Although unnecessary, the admissibility check of Definition 3.2 can be useful for efficiency reasons, since it can decrease the size of dispute trees. The next theorem is analogous to Theorem 5.1 of [8].

Theorem 3.2.

(i) *If $\mathcal{T}$ is an admissible dispute tree for an argument a then the defence set of $\mathcal{T}$ is admissible.*
(ii) *If a is an argument and $a \in A$ where A is an admissible set of arguments then there exists an admissible dispute tree for a with defence set A' such that $A' \subseteq A$ and A' is admissible.*

Admissible dispute trees show how to extend an initial argument incrementally to an admissible set of arguments. However, these trees are non-constructive, because they can be infinite, as shown in Example 3.1.

3.2. Computing ideal sets of arguments

The following theorem provides the backbone for the notion of ideal dispute tree below.

Theorem 3.3. *An admissible set of arguments S is ideal iff for each argument a attacking S there exists no admissible set of arguments containing a.*

Thus, in order to support the computation of the ideal semantics for abstract argumentation frameworks, the definition of admissible dispute tree needs to be extended in order to incorporate the additional requirement indicated by the earlier theorem.

Definition 3.3. An admissible dispute tree $\mathcal{T}$ is *ideal* if and only if for no opponent node O in $\mathcal{T}$ there exists an admissible tree with root O.

Example 3.2. Consider the abstract framework in Example 2.1. Fig. 3 shows an ideal dispute tree for b. This tree is ideal since (1) it is admissible, and (2) there exists no admissible tree with root a (since a attacks itself).

Theorem 3.4.

(i) *If $\mathcal{T}$ is an ideal dispute tree for an argument a then the defence set of $\mathcal{T}$ is ideal.*
(ii) *If a is an argument and $a \in A$ where A is an ideal set of arguments then there exists an ideal dispute tree for a with defence set A' and $A' \subseteq A$.*

Ideal dispute trees shorten the distance between the definition of ideal set of arguments and a concrete proof procedure for computing this set, because they show how to extend an initial argument incrementally to an ideal set of arguments. However, they are still non-constructive, for the same reasons that admissible dispute trees are non-constructive in the context of computing admissible sets of arguments. Indeed, in general (admissible and ideal), dispute trees may be infinite. One could fold them into finite trees by using some filtering mechanisms so that one

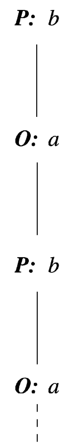


Fig. 3. An ideal dispute tree.

does not have to re-defend what has already been defended earlier on. We will indeed do this in the next section, for assumption-based frameworks where the notion of argument is not given as primitive. There, we will also show how to integrate the tasks of computing arguments with the task of defending them, and we will enforce some additional filtering mechanisms to avoid re-attacking attacks by the opponent, to obtain a more efficient procedure.

4. Computing ideal beliefs for assumption-based argumentation

An (admissible or ideal) dispute tree is an *abstraction* of a winning strategy for a dispute, because it does not show the construction of arguments and counter-arguments. The dispute derivations of [8] construct admissible dispute trees while constructing arguments for assumption-based argumentation frameworks. Our proof procedure for computing *I*deal *B*eliefs for assumption-based frameworks is defined in terms of *IB-dispute derivations*, adapted from (a variant of) the *dispute derivations* of [8] for computing *A*dmissible *B*eliefs, that we refer to here as *AB-dispute derivations*. Below, we first review AB-dispute derivations (Section 4.1), and then define IB-dispute derivations (Section 4.2). IB- and AB-dispute derivations are sequences of tuples corresponding to frontiers of dispute trees that are being constructed top–down, using backward reasoning to generate arguments, interleaving the construction of arguments, constructing and defeating attacks against these arguments, and checking that the generated tree is admissible (for AB-dispute derivations) or ideal (for IB-dispute derivations). During the construction of (admissible or ideal) dispute trees by means of (the appropriate kind of) dispute derivations, arguments do not need to be fully computed before being defended against attacks (if the arguments are proposed by the proponent) or defeated by counter-attacks (if the arguments are proposed by the opponent). Indeed, as soon as an assumption is encountered in the construction of an argument (by either the proponent or opponent), that assumption may be attacked by the adversary in the dispute. This has the advantage that failure can be detected sooner. However, this implies that arguments being constructed are only *potential arguments*, namely deductions $S \vdash \alpha$ whose premises S may or may not be assumptions. Potential arguments correspond to the backward construction of arguments: each potential argument may result in one, no or multiple arguments.

Example 4.1. Consider the assumption-based framework where $\mathcal{R}$ is

$$p \leftarrow q, \alpha$$

where α is an assumption. Then, $\{q, \alpha\} \vdash p$ is a potential argument, that will result (by backward deduction) in no actual argument. If however $\mathcal{R}$ contains also rules

$$q \leftarrow \beta$$
$$q \leftarrow \gamma$$

with β and γ assumptions, then the potential argument will result in two actual arguments, $\{\alpha, \beta\} \vdash p$ and $\{\alpha, \gamma\} \vdash p$.

By virtue of relying upon potential arguments, dispute derivations thus actually construct *concrete dispute trees* [8], that may correspond to one, no or multiple dispute trees.

Note that all our dispute derivations will be of finite length. This is because our ultimate goal is to develop effective proof procedures that can be used to support practical applications.

4.1. Computing admissible beliefs

The efficient construction of admissible dispute trees for assumption-based argumentation frameworks can be obtained via *AB-dispute derivations*, given in Section 4.1.2 below. These are a variant of the dispute derivations of [8], that improve upon them by being "more complete" (as we will see below). But first, in Section 4.1.1 we present a preliminary form of AB-dispute derivations, referred to as GB-dispute derivations as they compute *G*rounded *B*eliefs. GB-dispute derivations can be seen as an initial step between dispute trees and the actual, final form of AB-dispute derivations, in that they are correct, but highly incomplete and inefficient. Dispute derivations rely upon the re-interpretation of the notion of admissible dispute trees in terms of the following definitions, for assumption-based argumentation frameworks.

Definition 4.1. Let ABF be an assumption-based argumentation framework, and $AF \approx ABF$. Given a dispute tree $\mathcal{T}$ for AF:

- for any opponent node labelled by an argument b with child a proponent node labelled by an argument a if a attacks some assumption α in the set supporting b then α is said to be the *culprit* in b;
- the set of all assumptions supporting the arguments in the defence set of $\mathcal{T}$ is referred to as the *assumption-defence set* of $\mathcal{T}$.

The following theorem trivially holds:

Theorem 4.1. *Let ABF be an assumption-based argumentation framework, and $AF \approx ABF$. Given a dispute tree $\mathcal{T}$ for AF, $\mathcal{T}$ is admissible if and only if no culprit in the argument of an opponent node belongs to the assumption-defence set of $\mathcal{T}$.*

The following is a direct corollary of Theorem 3.2, and its proof is the same as that of Theorem 5.1 in [8].

Corollary 4.1. *Given a dispute tree $\mathcal{T}$ for an assumption-based argumentation framework*:

(i) *If $\mathcal{T}$ is an admissible dispute tree for an argument a then the assumption-defence set of $\mathcal{T}$ is an admissible set of assumptions.*
(ii) *If a is an argument supported by a set of assumptions A_0 and A is an admissible set of assumptions such that $A_0 \subseteq A$, then there exists an admissible dispute tree for a with assumption-defence set A' such that $A_0 \subseteq A' \subseteq A$ and A' is admissible.*

4.1.1. GB-dispute derivations

GB-dispute derivations are sequences of quadruples: $\langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i \rangle$, where $\mathcal{P}_i$ and $\mathcal{O}_i$ represent the (proponent and opponent) nodes in the frontier of the part of the tree generated at step i, together with the set of defence assumptions A_i and culprits C_i generated so far. $\mathcal{O}_i$ is a multi-set corresponding directly to the opponent nodes in the frontier—i.e. its members are multi-sets of sentences representing the state of all of the opponent's *potential arguments* against the proponent. The multi-set $\mathcal{P}_i$ is a flattened version of the proponent's potential arguments—i.e. its members are occurrences of sentences belonging to any of the proponent's potential arguments.

The first step of a dispute derivation represents the root of the dispute tree. Each step in a dispute derivation represents the selection of a node in the frontier of the dispute tree and its replacement by its children. Any node in the frontier can be selected for this purpose. Different selections give rise to different derivations, but do not affect completeness, because they simply represent different ways of generating the same dispute tree.

In GB-dispute derivations:

- the set of culprits C_i is used to filter potential defence arguments (step 1(ii)), in that potential defence arguments whose intersection with the set of culprits C_i is non-empty are disregarded;
- the set of defence assumptions A_i is used to filter potential culprits (step 2(i)(b)), in that a potential culprit is disregarded if it has already been chosen as a defence assumption,

so that the final assumption-defence set constructed by the derivation does not attack itself (Theorem 4.1).

Definition 4.2. Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ be an assumption based framework. Given a selection function, a *GB-dispute derivation of a defence set A* for a sentence α is a finite sequence of quadruples

$$\langle \mathcal{P}_0, \mathcal{O}_0, A_0, C_0 \rangle, \ldots, \langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i \rangle, \ldots, \langle \mathcal{P}_n, \mathcal{O}_n, A_n, C_n \rangle$$

where

$$\mathcal{P}_0 = \{\alpha\} \qquad A_0 = \mathcal{A} \cap \{\alpha\} \qquad \mathcal{O}_0 = C_0 = \{\}$$
$$\mathcal{P}_n = \mathcal{O}_n = \{\} \qquad A = A_n$$

and for every $0 \leqslant i < n$, only one σ in $\mathcal{P}_i$ or one S in $\mathcal{O}_i$ is selected, and:

1. If $\sigma \in \mathcal{P}_i$ is selected then
   (i) if σ is an assumption, then
   $$\mathcal{P}_{i+1} = \mathcal{P}_i - \{\sigma\} \qquad A_{i+1} = A_i \qquad C_{i+1} = C_i \qquad \mathcal{O}_{i+1} = \mathcal{O}_i \cup \{\{\bar{\sigma}\}\}$$
   (ii) if σ is not an assumption, then there exists some inference rule $\sigma \leftarrow R \in \mathcal{R}$ such that $C_i \cap R = \{\}$ (*filtering of potential defence arguments by culprits*) and
   $$\mathcal{P}_{i+1} = \mathcal{P}_i - \{\sigma\} \cup R \qquad A_{i+1} = A_i \cup (\mathcal{A} \cap R)$$
   $$C_{i+1} = C_i \qquad \mathcal{O}_{i+1} = \mathcal{O}_i$$
2. If S is selected in $\mathcal{O}_i$ and σ is selected in S then
   (i) if σ is an assumption, then
   (a) either σ is ignored, i.e.
   $$\mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} \cup \{S - \{\sigma\}\} \qquad \mathcal{P}_{i+1} = \mathcal{P}_i$$
   $$A_{i+1} = A_i \qquad C_{i+1} = C_i$$
   (b) or $\sigma \notin A_i$ (*filtering of culprits by defence assumptions*) and
   $$\mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} \qquad \mathcal{P}_{i+1} = \mathcal{P}_i \cup \{\bar{\sigma}\}$$
   $$A_{i+1} = A_i \cup (\{\bar{\sigma}\} \cap \mathcal{A}) \qquad C_{i+1} = C_i \cup \{\sigma\}$$
   (ii) if σ is not an assumption, then
   $$\mathcal{P}_{i+1} = \mathcal{P}_i \qquad A_{i+1} = A_i \qquad C_{i+1} = C_i$$
   $$\mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} \cup \big\{ S - \{\sigma\} \cup R \;\big|\; \sigma \leftarrow R \in \mathcal{R} \big\}$$

Note that step 2(i)(a) is not needed to guarantee soundness, but is helpful to guarantee success in finding GB-dispute derivations in many cases, as illustrated in Example 7.2 in [8].

GB-dispute derivations compute support sets of grounded beliefs, as follows:

Theorem 4.2. *Given a GB-dispute derivation of a defence set A for a sentence* α:

- *A is admissible and it is contained in the grounded set of assumptions*;
- *there exists* $A' \subseteq A$ *and an argument* $A' \vdash \alpha$.

Note that GB-dispute derivations succeed in many cases where other procedures for computing grounded beliefs fail. However, GB-dispute derivations are "highly incomplete" for admissibility, in that they fail to compute admissible sets in many cases, corresponding to infinite dispute trees.

Example 4.2. Let an assumption-based framework have $\mathcal{R}$ with

$$\neg\alpha \leftarrow \beta$$
$$\neg\beta \leftarrow \alpha$$

$\mathcal{A} = \{\alpha, \beta\}$ and $\bar{\alpha} = \neg\alpha$, $\bar{\beta} = \neg\beta$. Then, there exists no GB-dispute derivation for $\neg\alpha$, as shown by the failed search for one such derivation below:

i	$\mathcal{P}_i$	$\mathcal{O}_i$	A_i	C_i	
0	$\{\neg\alpha\}$	$\{\}$	$\{\}$	$\{\}$	
1	$\{\beta\}$	$\{\}$	$\{\beta\}$	$\{\}$	by 1.ii
2	$\{\}$	$\{\{\neg\beta\}\}$	$\{\beta\}$	$\{\}$	by 1.i
3	$\{\}$	$\{\{\alpha\}\}$	$\{\beta\}$	$\{\}$	by 2.ii
4	$\{\neg\alpha\}$	$\{\}$	$\{\beta\}$	$\{\alpha\}$	by 2.i.b
5	...				

However, there exists an infinite admissible dispute tree for $\neg\alpha$, whose defence set is $\{\beta\}$.

Moreover, the given notion of GB-derivation can be inefficient, as illustrated by the following example.

Example 4.3. Let an assumption-based framework have $\mathcal{R}$ with

$$p \leftarrow \alpha$$
$$\neg\alpha \leftarrow \beta$$
$$\neg\alpha \leftarrow r$$
$$r \leftarrow \beta$$
$$\neg\beta$$

$\mathcal{A} = \{\alpha, \beta\}$ and $\bar{\alpha} = \neg\alpha$, $\bar{\beta} = \neg\beta$. Then, a GB-dispute derivation for p is shown below:

i	$\mathcal{P}_i$	$\mathcal{O}_i$	A_i	C_i	
0	$\{p\}$	$\{\}$	$\{\}$	$\{\}$	
1	$\{\alpha\}$	$\{\}$	$\{\alpha\}$	$\{\}$	by 1.ii
2	$\{\}$	$\{\{\neg\alpha\}\}$	$\{\alpha\}$	$\{\}$	by 1.i
3	$\{\}$	$\{\{\beta\}, \{r\}\}$	$\{\alpha\}$	$\{\}$	by 2.ii
4	$\{\neg\beta\}$	$\{\{r\}\}$	$\{\alpha\}$	$\{\beta\}$	by 2.i.b
5	$\{\}$	$\{\{r\}\}$	$\{\alpha\}$	$\{\beta\}$	by 1.ii
6	$\{\}$	$\{\{\beta\}\}$	$\{\alpha\}$	$\{\beta\}$	by 2.ii
7	$\{\neg\beta\}$	$\{\}$	$\{\alpha\}$	$\{\beta, \beta\}$	by 2.i.b
8	$\{\}$	$\{\}$	$\{\alpha\}$	$\{\beta, \beta\}$	by 1.ii

Obviously, steps 7–8 are wasteful, as the culprit β has already been defeated.

4.1.2. AB-dispute derivations

These new dispute derivations incorporate a filtering by defence assumptions so that they can (finitely) compute infinite admissible dispute trees, in examples such as Example 4.2 above. Moreover, AB-dispute derivations incorporate a filtering by culprit assumptions so that they can be more efficient, in examples such as Example 4.3 above. Concretely, the set of defence assumptions A_i is used both to filter proponent assumptions in $\mathcal{P}_i$, so they are not considered redundantly more than once, and to filter potential culprit assumptions in $\mathcal{O}_i$, so that the final defence set A constructed by the derivation does not attack itself. The set of culprits C_i is similarly used both to filter potential culprit assumptions in $\mathcal{O}_i$, so they are not counter-attacked redundantly more than once, and to filter proponent assumptions in $\mathcal{P}_i$, so that A does not attack itself.

Definition 4.3. Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ be an assumption based framework. Given a selection function, an *AB-dispute derivation of a defence set* A for a sentence α is a finite sequence of quadruples

$$\langle \mathcal{P}_0, \mathcal{O}_0, A_0, C_0 \rangle, \ldots, \langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i \rangle, \ldots, \langle \mathcal{P}_n, \mathcal{O}_n, A_n, C_n \rangle$$

where

$$\begin{array}{lll} \mathcal{P}_0 = \{\alpha\} & A_0 = \mathcal{A} \cap \{\alpha\} & \mathcal{O}_0 = C_0 = \{\} \\ \mathcal{P}_n = \mathcal{O}_n = \{\} & A = A_n & \end{array}$$

and for every $0 \leqslant i < n$, only one σ in $\mathcal{P}_i$ or one S in $\mathcal{O}_i$ is selected, and:

1. If $\sigma \in \mathcal{P}_i$ is selected then
   (i) if σ is an assumption, then

$$\mathcal{P}_{i+1} = \mathcal{P}_i - \{\sigma\} \qquad A_{i+1} = A_i \qquad C_{i+1} = C_i \qquad \mathcal{O}_{i+1} = \mathcal{O}_i \cup \{\{\bar{\sigma}\}\}$$

   (ii) if σ is not an assumption, then there exists some inference rule $\sigma \leftarrow R \in \mathcal{R}$ such that $C_i \cap R = \{\}$ and

$$\mathcal{P}_{i+1} = \mathcal{P}_i - \{\sigma\} \cup (R - A_i) \quad (\textit{filtering of defence assumptions by defences})$$
$$A_{i+1} = A_i \cup (\mathcal{A} \cap R)$$
$$C_{i+1} = C_i \qquad \mathcal{O}_{i+1} = \mathcal{O}_i$$

2. If S is selected in $\mathcal{O}_i$ and σ is selected in S then
   (i) if σ is an assumption, then
      (a) either σ is ignored, i.e.

$$\begin{array}{ll} \mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} \cup \{S - \{\sigma\}\} & \mathcal{P}_{i+1} = \mathcal{P}_i \\ A_{i+1} = A_i & C_{i+1} = C_i \end{array}$$

      (b) or $\sigma \notin A_i$ and $\sigma \in C_i$ (*filtering of culprits by culprits*)[4] and

$$\mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} \qquad \mathcal{P}_{i+1} = \mathcal{P}_i \qquad A_{i+1} = A_i \qquad C_{i+1} = C_i$$

      (c) or $\sigma \notin A_i$ and $\sigma \notin C_i$ (*filtering of culprits by culprits*)[5] and
         (c.1) if $\bar{\sigma}$ is not an assumption, then

$$\begin{array}{ll} \mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} & \mathcal{P}_{i+1} = \mathcal{P}_i \cup \{\bar{\sigma}\} \\ A_{i+1} = A_i & C_{i+1} = C_i \cup \{\sigma\} \end{array}$$

         (c.2) if $\bar{\sigma}$ is an assumption, then

$$\mathcal{O}_{i+1} = \mathcal{O}_i - \{S\}$$
$$\mathcal{P}_{i+1} = \mathcal{P}_i \quad (\textit{filtering of defence assumptions by defences})$$
$$A_{i+1} = A_i \cup \{\bar{\sigma}\} \qquad C_{i+1} = C_i \cup \{\sigma\}$$

   (ii) if σ is not an assumption, then

$$\mathcal{P}_{i+1} = \mathcal{P}_i \qquad A_{i+1} = A_i \qquad C_{i+1} = C_i$$
$$\mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} \cup \big\{S - \{\sigma\} \cup R \;\big|\; \sigma \leftarrow R \in \mathcal{R}, \text{ and } R \cap C_i = \{\}\big\}$$
$$(\textit{filtering of culprits by culprits})$$

[4] In [8], this case (b) was missing. Our new case here provides an additional filtering of culprits by culprits without affecting the correctness of the procedure.

[5] In [8], the condition $\sigma \notin C_i$ in case (c) and case (c.2) were missing. Our new case here provides an additional filtering of culprits by culprits without affecting the correctness of the procedure. Moreover, case (c.2) takes into account the situation in which the contrary of the chosen culprit is an assumption in turn. Notice that case (c.2) assumes that if the contrary of α is β then the contrary of β is α, as in all the frameworks we use in this paper. If this is not the case, case (c.2) can be simply disregarded.

Example 4.4. Consider the assumption-based framework in Example 4.2. An AB-dispute derivation for $\neg\alpha$ is

i	$\mathcal{P}_i$	$\mathcal{O}_i$	A_i	C_i	
0	$\{\neg\alpha\}$	$\{\}$	$\{\}$	$\{\}$	
1	$\{\beta\}$	$\{\}$	$\{\beta\}$	$\{\}$	by 1.ii
2	$\{\}$	$\{\{\neg\beta\}\}$	$\{\beta\}$	$\{\}$	by 1.i
3	$\{\}$	$\{\{\alpha\}\}$	$\{\beta\}$	$\{\}$	by 2.ii
4	$\{\neg\alpha\}$	$\{\}$	$\{\beta\}$	$\{\alpha\}$	by 2.i.c.1
5	$\{\}$	$\{\}$	$\{\beta\}$	$\{\alpha\}$	by 1.ii

Indeed, $\neg\alpha$ is an admissible belief, and $\{\beta\} \vdash \neg\alpha$ is an admissible argument.

AB-derivations are guaranteed to compute admissible beliefs, as follows:

Theorem 4.3. *For every AB-dispute derivation of a defence set A for a sentence α, the defence set A is admissible, and there exists some $A' \subseteq A$ that supports an argument for α.*

As discussed in [8], AB-dispute derivations are not complete in general. In this paper, we give a sufficient condition for their completeness, thus providing a sufficient condition for the soundness of IB-dispute derivations defined later on. For simplicity, we will prove this result for the simplified assumption-based frameworks used throughout the paper for the examples. These frameworks fulfil the following requirements:

- All sentences in $\mathcal{L}$ are atoms $p, q, \ldots, \alpha, \beta, \ldots$ or negations of atoms $\neg p, \neg q, \ldots, \neg\alpha, \neg\beta, \ldots$ (i.e. $\mathcal{L}$ is a set of *literals*).
- The set of assumptions $\mathcal{A}$ is a subset of the set of all literals that do not occur as the conclusion of any inference rule in $\mathcal{R}$.
- The contrary of any assumption α is $\neg\alpha$; the contrary of any assumption $\neg\alpha$ is α.

Notation 4.1. Let *ABF* be an assumption-based framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, {}^{-} \rangle$. By ABF^+, we will denote the framework obtained by deleting all assumptions appearing in the premises of the inference rules of $\mathcal{R}$.

Below, given *ABF*, we use the notion of *dependency graph* of ABF^+, defined in a way similar to the *atom dependency graph* for logic programming (see, e.g., the review in [2]). The dependency graph of ABF^+ is a directed graph where:

- the nodes are the atoms occurring in ABF^+;
- a (directed) arc from a node p to a node q is in the graph if and only if there exists a rule $p \leftarrow B$ in ABF^+ such that q occurs in B.

Definition 4.4. An assumption-based framework *ABF* is *positively acyclic* (or *p-acyclic* for short) if the dependency graph of ABF^+ is acyclic.

It is easy to see that the following lemma holds:

Lemma 4.1. *Given an assumption based framework, let an* infinite partial *deduction be an infinite sequence of steps defined as in Definition* 2.4.

Given a p-acyclic framework, there exists no infinite partial deduction.

Note that non-p-acyclic frameworks are rarely encountered in practice, and all assumption-based frameworks we have used in this paper for illustration purposes are p-acyclic. Note that p-acyclic frameworks are not guaranteed to be *coherent* [7]. For example, the assumption-based framework with $\mathcal{R} = \{\neg\alpha \leftarrow \alpha\}$, $\mathcal{A} = \{\alpha\}$ and $\bar{\alpha} = \neg\alpha$ is p-acyclic but not coherent. Moreover, coherent frameworks are not guaranteed to be p-acyclic. For example, the assumption-based framework with $\mathcal{R} = \{p \leftarrow p\}$ and $\mathcal{A} = \{\alpha\}$ is coherent (it admits a single preferred and stable set of assumptions $\{\alpha\}$) but not p-acyclic.

In the case of p-acyclic frameworks with a finite underlying language $\mathcal{L}$ the dispute derivations of Definition 4.3 are complete, in the following sense:

Theorem 4.4. *Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, {}^{-} \rangle$ be a p-acyclic assumption-based framework such that $\mathcal{L}$ is finite. Then, for each literal α, if α is an admissible belief then*

- *there exists a dispute derivation for α;*
- *for each admissible set of assumptions Δ, if Δ supports an argument for α then there is a dispute derivation of defence set A for α such that $A \subseteq \Delta$ and a subset of A supports an argument for α.*

4.2. Computing ideal beliefs

In this section we define *IB-dispute derivations*, adapted from AB-dispute derivations. Like AB-dispute derivations, IB-dispute derivations are sequences of tuples, but these tuples are of the form $\langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i, \mathcal{F}_i \rangle$. $\mathcal{F}_i$ is a novel component intuitively holding all potential attacks against the proponent: by virtue of Theorem 3.3 IB-dispute derivations need to make sure that no admissible set of assumptions containing any attack in $\mathcal{F}_i$ exists. This check is ultimately performed by a new kind of (subsidiary) dispute derivations, that we call *Fail-dispute derivations*. In Section 4.2.1 below, we give IB-dispute derivations, relying upon a high-level notion of *Fail* instead of Fail-dispute derivations. In Section 4.2.2 we define Fail-dispute derivations computing *Fail*.

Before we define these new kinds of dispute derivation let us introduce few notations. The notion of dispute derivation in Definition 4.3 can be extended to a set of sentences S instead of just a single sentence α, by setting $\mathcal{P}_0$ to S. Then:

Notation 4.2.

- Given a set of assumptions S, we write $S \Vdash \alpha$ if there exists an argument $A \vdash \alpha$ such that $A \subseteq S$. Moreover, given a set of sentences P, we write $S \Vdash P$ meaning $S \Vdash \alpha$, for each $\alpha \in P$.
- Let P be a set of sentences in $\mathcal{L}$. By *Fail*(P) we mean that there exists no admissible set E of assumptions such that $E \Vdash P$.

IB-dispute derivations are sequences of tuples of the form $\langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i, \mathcal{F}_i \rangle$, where

- the new component $\mathcal{F}_i$ holds all multisets S for which we want to prove that *Fail*(S) (these are the potential attacks S dealt with in step 2. of AB-dispute derivation).
- $\mathcal{P}_i, \mathcal{O}_i, A_i, C_i$ are as in AB-dispute derivations, except that sentences occurring in the multisets in $\mathcal{O}_i$ may be *marked*.

Notation 4.3. Given a multiset of sentences S:

- S_u is the multiset of *unmarked* sentences in S;
- $m(\sigma, S)$ is the multiset S where $\sigma \in S$ *becomes marked*;
- $u(S)$ is S where the marked sentences are *unmarked*.

Intuitively, IB-dispute derivations compute an admissible support for the given sentence α while trying to check that no admissible set attacks it. As soon as a (potential) attack is found, this is stored in the $\mathcal{F}$ component of the tuple to check that this fails to be/become admissible. Whenever a potential culprit is ignored in a potential attack, this is marked so that it will not be selected again. Selected elements in the potential attacks in the $\mathcal{O}$ component are chosen amongst the unmarked elements. Thus, we will impose that, given a multiset S in $\mathcal{O}_i$, the selection function will only select unmarked sentences in S_u.

4.2.1. IB-dispute derivations

Definition 4.5. Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-} \rangle$ be an assumption based framework. Given a selection function, an *IB-dispute derivation of an ideal support* A for a sentence α is a finite sequence of tuples

$$\langle \mathcal{P}_0, \mathcal{O}_0, A_0, C_0, \mathcal{F}_0 \rangle, \ldots, \langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i, \mathcal{F}_i \rangle, \ldots, \langle \mathcal{P}_n, \mathcal{O}_n, A_n, C_n, \mathcal{F}_n \rangle$$

where

$$\mathcal{P}_0 = \{\alpha\} \qquad A_0 = \mathcal{A} \cap \mathcal{P}_0 \qquad \mathcal{O}_0 = C_0 = \mathcal{F}_0 = \{\}$$
$$\mathcal{P}_n = \mathcal{O}_n = \mathcal{F}_n = \{\} \qquad A = A_n$$

and for every $0 \leqslant i < n$, only one σ in $\mathcal{P}_i$ or one S in $\mathcal{O}_i$ or one S in $\mathcal{F}_i$ is selected, and:

1. If $\sigma \in \mathcal{P}_i$ is selected then
   (i) if σ is an assumption, then

$$\mathcal{P}_{i+1} = \mathcal{P}_i - \{\sigma\} \qquad A_{i+1} = A_i \qquad C_{i+1} = C_i$$
$$\mathcal{O}_{i+1} = \mathcal{O}_i \cup \{\{\overline{\sigma}\}\} \qquad \mathcal{F}_{i+1} = \mathcal{F}_i$$

   (ii) if σ is not an assumption, then there exists some inference rule $\sigma \leftarrow R \in \mathcal{R}$ such that $C_i \cap R = \{\}$ and

$$\mathcal{P}_{i+1} = \mathcal{P}_i - \{\sigma\} \cup (R - A_i) \qquad A_{i+1} = A_i \cup (\mathcal{A} \cap R) \qquad C_{i+1} = C_i$$
$$\mathcal{O}_{i+1} = \mathcal{O}_i \qquad \mathcal{F}_{i+1} = \mathcal{F}_i$$

2. If S is selected in $\mathcal{O}_i$ then σ is selected in S_u and
   (i) if σ is an assumption, then
      (a) either σ is ignored, i.e.

$$\mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} \cup \{m(\sigma, S)\} \qquad \mathcal{P}_{i+1} = \mathcal{P}_i \qquad A_{i+1} = A_i$$
$$C_{i+1} = C_i \qquad \mathcal{F}_{i+1} = \mathcal{F}_i$$

      (b) or $\sigma \notin A_i$ and $\sigma \in C_i$ and

$$\mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} \qquad \mathcal{P}_{i+1} = \mathcal{P}_i \qquad A_{i+1} = A_i$$
$$C_{i+1} = C_i \qquad \mathcal{F}_{i+1} = \mathcal{F}_i \cup \{u(S)\}$$

      (c) or $\sigma \notin A_i$ and $\sigma \notin C_i$ and
         (c.1) if $\overline{\sigma}$ is not an assumption, then

$$\mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} \qquad \mathcal{P}_{i+1} = \mathcal{P}_i \cup \{\overline{\sigma}\} \qquad A_{i+1} = A_i$$
$$C_{i+1} = C_i \cup \{\sigma\} \qquad \mathcal{F}_{i+1} = \mathcal{F}_i \cup \{u(S)\}$$

         (c.2) if $\overline{\sigma}$ is an assumption, then

$$\mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} \qquad \mathcal{P}_{i+1} = \mathcal{P}_i \qquad A_{i+1} = A_i \cup \{\overline{\sigma}\}$$
$$C_{i+1} = C_i \cup \{\sigma\} \qquad \mathcal{F}_{i+1} = \mathcal{F}_i \cup \{u(S)\}$$

   (ii) if σ is not an assumption, then

$$\mathcal{P}_{i+1} = \mathcal{P}_i \qquad A_{i+1} = A_i \qquad C_{i+1} = C_i$$

$$\mathcal{F}_{i+1} = \mathcal{F}_i \cup \{S - \{\sigma\} \cup R \mid \sigma \leftarrow R \in \mathcal{R} \text{ and } R \cap C_i \neq \{\}\}$$

$$\mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} \cup \{S - \{\sigma\} \cup R \mid \sigma \leftarrow R \in \mathcal{R} \text{ and } R \cap C_i = \{\}\}$$

3. If S is selected in $\mathcal{F}_i$ then *Fail*(S) and

$$\mathcal{O}_{i+1} = \mathcal{O}_i \qquad \mathcal{P}_{i+1} = \mathcal{P}_i \qquad A_{i+1} = A_i$$
$$C_{i+1} = C_i \qquad \mathcal{F}_{i+1} = \mathcal{F}_i - \{S\}$$

Example 4.5. Consider the assumption-based framework in Example 2.2. An IB-dispute derivation for $\neg\alpha$ is $\langle\mathcal{P}_0, \mathcal{O}_0, A_0, C_0, \mathcal{F}_0\rangle, \ldots, \langle\mathcal{P}_6, \mathcal{O}_6, A_6, C_6, \mathcal{F}_6\rangle$ where

i	$\mathcal{P}_i$	$\mathcal{O}_i$	A_i	C_i	$\mathcal{F}_i$	
0	$\{\neg\alpha\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	
1	$\{\beta\}$	$\{\}$	$\{\beta\}$	$\{\}$	$\{\}$	by 1.ii
2	$\{\}$	$\{\{\neg\beta\}\}$	$\{\beta\}$	$\{\}$	$\{\}$	by 1.i
3	$\{\}$	$\{\{\alpha\}\}$	$\{\beta\}$	$\{\}$	$\{\}$	by 2.ii
4	$\{\neg\alpha\}$	$\{\}$	$\{\beta\}$	$\{\alpha\}$	$\{\{\alpha\}\}$	by 2.i.c.1
5	$\{\}$	$\{\}$	$\{\beta\}$	$\{\alpha\}$	$\{\{\alpha\}\}$	by 1.ii
6	$\{\}$	$\{\}$	$\{\beta\}$	$\{\alpha\}$	$\{\}$	by 3, since *Fail*($\{\alpha\}$) holds

Hence, $\neg\alpha$ is an ideal belief and $\{\beta\}$ is the computed ideal support for $\neg\alpha$.

IB-dispute derivations are sound and, for p-acyclic frameworks with an underlying finite language, complete, as proven by the following theorems.

Theorem 4.5. *If there exists an IB-dispute derivation for α, then α is an ideal belief.*

Theorem 4.6. *Given a p-acyclic framework with an underlying finite language, if α is an ideal belief then there exists an IB-dispute derivation for α.*

4.2.2. Fail-dispute derivations

Fail(S) at step 3 of IB-dispute derivations can be computed by means of a new kind of dispute derivations, that we refer to as *Fail-dispute derivations*, obtained again by adapting the dispute derivations of [8].

Definition 4.6. Let $\langle\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-\rangle$ be an assumption based framework. Given a selection function, a *Fail-dispute derivation* of a multiset of sentences S is a sequence $\mathcal{D}_0, \ldots, \mathcal{D}_n$ such that each $\mathcal{D}_i$ is a set of quadruples of the form $\langle\mathcal{P}, \mathcal{O}, A, C\rangle$ where

$$\mathcal{D}_0 = \{\langle S, \{\}, \mathcal{A} \cap S, \{\}\rangle\} \qquad \mathcal{D}_n = \{\}$$

and, for every $0 \leqslant i < n$, if a quadruple $Q = \langle\mathcal{P}, \mathcal{O}, A, C\rangle$ is selected in $\mathcal{D}_i$ then either $\mathcal{P} \neq \{\}$ or $\mathcal{O} \neq \{\}$, and

1. If an element O from $\mathcal{O}$ is selected, then
   (a) if $O = \{\}$ then $\mathcal{D}_{i+1} = \mathcal{D}_i - \{Q\}$;
   (b) if $O \neq \{\}$ then let $\sigma \in O$ be the selected sentence in O:
      i. if σ is not an assumption then $\mathcal{D}_{i+1} = \mathcal{D}_i - \{Q\} \cup \{Q'\}$ where Q' is obtained from Q as in step (2.ii) of Definition 4.3;
      ii. if σ is an assumption then there are two cases:
         Case 1: $\sigma \notin A$. Then $\mathcal{D}_{i+1} = \mathcal{D}_i - \{Q\} \cup \{Q_0, Q_1\}$ where Q_0 is obtained from Q as in step (2.i.a) and Q_1 is obtained from Q as in steps (2.i.b) or (2.i.c) (as applicable) of Definition 4.3;
         Case 2: $\sigma \in A$. Then $\mathcal{D}_{i+1} = \mathcal{D}_i - \{Q\} \cup \{Q_0\}$ where Q_0 is obtained from Q as in step (2.i.a) of Definition 4.3;
2. If a $\sigma \in \mathcal{P}$ is selected, then
   (a) if σ is an assumption then $\mathcal{D}_{i+1} = \mathcal{D}_i - \{Q\} \cup \{Q'\}$ where Q' is obtained from Q as in step (1.i) of Definition 4.3;
   (b) if σ is not an assumption then $\mathcal{D}_{i+1} = \mathcal{D}_i - \{Q\} \cup \{Q'\}$ where there is a rule $\sigma \leftarrow R$ such that Q' is obtained from Q as in step (1.ii) of Definition 4.3.

Example 4.6. Consider the assumption-based framework in Example 2.2. We show here a Fail-dispute derivation of $\{\alpha\}$.

$\mathcal{D}_0 = \{\langle\{\alpha\}, \{\}, \{\alpha\}, \{\}\rangle\}$	applying step 2, we have:
$\mathcal{D}_1 = \{\langle\{\}, \{\{\neg\alpha\}\}, \{\alpha\}, \{\}\rangle\}$	applying step (1.b), we have:
$\mathcal{D}_2 = \{\langle\{\}, \{\{\alpha\}, \{\beta\}\}, \{\alpha\}, \{\}\rangle\}$	applying step (1.b) by selecting $O = \{\alpha\}$ in $\{\{\alpha\}, \{\beta\}\}$ we have:
$\mathcal{D}_3 = \{\langle\{\}, \{\{\}, \{\beta\}\}, \{\alpha\}, \{\}\rangle\}$	applying step (1.a) by selecting $O = \{\}$, we have:
$\mathcal{D}_4 = \{\}$	

Theorem 4.7. *If there exists a Fail-dispute derivation for a multiset of sentences S then $Fail(S)$ holds.*

IB-dispute derivations in which Fail-dispute derivations are used to check whether *Fail*(S) holds, for any S, are sound, as follows:

Corollary 4.2. *If there exists an IB-dispute derivation for α in which Fail-dispute derivations are used to check whether Fail holds, then α is an ideal belief.*

Fail-dispute derivations are complete if AB-dispute derivation are complete for the admissibility semantics. Thus:

Theorem 4.8. *For p-acyclic frameworks with a finite underlying language, if there exists no admissible argument supporting S then there is a Fail-dispute derivation for S.*

From the above theorem, it follows immediately that

Corollary 4.3. *For p-acyclic frameworks with a finite underlying language, IB-dispute derivations in which Fail-dispute derivations are used to check Fail are complete.*

5. Related work

We are not aware of any proof procedure for ideal abstract argumentation. However, there are a number of existing tools for computing sceptical argumentation, notably the TPI procedure [21] and the procedure of [5], for computing the sceptical preferred semantics, and the tools of [14] and [22], for computing the grounded semantics.

The sceptical TPI procedure [21] is defined in terms of the credulous TPI dispute procedure, as follows: an argument is in all preferred extensions if it can be defended in every credulous TPI dispute and none of the attacks against it can be defended in every credulous TPI dispute. This procedure is proven to be sound and complete for *coherent* frameworks [10,21], i.e. frameworks for which the preferred and stable semantics always coincide [7]. Instead, for the ideal semantics, our dispute trees are always sound and complete and our dispute derivations are

- sound for all (coherent and non-coherent) frameworks and
- complete for (coherent and non-coherent) frameworks, as soon as they are p-acyclic (and with a finite underlying language).

Indeed, the completeness results in Section 4.2 only require the p-acyclicity and finiteness of the underlying language conditions, and (as discussed in Section 4.1.2) p-acyclic frameworks are not coherent in general. Note that the restriction to p-acyclic frameworks is rather natural and, in our experience, most assumption-based frameworks are naturally p-acyclic. Indeed, p-acyclicity amounts to the absence of recursive loops in deductions in assumption-based argumentation, e.g. given by rules of the form $p \leftarrow p$.

The algorithm of [5] computes the sceptical preferred semantics for abstract argumentation as follows. Given an argument a, the algorithm proceeds in two separate steps: it first checks that a is not attacked by any admissible set; then, it shows that there exists no preferred extension not containing a. Our IB-dispute derivations are leaner in the sense that they do not require the second step: they just need to compute an admissible set containing a and show that no preferred extension attacks it. Further, these two steps are integrated within IB-dispute derivations. Finally, differently from the algorithm of [5], our IB-dispute derivations are fully defined as disputes between a proponent and an opponent.

Kakas and Toni [14,20] developed argumentation-theoretic proof procedures for the grounded semantics. This is explicitly defined only for logic programs, but, as the authors remark, it can be generalised to any flat assumption-based framework. Compared with their procedure for the grounded semantics, our GB-dispute derivations perform more filtering and are thus more efficient. Moreover, their procedure is defined in terms of argumentation trees whose dialectic nature is less clear than that of GB-dispute derivations.

Vreeswijk [22] presents an algorithm for computing simultaneously (the relevant part of) the grounded extension and all (relevant parts of) the admissible extensions supporting a given belief. The algorithm works by enforcing a labelling (in, out, undecided) on each argument encountered during the computation. It performs filtering in order to terminate as early as possible. Although this algorithm computes all possible admissible sets supporting a given argument, it does not provide an answer whether the argument in question is sceptically supported or not unless it belongs to the grounded extension.

6. Conclusions

We have proposed new proof procedures for computing the ideal semantics, adapted from [1], for argumentation in both abstract and assumption-based frameworks. We have argued that this is a good semantics for performing sceptical argumentation, as it is easily computed and is not overly sceptical.

The proof procedure for abstract argumentation is defined in terms of ideal dispute trees, adapted from the trees of [8] for computing admissible arguments for assumption-based argumentation. The proof procedure for assumption-based argumentation is defined in terms of IB-dispute derivations and Fail-dispute derivations, both adapted from (extensions of) the dispute derivations of [8]. All these derivations extend and generalise standard SLD-based derivations in logic programming, as discussed in [8].

We have proven that the new dispute trees are sound and complete (with respect to the ideal semantics) for any abstract argumentation frameworks, and the new dispute derivations are (with respect to the ideal semantics) sound for any assumption-based frameworks and complete for any p-acyclic assumption-based frameworks with a finite underlying language. In order to prove this completeness result, we have proven a novel completeness result (with respect to the admissible semantics), in the case of p-acyclic frameworks with a finite underlying language, for (a variant of) the procedure proposed in [8]. We have also developed a form of dispute derivations for computing the sceptical grounded semantics for assumption-based argumentation frameworks.

We have exported the ideal semantics of [1] to both abstract and assumption-based argumentation, and studied to some extent its relationship with other, existing semantics. It would be interesting to study further these relationships, for example to see whether in any specific kinds of frameworks the ideal set of arguments always coincides with the intersection of all preferred sets of arguments.

Our procedures build upon the procedures for computing credulous admissible argumentation proposed in [8]. Other procedures exist for computing admissible sets of arguments for abstract argumentation, for instance [5,10,21]. It would be interesting to study whether these procedures could also be extended to compute ideal sets of arguments.

A preliminary implementation of our procedures has been given in the CaSAPI system [11], that has been used to support conflict-resolution amongst mental attitudes in rational agents [12] and decision making [17]. The implementation allows to compute grounded/admissible/ideal supports for given beliefs and visualise, in a rudimental manner, the corresponding dispute derivations. A full evaluation of the implementation and further experimentation are ongoing work.

Finally, it would be interesting to complement the procedures for the ideal semantics in this paper by means of an analysis of the complexity of the ideal semantics for abstract and assumption-based argumentation. Intuitively, this semantics is more complex than the admissible semantics (as its computation requires computing an admissible set first and then performing a check on it). Moreover, its computation is at most that of computing all preferred extensions. Following the results for the admissible and preferred semantics of [6] for (several instances of) assumption-based frameworks, this means that the problem of computing the ideal set of arguments is, for example, at least *NP*-complete and at most Π_2^p-complete for the logic programming instance of assumption-based argumentation (with an underlying

P-complete deductive system), and at least Σ_2^p-complete and at most Π_3^p-complete for the default logic instance of assumption-based argumentation (with an underlying NP-complete deductive system). A full analysis of the computational complexity of the ideal semantics is beyond the scope of this paper and is left for future work.

Acknowledgements

This work was partially funded by the Sixth Framework IST programme of the EC, under the 035200 ARGUGRID project. The third author has also been supported by a UK Royal Academy of Engineering/Leverhulme Trust Senior Research Fellowship. The authors would like to thank all participants of COMMA 2006, and in particular Martin Caminada, for helpful comments and suggestions on an earlier version of this paper.

Appendix A. Proofs

Proof of Lemma 2.1. We observe that given two admissible sets, their union is still admissible, if it does not attack itself. Let now X and Y be two ideal sets, and hence admissible and contained in each preferred set. Since $X \cup Y$ is still contained in each preferred set, it does not attack itself and, by the previous observation, it is still admissible, and hence ideal. □

Proof of Theorem 2.1. (i) follows immediately from Lemma 2.1. Let now $\mathcal{I}$ be the maximal ideal set and x be an argument such that $\mathcal{I}$ attacks every attack against x. Obviously, $\mathcal{I} \cup \{x\}$ is still admissible and it is contained in every preferred set. Hence $x \in \mathcal{I}$ by maximality of $\mathcal{I}$. This proofs (ii).

(iii) follows immediately from (ii) and from Definition 2.2. Finally, (iv) is a direct consequence of Definition 2.2. □

Proof of Theorem 2.2. Let A be an admissible set of assumptions wrt to ABF, and let X_A be the set of arguments in AF containing all arguments supported by any subset of A. Assume that X_A attacks itself. Hence, there exists arguments $a, b \in X_A$ such that a attacks b. Let $a \approx A_1 \vdash \alpha$ and $b \approx A_2 \vdash \beta$, with $A_1 \subseteq A$ and $A_2 \subseteq A$. Hence, by Definition 2.7, $\alpha = \overline{\gamma}$ for some $\gamma \in A_2$, contradicting the fact that A, being admissible, does not attack itself. Hence X_A does not attack itself.

Assume now there exists two arguments x, y such that $x \in X_A$, y attacks x but X_A does not attack y. Again, let $x \approx X \vdash \vartheta$ and $y \approx Y \vdash \varphi$. Since y attacks x, $\varphi = \bar{\delta}$ for some $\delta \in X$. But since $(X \vdash \vartheta) \in A$ and A is admissible, A attacks $\{Y \vdash \varphi\}$, i.e. there exists $\eta \in Y$ such that $\vartheta = \bar{\eta}$. Hence, by construction, x attacks y. Contradiction. This concludes the proof of (i) as far as admissibility is concerned. The proof for grounded/ideal sets is similar and thus omitted.

Consider now (ii) and an admissible set X of arguments in Arg. For each argument $x \in Arg$, let $assumptions(x)$ denote the set of assumptions supporting x (i.e. the set A if $x \approx A \vdash \alpha$). Let now

$$\mathcal{X} = \bigcup_{x \in X} assumptions(x)$$

Assume that $\mathcal{X}$ is not admissible. Then, either $\mathcal{X}$ attacks itself or $\mathcal{X}$ does not attack itself but there is a set of assumptions B such that B attacks $\mathcal{X}$ and $\mathcal{X}$ does not attack B.

Assume $\mathcal{X}$ attacks itself. Then there exists an argument $B \vdash \beta$ such that $B \subseteq \mathcal{X}$ and $\beta = \bar{\alpha}$ for some $\alpha \in \mathcal{X}$. Let $b \approx B \vdash \beta$. It is clear that $\{b\}$ attacks X and, by admissibility of X, there exists some $a \in X$ such that a attacks b. Let $a \approx A' \vdash \alpha'$. Then, $\alpha' = \overline{\beta'}$ for some $\beta' \in B$. Since $B \subseteq \mathcal{X}$, $\beta' \in \mathcal{X}$ and hence a attacks any argument $x' \in X$ such that $\beta' \in assumptions(x')$. Thus, X attacks itself, contradicting the admissibility of X.

Assume now there exists a set of assumptions B such that B attacks $\mathcal{X}$ and $\mathcal{X}$ does not attack B. Hence, there exists an argument $B \vdash \beta$ such that $\beta = \bar{\alpha}$ for some $\alpha \in \mathcal{X}$. Let $b \approx B \vdash \beta$. It is clear that $\{b\}$ attacks X and, by admissibility of X, there exists some $a \in X$ such that a attacks b. Let $a \approx A' \vdash \alpha'$. Then, $\alpha' = \overline{\beta'}$ for some $\beta' \in B$. Since, by construction, $A' \subseteq \mathcal{X}$, by Definition 2.8 $\mathcal{X}$ attacks B. Contradiction! This concludes the proof of (ii) as far as admissible sets are concerned. The proof for grounded/ideal sets is similar and thus omitted. □

Proof of Theorem 3.1. Let P, O be two nodes of a dispute tree such that P is a proponent node and O is an opponent node, and P and O are labelled by the same argument. Then the pair (P, O) is called a *conflicting pair*.

Given two conflicting pairs (P, O) and (P', O'), we define $(P, O) \sqsubset (P', O')$ if P' is a child of O and O' is a child of P. Clearly, the existence of an infinite chain $(P_0, O_0) \sqsubset (P_1, O_1) \sqsubset \cdots$ in a dispute tree implies that the tree is infinite.

We show that any non-admissible dispute tree is infinite. Let $\mathcal{T}'$ be a non-admissible tree. Hence there exists a conflicting pair (P, O) in $\mathcal{T}'$, such that P and O are labelled by the same argument, say p. Let P' be the unique child of O, and assume P' is labelled by p'. By definition of dispute tree, p' attacks p. Hence, there exists a child of P, say O', which is an opponent node and is labelled by p'. Hence, (P', O') is a conflicting pair such that $(P, O) \sqsubset (P', O')$. □

Proof of Theorem 3.2. (i) Let D be the defence set of $\mathcal{T}$. Assume D attacks itself. Then there exist arguments $a, b \in D$ such that a attacks b. Hence, a proponent node P labelled by b has, among others, a child O which is an opponent node and is labelled by a. Hence, a is a label of both a proponent and an opponent node in $\mathcal{T}$, contradicting the admissibility of $\mathcal{T}$. Assume now that there exists an argument b such that b attacks D but D does not attack b. Hence, for some argument $a \in D$ and for some proponent node P labelled by a, there exists an opponent node O labelled by b which is a child of P. By definition of dispute tree, O has a unique child P' which is labelled by some argument a' such that a' attacks b. Since $a' \in D$, D attacks b, which is a contradiction. This concludes the proof of (i).

(ii) We inductively construct a dispute tree where nodes have a ranked assigned such that the proponent nodes have an even rank and the opponent nodes have an odd rank. Moreover, by construction, each proponent node is labelled by an argument in A, and each opponent node is labelled by an argument $b \notin A$.

(1) The root is labelled by a, and is the unique node of rank 0.
(2) Assume that we have constructed all nodes with rank less than or equal than $i = 2k$, $k \geqslant 0$. Let P be a proponent node of rank i, labelled by an argument $a \in A$. Then, for each argument b attacking a, P has one child O which is an opponent node, is labelled by b and has rank $i + 1$. Notice that each such $b \notin A$, by admissibility of A. Since $a \in A$, by admissibility of A there exists $a' \in A$ such that a' attacks b. Hence, we construct the unique child P' of O which is a proponent node, is labelled by a' and has rank $i + 2$.

Notice that the above construction may end up at some finite rank i. In this case, the dispute tree we have constructed is finite (hence admissible) and the defence set A' of the tree is a subset of A by construction. Otherwise, if the dispute tree is infinite, we have by construction an infinite admissible dispute tree. The admissibility of its defence set follows from part (i) of the theorem. □

Proof of Theorem 3.3. Assume S is an ideal set and assume there exists an argument a and an admissible set of arguments A such that a attacks S and $a \in A$. Since A is admissible, there exists a preferred set of arguments P such that $A \subseteq P$. By definition, $S \subseteq P$, hence P attacks itself, contradicting its admissibility.

Assume now that for each argument a attacking S there is no admissible set of arguments containing a, and assume S is admissible. Suppose that there exists a preferred set P such that $S \nsubseteq P$. We show that $P \cup S$ is still admissible, contradicting the fact that P is maximally admissible. By hypothesis, no argument $a \in P$ attacks S, since otherwise there would exist an attack against S contained in an admissible set, contradicting the hypothesis. Hence P cannot attack S. But S cannot attack P either, since otherwise, by admissibility of P, P would attack S. Hence S and P are two admissible sets which do not attack each other, and this implies that $S \cup P$ is still admissible. Contradiction. □

Proof of Theorem 3.4. (i) Let $\mathcal{T}$ be an ideal dispute tree for an argument a and let D be the defence set of $\mathcal{T}$. Assume D is not ideal. Then, by Theorem 3.3, there exists an argument b such that b attacks D and b is contained in some admissible set B. Hence, by Theorem 3.2(ii), there exists an admissible dispute tree for b. Since b must be the label of some opponent node in $\mathcal{T}$, we have a contradiction with the assumption that $\mathcal{T}$ is an ideal tree.

(ii) Let a be an argument belonging to an ideal set A and consider the admissible dispute tree $\mathcal{T}$ for a constructed as in the proof of Theorem 3.3(ii). We show that such a tree is ideal. Assume the contrary, i.e. for some opponent node O labelled by some argument b, there exists an admissible dispute tree $\mathcal{T}'$ for b. Let B be the defence set of $\mathcal{T}'$. By Theorem 3.2(i) B is admissible. Let D be the defence set of $\mathcal{T}$. By construction of $\mathcal{T}$, we have that $D \subseteq A$, b attacks D and b is contained in the admissible set B. This is a contradiction by Theorem 3.3. □

Proof of Theorem 4.1. Trivial, by definition of culprit and of admissibility check. □

Proof of Theorem 4.2. We need first to introduce some new concepts.

Definition A.1. A *partial support tree* of a sentence α (wrt a selection function *sl*) is defined as follows:

1. The root is a node labelled by α.
2. Let N be a node labelled σ. If σ is an assumption, then N is a terminal node. If σ is not an assumption and N is not terminal then there exists some inference rule $\frac{S}{\sigma} \in \mathcal{R}$ and there exists exactly one child of N labelled by δ' for each $\delta' \in S$. If S is empty, N has exactly one child labelled by "true".

A (*full*) *support tree* of α is a partial support tree of α such that a node labelled by a non-assumption σ is terminal iff σ is "true".

It is easy to see that *support trees of α are tree representations of backward deductions of α while partial support trees are tree representations of partial backward deductions*.

Definition A.2. A *partial GB-dispute derivation* is defined as a GB-dispute derivation, dropping the requirement that the components $\mathcal{P}$, $\mathcal{O}$ of the last tuple are empty.

We introduce a notion of partial *tree-based* GB-dispute derivations for illustrating the tree structure of partial GB-dispute derivations where proponent and opponent elements in the GB-dispute derivation are represented by frontier nodes of trees. Intuitively, a partial tree-based GB-dispute derivations is a sequence of dispute trees where partial GB-dispute trees are trees whose nodes are either proponent or opponent nodes. Proponent nodes are labelled by single sentences while opponent nodes are labelled by multisets of sentences where some of these sentences may be marked. *Nodes that have no children and are not labelled by* true *or* false *are called frontier nodes*.

Definition A.3. Given a selection function, a *partial tree-based GB-dispute derivation* for a sentence α is a finite sequence of triples

$$\langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$$

where

- $A_0 = \mathcal{A} \cap \{\alpha\}$,
- $C_0 = \{\}$,
- each T_i is a partial dispute tree,
- the only node in T_0 is its root that is a proponent node labelled by α, and

for every $0 \leqslant i$, exactly one frontier node in T_i is selected, and:

1. If the selected node N is a proponent node labelled by σ then
   - If σ is a non-assumption then there exists some inference rule $\frac{R}{\sigma}$ and

     $$C_i \cap R = \{\}$$
     $$A_{i+1} = A_i \cup (\mathcal{A} \cap R)$$
     $$C_{i+1} = C_i$$

     and T_i is expanded into T_{i+1} by adding, for each literal L in R, a child to N that is a proponent node labelled by L. If R is empty then N has exactly one child, a proponent node labelled by *true*.
   - If σ is an assumption then

     $$A_{i+1} = A_i$$
     $$C_{i+1} = C_i$$

and T_i is expanded into T_{i+1} by adding a child to N that is an opponent node labelled by $\{\overline{\sigma}\}$. Note that $\overline{\sigma}$ is unmarked.

2. If the selected node N is an opponent node labelled by S and an unmarked σ is selected in S then
   (i) if σ is an assumption, then
   expand T_i into T_{i+1} by adding exactly one child to N which is either
   – an opponent node labelled by S with σ becoming marked and

$$A_{i+1} = A_i$$
$$C_{i+1} = C_i$$

   – or an proponent node labelled by $\overline{\sigma}$, and $\sigma \notin A_i$ and

$$A_{i+1} = A_i$$
$$C_{i+1} = C_i \cup \{\sigma\}$$

   (ii) If σ is a non-assumption then

$$A_{i+1} = A_i$$
$$C_{i+1} = C_i$$

   and T_i is expanded into T_{i+1} by adding for each rule $\frac{R}{\sigma}$ a child M to N that is an opponent node labelled by $S - \{\sigma\} \cup R$. If no such rule exists, then N has exactly one child, an opponent node labelled by *false*. Note that the sentences in R are unmarked.

Each partial tree-based GB-dispute derivation

$$\mathcal{T} = \langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$$

can be uniquely transformed into a partial GB-dispute derivation

$$fl(\mathcal{T}) = \langle \mathcal{P}_0, \mathcal{O}_0, A_0, C_0\rangle, \ldots, \langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i\rangle, \ldots, \langle \mathcal{P}_n, \mathcal{O}_n, A_n, C_n\rangle$$

by defining for each i,

$\mathcal{P}_i$ = set of sentences labelling frontier proponent nodes in T_i;
$\mathcal{O}_i$ = set of multisets of sentences labelling frontier opponent nodes in T_i (minus the marked sentences in them).

Definition A.4. Given a selection function, a (*full*) *tree-based GB-dispute derivation of a defence set A* for a sentence α is a finite partial tree-based GB-dispute derivation

$$\langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$$

such that $A = A_n$, and there are no frontier nodes in T_n.

It is not difficult to see that $\mathcal{T}$ is a tree-based GB-dispute derivation iff $fl(\mathcal{T})$ is a GB-dispute derivation.

Trees appearing in partial tree-based GB-dispute derivations are also called *partial GB-dispute trees*. Trees appearing in the last tuple of a tree-based GB-dispute derivation are also called (*full*) *GB-dispute trees*. A path in a partial dispute tree is called a *proponent* (*resp. opponent*) *path* if all the nodes on it are proponent (resp. opponent) nodes.

Let T be a partial GB-dispute tree. The *relative root of a proponent* (*resp. opponent*) *node N in T* is a proponent (resp. opponent) node M in T such that there is a proponent (resp. opponent) path from M to N and if M is not the root of T then the parent of M is an opponent (resp. proponent) node in T. We often also simply call N a *relative root* if N is the relative root of itself.

Definition A.5. Let T be a partial GB-dispute tree and N a proponent node in T. The *context tree of N*, denoted by $ct(N)$, is a subtree of T defined as follows:

– The root of $ct(N)$ is the relative root of N;

– A node N' in T belongs to $ct(N)$ iff it is a proponent node and there is a proponent path from the relative root of N to N'.

It is easy to see that *if N is a proponent node in a partial (resp. full) GB-dispute tree then $ct(N)$ is a partial (resp. full) support tree of α where α is the sentence labelling the relative root of N.*

A *context path* in a partial GB-dispute tree is an opponent path from a relative root.

Let $p = N_0, \ldots, N_k$ be a context path and $S_0, \ldots, S_k$ be the sets labelling the nodes in p. Then $S_0 = \{\overline{\sigma}\}$ for some assumption σ and $S_0, \ldots, S_k$ is a partial backward deduction of σ.

Proof of Theorem 4.2. Let

$$\langle \mathcal{P}_0, \mathcal{O}_0, A_0, C_0\rangle, \ldots, \langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i\rangle, \ldots, \langle \mathcal{P}_n, \mathcal{O}_n, A_n, C_n\rangle$$

be a GB-dispute derivation and

$$\langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$$

be the corresponding tree-based GB-dispute derivation.

By induction on $0 \leqslant i \leqslant n$, we can easily show that $C_i \cap A_i = \emptyset$.

It is not difficult to see that the following lemmas hold:

Lemma A.1. *Let $\sigma \in A_n$ and S be a set of assumptions such that $S \vdash \overline{\sigma}$. Then there is a context path p from some opponent node N in T_n such that N is labelled by $\{\overline{\sigma}\}$ and p is labelled by a sequence $S_0, \ldots, S_k$ such that an assumption is selected at S_k and $S_0, \ldots, S_k$ can be extended into a full backward deduction $S_0, \ldots, S_k, \ldots, S_m$ such that $S_m = S$.*

Lemma A.2. *Let $\sigma \in C_n$ and N be a proponent node labelled by $\overline{\sigma}$. Then the context tree $ct(N)$ is a full support tree such that the set of assumptions labelling the terminal nodes in $ct(N)$ is a subset of A_n.*

The proof that A_n is an admissible subset of the grounded extension is based on the easy fact that if S is an admissible subset of the grounded extension and E is a set of assumptions *acceptable* wrt S (i.e. each argument attacking E is attacked by S), then $S \cup E$ is also an admissible subset of the grounded extension.

Let p be a path in a GB-dispute tree. A *type change* in p is a pair (N, M) of nodes in p such that the types of N, M are different and M is a child of N. The *type-height* (or simply *t-height*) of p is the number of type changes in p. The *t-height of a dispute tree* is the maximum of the t-heights of the paths in it.

The proof is done by induction on the t-height of the GB-dispute tree T_n.

Base case: The t-height is 0. Hence A_n is empty. The theorem holds obviously.

Inductive case: Suppose the theorem holds for cases of t-height less than or equal k. Suppose that the t-height of T_n is $k + 1$. Let $\mathcal{N}$ be the set of all proponent relative roots. Let B denote the set of assumptions labelling the proponent successors of nodes in $\mathcal{N}$. From the induction hypothesis, B is an admissible subset of the grounded extension. Let C be the set of assumptions labelling the terminal nodes of $ct(M)$ where M is the root of the GB-dispute tree. It is clear that $C \subseteq A_n$. Let S be an argument attacking an assumption $\sigma \in C$. From Lemma A.1, t here is a context path p from some opponent node N in T_n such that N is labelled by $\{\overline{\sigma}\}$ and p is labelled by a sequence $S_0, \ldots, S_k$ such that an assumption δ is selected at S_k and $S_0, \ldots, S_k$ can be extended into a full backward deduction $S_0, \ldots, S_k, \ldots, S_m$ such that $S_m = S$.

Therefore there is a proponent node $N \in \mathcal{N}$ such that N is labelled by $\overline{\delta}$. Therefore B attacks δ. Hence B attacks S. This means C is acceptable wrt B. Hence $A_n = B \cup C$ is an admissible subset of the grounded extension. □

Proof of Theorem 4.3. We introduce a notion of *partial tree-based AB-dispute derivation* as an equivalent of AB-dispute derivation. The differences between the new notion and the original one lies in the introduction of dispute trees with *marked* frontier nodes for representing new filtering mechanisms.

Definition A.6. Given a selection function, a *partial tree-based AB-dispute derivation* for a sentence α is a finite sequence of triples

$$\langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$$

where

$$A_0 = \mathcal{A} \cap \{\alpha\}$$
$$C_0 = \{\} \quad \text{and}$$

the only node in T_0 is its root that is an unmarked proponent node labelled by α, and for every $0 \leqslant i$, exactly one unmarked frontier node in T_i is selected, and:

1. If the selected node N is a proponent node labelled by σ then
   – If σ is a non-assumption then there exists some inference rule $\frac{R}{\sigma}$ and $C_i \cap R = \{\}$ (*filtering of defence assumptions by culprits*),

   $$A_{i+1} = A_i \cup (\mathcal{A} \cap R)$$
   $$C_{i+1} = C_i$$

   and T_i is expanded into T_{i+1} by adding, for each literal L in R, a child to N that is a proponent node labelled by L, and mark those labelled by assumptions in $R \cap A_i$ (*filtering of defence assumptions by defences*). All the other new nodes are unmarked. If R is empty then N has exactly one child, a proponent node labelled by *true*.
   – If σ is an assumption then

   $$A_{i+1} = A_i$$
   $$C_{i+1} = C_i$$

   and T_i is expanded into T_{i+1} by adding a child to N that is an opponent node labelled by $\{\overline{\sigma}\}$. The new node is unmarked and $\overline{\sigma}$ is also unmarked.
2. If the selected node N is an opponent node labelled by S and an unmarked σ is selected in S then
   (a) if σ is an assumption, then
      i. either expand T_i into T_{i+1} by adding exactly one child to N which is an unmarked opponent node labelled by S with a marked σ in S and

      $$A_{i+1} = A_i$$
      $$C_{i+1} = C_i$$

      ii. or $\sigma \notin A_i$ and $\sigma \in C_i$ (*filtering culprits by culprits*), and mark N and

      $$A_{i+1} = A_i$$
      $$C_{i+1} = C_i$$

      iii. or $\sigma \notin A_i$ and $\sigma \notin C_i$, and
         A. if $\overline{\sigma}$ is not an assumption then expand T_i into T_{i+1} by adding exactly one child to N which is an unmarked proponent node labelled by $\overline{\sigma}$, and

         $$A_{i+1} = A_i$$
         $$C_{i+1} = C_i \cup \{\sigma\}$$

         B. if $\overline{\sigma}$ is an assumption then mark N and

         $$A_{i+1} = A_i \cup \{\overline{\sigma}\}$$
         $$C_{i+1} = C_i \cup \{\sigma\}$$

   (b) If σ is a non-assumption then

   $$A_{i+1} = A_i$$
   $$C_{i+1} = C_i$$

   and T_i is expanded into T_{i+1} by adding for each rule $\frac{R}{\sigma}$ such that $R \cap C_i = \{\}$, a child M to N that is an unmarked opponent node labelled by $S - \{\sigma\} \cup R$. The sentences in R are unmarked in $S - \{\sigma\} \cup R$. If no such rule exists, then N has exactly one child, an opponent node labelled by *false*.

Each partial tree-based AB-dispute derivation

$$\mathcal{T} = \langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$$

can be uniquely transformed into a partial AB-dispute derivation

$$fl(\mathcal{T}) = \langle \mathcal{P}_0, \mathcal{O}_0, A_0, C_0\rangle, \ldots, \langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i\rangle, \ldots, \langle \mathcal{P}_n, \mathcal{O}_n, A_n, C_n\rangle$$

by defining, for each i,

$\mathcal{P}_i$ = set of sentences labelling the unmarked frontier proponent nodes in T_i;
$\mathcal{O}_i$ = set of multisets of sentences labelling the unmarked frontier opponent nodes in T_i (minus the marked sentences in them).

Given a selection function, a (*full*) *tree-based AB-dispute derivation of a defence set A* for a sentence α is a finite partial tree-based AB-dispute derivation

$$\langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$$

such that $A = A_n$, and all frontier nodes in T_n are marked nodes.

It is not difficult to see that $\mathcal{T}$ is a tree-based AB-dispute derivation iff $fl(\mathcal{T})$ is a (full) AB-dispute derivation.

Trees appearing in partial tree-based AB-dispute derivation are also called *partial AB-dispute trees*. Trees appearing in the last tuple of a tree-based AB-dispute derivation are also called (*full*) *AB-dispute trees*. A path in a partial dispute tree is called a *proponent* (*resp. opponent*) *path* if all the nodes on it are proponent (resp. opponent) nodes.

The notions of *relative root, context tree and context path* are defined for partial AB-dispute trees similarly to those defined for partial GB-dispute trees.

Similarly, it holds also that *if N is a proponent node in a partial* (*resp. full*) *AB-dispute tree then $ct(N)$ is a partial* (*resp. full*) *support tree of α where α is the sentence labelling the root of $ct(N)$.*

It also holds that if *$p = N_0, \ldots, N_k$ is the context path of N_k and $S_0, \ldots, S_k$ are the sets labelling the nodes in p, then $S_0 = \{\overline{\sigma}\}$ for some assumption σ and $S_0, \ldots, S_k$ is a partial backward deduction of σ.*

Let

$$\langle \mathcal{P}_0, \mathcal{O}_0, A_0, C_0\rangle, \ldots, \langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i\rangle, \ldots, \langle \mathcal{P}_n, \mathcal{O}_n, A_n, C_n\rangle$$

be an AB-dispute derivation and

$$\langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$$

be the corresponding tree-based AB-dispute derivation. By induction on $0 \leqslant i \leqslant n$, we can easily show that $C_i \cap A_i = \emptyset$.

It is not difficult to see that the following lemmas hold.

Lemma A.3. *Let $\sigma \in A_n$ and S be a set of assumptions such that $S_0, \ldots, S_k, \ldots, S_m$ is a full backward deduction of $\overline{\sigma}$ and $S = S_m$. Then there is an opponent node N in T_n labelled by $\{\overline{\sigma}\}$ such that there is a opponent path in T_n from N to some node M labelled by the sequence $S_0, \ldots, S_k$ in T_n, and*

1. *either the rule $\frac{R}{\sigma}$ used at step S_k in the deduction satisfies $C_i \cap R \neq \{\}$,*
2. *or an assumption is selected at S_k.*

Lemma A.4. *Let $\sigma \in C_n$. Then there is a proponent node N labelled by $\overline{\sigma}$ and for each such nodes N, the context tree $ct(N)$ is a full support tree whose set of assumptions labelling the terminal nodes is a subset of A_n.*

Proof of Theorem 4.3. Let S be an argument attacking $\sigma \in A_n$. From Lemma A.3, there is a full backward deduction of $\overline{\sigma}$: $S_0, \ldots, S_k, \ldots, S_m$, $S = S_m$. Then there is an opponent node N in T_n labelled by $\{\overline{\sigma}\}$ such that there is a opponent path in T_n from N labelled by the sequence $S_0, \ldots, S_k$ in T_n, and

1. either the rule $\frac{R}{\sigma}$ used at step S_k in the deduction satisfies $C_k \cap R \neq \{\}$,
2. or an assumption is selected at S_k.

In the first case, it is clear that $C_k \cap R \subseteq \mathcal{A} \cap R \subseteq S$. Hence there is an assumption $\delta \in C_n \cap S$.

In the second case, let δ be the assumption selected at S_k. Therefore $\delta \in C_n$. Since $\delta \in S$, we get $\delta \in C_n \cap S$.

Let M be a proponent node labelled by $\bar{\delta}$. From Lemma A.4, $ct(M)$ is a full support tree such that all the assumptions labelling its terminal nodes belong to A_n. Hence $A_n \vdash \bar{\delta}$. Hence A_n attacks S.

Suppose that A_n is not conflict free. Hence there is $\sigma \in A_n$ such that $S \vdash \bar{\sigma}$ for some $S \subseteq A_n$. From our above elaboration, there is $\delta \in C_n \cap S$. Contradiction to the fact that C_n, A_n are disjoint. □

Proof of Theorem 4.4. From the correspondence between tree-based AB-dispute derivations and AB-dispute derivations, we will work on the tree-based version. Let S be an admissible set of assumptions and α be a sentence such that $S \Vdash \alpha$.

An opponent node in a AB-dispute tree is called *locally terminal* if either it is marked or its only child is a proponent node.

We show now that there is no infinite partial AB-dispute derivation in p-acyclic, finite assumption-based framework.

Viewing an infinite partial AB-dispute derivation as an infinite partial tree-based AB-dispute derivation

$$\langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots$$

we can conclude that there is an infinite path p in the tree T that is the limit of trees T_i's.

Due to the p-acyclicity, there are infinitely many locally terminal opponent nodes in p.

Due to the finiteness of the language, there exists n such that for all $m \geqslant n$, $A_m = A_n$ and $C_m = C_n$.

Let N be a locally terminal node on p and N belongs to T_m, $m \geqslant n$. Therefore the assumption that is selected at N must belong to C_n. Hence there is no children of N in T_{m+1} and hence in T. Contradiction to the assumption of the infinite length of p. Hence there is no infinite path in T. Hence there is no infinite partial AB-dispute derivation.

Let

$$\langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$$

be a partial tree-based AB-dispute derivation for α, $S \Vdash \alpha$ satisfying:

- whenever a proponent node N labelled by γ with $S \Vdash \gamma$, is selected then a rule $\frac{R}{\gamma}$ is selected for expansion such that $S \Vdash R$, and
- when an opponent node labelled by O is selected together with an assumption δ from O then δ is ignored iff $S \nVdash \bar{\delta}$.

It is easy to prove the following lemma inductively for $0 \leqslant i \leqslant n$:

Lemma A.5.

1. *Let N be a proponent node labelled by a non-assumption γ in T_i. Then $S \Vdash \gamma$.*
2. *Let N be a proponent node labelled by an assumption σ. Then $\sigma \in S$.*
3. $A_i \subseteq S$.
4. *For each $\delta \in C_i$, S attacks δ.*
5. *Suppose that M is an opponent node labelled by O in T_i such that all sentences in O are assumptions. Further let N be the relative root of M. Then N is labelled by $\{\bar{\sigma}\}$ for some $\sigma \in S$, and the context path from N to M in T_i is a full backward deduction of $\bar{\sigma}$.*
6. *If a sentence δ in a multiset labelling an opponent node is marked, it is an assumption such that $S \nVdash \bar{\delta}$.*

We have proved that there is no infinite partial tree-based AB-dispute derivation. Let us suppose now that there exists a partial tree-based AB-dispute derivation that can not be expanded further. If this derivation is a full filtered tree-based dispute derivation then we are done. Suppose now the contrary.

Looking at the definition of AB-dispute derivation, there are two cases:

Case 1: The selected node is an unmarked proponent node labelled by a non-assumption, or
Case 2: The selected node is an unmarked opponent node labelled by a multiset S_0. We analyse the two cases in turn.

Case 1: Let the non-assumption selected be γ. From the Lemma A.5, it follows that $S \Vdash \gamma$. Hence there is a rule $\frac{R}{\gamma}$ such that $S \Vdash R$. Therefore, $R \cap \mathcal{A} \subseteq S$. Since S is admissible, and S attacks every assumption in C_n, it follows that $R \cap \mathcal{A} \cap C_n$ is empty. Hence it is possible to expand the derivation at this step. Contradiction, therefore this case does not occur.

Case 2: There are again two cases here:

Case 2.1: All sentences in S_0 are marked. From Lemma A.5, assertion (5), it follows that S_0 is an argument against an assumption in S. From Lemma A.5, assertion (6), it follows that $S \nVdash \bar{\delta}$ for each $\delta \in S_0$. This is a contradiction to the admissibility of S. Hence this case is not possible.

Case 2.2: There is at least on unmarked sentence in S_0. Therefore the sentence selected is an assumption δ since the derivation cannot be expanded further. As step (2.i.a) could always be applicable if $S \nVdash \bar{\delta}$, it follows $S \Vdash \bar{\delta}$. It is clear that one of the steps (2.i.b) or (2.i.c) in the definition of AB-dispute derivation is possible. Hence the derivation could be expanded. hence this case is also not possible.

We have thus proved that there exists a full tree-based AB-dispute derivation for α whose defence assumptions is a subset of S. $\square$

Proof of Theorem 4.5. Similarly to the proofs of the previous theorems, we introduce a notion of partial tree-based IB-dispute derivation as an equivalent of partial IB-dispute derivation. The differences between the new notion and the tree-based AB-dispute derivation lies in the introduction of *checked locally terminal opponent nodes* referring to the elements in $\mathcal{F}_i$ where an opponent node in a AB-dispute tree is called *locally terminal* if either it is marked or its only child is a proponent node.

Definition A.7. The definition of a partial tree-based IB-dispute derivation is similar to the definition of partial tree-based AB-dispute derivation with the following modifications:

– In step (2.b) where an opponent node labelled by S and a non-assumption σ is selected, T_i is expanded into T_{i+1} by adding for each rule $\frac{R}{\sigma}$ a child M to N that is an opponent node labelled by $S - \{\sigma\} \cup R$. The new node is unmarked if $R \cap C_i = \{\}$. Otherwise it is marked. The sentences in R are unmarked in $S - \{\sigma\} \cup R$. If no such rule exists, then N has exactly one child, an opponent node labelled by *false*,

$$A_{i+1} = A_i$$
$$C_{i+1} = C_i$$

– A new step 3 is added, where a locally terminal and unchecked opponent node N labelled by S is selected and $\mathit{Fail}(u(S))$ holds and N becomes checked.

Each partial tree-based IB-dispute derivation

$$\mathcal{T} = \langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$$

can be uniquely transformed into a partial IB-dispute derivation

$$\mathit{fl}(\mathcal{T}) = \langle \mathcal{P}_0, \mathcal{O}_0, A_0, C_0, \mathcal{F}_0\rangle, \ldots, \langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i, \mathcal{F}_i\rangle, \ldots, \langle \mathcal{P}_n, \mathcal{O}_n, A_n, C_n, \mathcal{F}_n\rangle$$

by defining for each i,

$\mathcal{P}_i$ = set of sentences labelling the unmarked frontier proponent nodes in T_i,
$\mathcal{O}_i$ = set of multisets of sentences labelling the unmarked frontier opponent nodes in T_i (minus the marked sentences in them),
$\mathcal{F}_i$ = set of multisets of sentences labelling the unchecked locally minimal opponent nodes in T_i (where all marked sentences in these multisets are unmarked).

Given a selection function, a (*full*) *tree-based IB-dispute derivation of a defence set A* for a sentence α is a finite partial tree-based IB-dispute derivation $\langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$ such that $A = A_n$, and all frontier nodes in T_n are marked, and all locally minimal opponent nodes are checked.

It is not difficult to see that $\mathcal{T}$ is a tree-based IB-dispute derivation iff $fl(\mathcal{T})$ is a (full) IB-dispute derivation.

Trees appearing in partial tree-based IB-dispute derivation are also called *partial IB-dispute trees*. Trees appearing in the last tuple of a tree-based IB-dispute derivation are also called (*full*) *IB-dispute trees*. A path in a partial dispute tree is called a *proponent* (*resp. opponent*) *path* if all the nodes on it are proponent (resp. opponent) nodes.

The notions of *relative root, context tree and context path* are defined for partial IB-dispute trees similarly to those defined for partial AB-dispute trees.

Similarly, it holds also that *if N is a proponent node in a partial* (*resp. full*) *IB-dispute tree then $ct(N)$ is a partial* (*resp. full*) *support tree of α where α is the sentence labelling the root of $ct(N)$.*

It also holds that if $p = N_0, \ldots, N_k$ *is the context path of* N_k *and* $S_0, \ldots, S_k$ *be the sets labelling the nodes in* p. *Then* $S_0 = \{\overline{\sigma}\}$ *for some assumption* σ *and* $S_0, \ldots, S_k$ *is a partial backward deduction of* σ.

Let

$$\langle \mathcal{P}_0, \mathcal{O}_0, A_0, C_0, \mathcal{F}_0\rangle, \ldots, \langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i, \mathcal{F}_i\rangle, \ldots, \langle \mathcal{P}_n, \mathcal{O}_n, A_n, C_n, \mathcal{F}_n\rangle$$

be an IB-dispute derivation and

$$\langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$$

be the corresponding tree-based IB-dispute derivation.

By induction on $0 \leqslant i \leqslant n$, we can easily show that $C_i \cap A_i = \emptyset$.

It is not difficult to see that the following lemmas hold.

Lemma A.6. *Let $\sigma \in A_n$ and S be a set of assumptions such that $S_0, \ldots, S_k, \ldots, S_m$ be a full backward deduction of $\overline{\sigma}$ and $S = S_m$. Then there is a relative root N in T_n labelled by $\{\overline{\sigma}\}$ such that there is a context path labelled by the sequence $S_0, \ldots, S_k$, in T_n from N to a locally terminal opponent node labelled by S_k.*

Lemma A.7. *Let $\sigma \in C_n$. then there is a proponent node N labelled by $\overline{\sigma}$ and for each such node N, the context tree $ct(N)$ is a full support tree whose set of assumptions labelling the terminal nodes is a subset of A_n.*

It is clear that A_n is admissible. It remains to show that A_n is ideal. Suppose that there exists an admissible set S such that there exists an argument $S_0 \subseteq S$ against an assumption $\sigma \in A_n$. From Lemma A.6, there is a relative root N in T_n labelled by $\{\overline{\sigma}\}$ such that there is a context path labelled by a sequence $S_0, \ldots, S_k$, in T_n from N to a locally terminal opponent node M labelled by S_k and $S_0, \ldots, S_k$ can be extended into a full backward deduction $S_0, \ldots, S_k, \ldots, S_m$ of $\overline{\sigma}$ and $S_0 = S_m$. Hence $S_0 \Vdash S_k$. Because T_n is a full IB-dispute tree, all locally terminal opponent nodes are checked. Hence $Fail(u(S_k))$ holds, i.e. there is no admissible set E of assumptions such that $E \Vdash S_k$, contradiction to the fact that $S_0 \Vdash S_k$ and $S_0 \subseteq S$.

Proof of Theorem 4.6. From the construction in the proof of Theorem 4.4, there exists a tree-based AB-dispute derivation

$$\langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle$$

for α. Let $\mathcal{N} = \{N_1, \ldots, N_k\}$ be the set of locally terminal nodes in T_n.

Extend this derivation into a tree-based AB-dispute derivation

$$\langle T_0, A_0, C_0\rangle, \ldots, \langle T_i, A_i, C_i\rangle, \ldots, \langle T_n, A_n, C_n\rangle,$$
$$\langle T_{n+1}, A_{n+1}, C_{n+1}\rangle, \ldots, \langle T_{n+k}, A_{n+k}, C_{n+k}\rangle$$

as follows:

– In step (2.b) where an opponent node labelled by S and a non-assumption σ is selected, T_i is expanded into T_{i+1} by adding for each rule $\frac{R}{\sigma}$ a child M to N that is an opponent node labelled by $S - \{\sigma\} \cup R$. The new node is unmarked if $R \cap C_i = \{\}$. Otherwise it is marked. The sentences in R are unmarked in $S - \{\sigma\} \cup R$. If no such rule exists, then N has exactly one child, an opponent node labelled by *false*,

$$A_{i+1} = A_i$$
$$C_{i+1} = C_i$$

– At each step i from $n+1$ to $n+k$, $Fail(u(S_i))$ is checked where S_i is the set labelling N_i.

It is obvious that the obtained derivation is a tree-based IB-dispute derivation for α. □

Proof of Theorem 4.7. To simplify the proofs, we introduce, as in other proofs, a notion of tree-based Fail-dispute derivation.

Given a selection function, a *tree-based Fail-dispute derivation* of a multiset of sentences S is a sequence $\mathcal{T}_0, \ldots, \mathcal{T}_n$ such that each $\mathcal{T}_i$ is a tree whose nodes are labelled by quadruples of the form $\langle \mathcal{P}, \mathcal{O}, A, C \rangle$ where

$\mathcal{T}_0$ contains exactly one node labelled by $\langle S, \{\}, \mathcal{A} \cap S, \{\} \rangle$, and
all terminal nodes in $\mathcal{T}_n$ are labelled by *false*, and
for every $0 \leqslant i < n$, if a node N labelled by a quadruple $Q = \langle \mathcal{P}, \mathcal{O}, A, C \rangle$ is selected in $\mathcal{T}_i$ then either $\mathcal{P} \neq \{\}$ or $\mathcal{O} \neq \{\}$, and

1. If an element O from $\mathcal{O}$ is selected, then
   (a) If $O = \{\}$ then $\mathcal{T}_{i+1}$ is obtained from $\mathcal{T}_i$ by adding an unique child to N labelled by *false*;
   (b) If $O \neq \{\}$ then let $\sigma \in O$ be the selected sentence in O:
      i. if σ is not an assumption then $\mathcal{T}_{i+1}$ is obtained from $\mathcal{T}_i$ by adding an unique child to N labelled by Q' where Q' is obtained from Q as in step (2.ii) of Definition 4.3;
      ii. if σ is an assumption then there are two cases:
         Case 1: $\mathcal{T}_{i+1}$ is obtained from $\mathcal{T}_i$ by adding two children to N labelled by Q_0, Q_1 where Q_0 is obtained from Q as in step (2.i.a) and Q_1 is obtained from Q as in steps (2.i.b) or (2.i.c) (as applicable) of Definition 4.3;
         Case 2: $\sigma \in A$. Then $\mathcal{D}_{i+1} = \mathcal{D}_i - \{Q\} \cup \{Q_0\}$ where Q_0 is obtained from Q as in step (2.i.a) of Definition 4.3;
2. If a $\sigma \in \mathcal{P}$ is selected, then
   (a) if σ is an assumption then $\mathcal{T}_{i+1}$ is obtained from $\mathcal{T}_i$ by adding an unique child to N labelled by Q' where Q' is obtained from Q as in step (1.i) of Definition 4.3;
   (b) if σ is not an assumption then $\mathcal{T}_{i+1}$ is obtained from $\mathcal{T}_i$ by adding for each Q' in the set $\{Q' \mid$ there is a rule $\sigma \leftarrow R$ such that Q' is obtained from Q as in step (1.ii) of Definition 4.3$\}$, a child of N labelled by Q'.

The correspondence between tree-based Fail-dispute derivations and Fail-dispute derivations are obvious. It is immediately to see that *there exists a Fail-dispute derivation for S if and only if there exists a tree-based Fail-dispute derivation for S*.

Let $\mathcal{T}_0, \ldots, \mathcal{T}_m$ be a tree-based Fail-dispute derivation for S. Suppose that $Fail(S)$ does not hold, i.e. there is a admissible set E such that $E \Vdash S$. We prove by induction on $0 \leqslant i \leqslant m$ that:

Lemma A.8. *There is a path from the root to a frontier node in $\mathcal{T}_i$ such that for each quadruple $Q = \langle \mathcal{P}, \mathcal{O}, A, C \rangle$ occurring on this path, $A \subseteq E$ and $E \Vdash P$.*

It is clear that the claim holds for $i = 0$.

Suppose the claim holds for i. Let N be a frontier node in T_i such that the quadruple $Q = \langle \mathcal{P}, \mathcal{O}, A, C \rangle$ labelling N satisfying $A \subseteq E$ and $E \Vdash \mathcal{P}$. If node N is not selected, then the claim obviously holds for T_{i+1}.

Suppose now that N is selected. We prove that N has a child M labelled by $Q' = \langle \mathcal{P}', \mathcal{O}', A', C' \rangle$ satisfying $A' \subseteq E$ and $E \Vdash \mathcal{P}'$. There are several cases:

Case 1: A sentence $\alpha \in \mathcal{P}$ is selected.

Case 1.1: α is an assumption. Then N has exactly one child in T_{i+1} and it is obvious that the claim holds for T_{i+1} as the child of N satisfies the required property.

Case 1.2: α is a non-assumption. Then there is a rule $\frac{R}{\alpha}$ such that $E \Vdash R$ since $E \Vdash \alpha$. Hence there is a child M of N in T_{i+1} where the quadruple labelling M is obtained from Q using the rule $\frac{R}{\alpha}$. It is obvious that M satisfies the required property in T_{i+1}.

Case 2: A set $O \in \mathcal{O}$ is selected together with an sentence $\alpha \in O$.

Case 2.1: α is a non-assumption. Then N has an unique child in T_{i+1} and it is obvious that this child satisfies the required property.

Case 2.2: α is an assumption. If $\alpha \in A$ then N has exactly one child whose label satisfies the required property.

Let $\alpha \notin A$. If E attacks α, then the child of N obtained according to the step (2.i.b) or (2.i.c) in the AB-dispute derivation is the frontier node in T_{i+1} satisfies the required property. If E does not attack α, then the child of N obtained according to the step (2.i.a) in the AB-dispute derivation is the frontier node in T_{i+1} satisfying the required properties.

From Lemma A.8, there is a frontier node in T_n labelled by $Q = \langle \mathcal{P}, \mathcal{O}, A, C \rangle$ such that $A \subseteq E$ and $E \Vdash P$. Contradiction to the definition of tree-based Fail-dispute derivation where all the frontier nodes in T_n are labelled by *false*. □

Proof of Theorem 4.8. Let *Fail*(S) hold. Suppose that there is no Fail-dispute derivation for S. From the definition of Fail-dispute derivation for S, it is clear that for each choice of a quadruple at step i, it is always possible to proceed to step $i + 1$. Hence, it follows immediately that there exists an infinite partial Fail-dispute derivation $\mathcal{D}_0, \ldots, \mathcal{D}_n, \ldots$ such that $\mathcal{D}_n \neq \emptyset$ for all n. Viewing an infinite partial Fail-dispute derivation as an infinite partial tree-based Fail-dispute derivation, it follows immediately that there is an infinite partial AB-derivation

$$\langle \mathcal{P}_0, \mathcal{O}_0, A_0, C_0 \rangle, \ldots, \langle \mathcal{P}_i, \mathcal{O}_i, A_i, C_i \rangle, \ldots$$

Viewing an infinite partial AB-derivation as an infinite partial tree-based AB-derivation

$$\langle T_0, A_0, C_0 \rangle, \ldots, \langle T_i, A_i, C_i \rangle, \ldots$$

we can conclude that there is an infinite path p in the tree T that is the limit of trees T_i's.

Due to the p-acyclicity, there are infinitely many locally terminal opponent nodes in p.

Due to the finiteness of the language, there exists n such that for all $m \geqslant n$, $A_m = A_n$ and $C_m = C_n$.

Let N be a locally terminal node on p and N belongs to T_m, $m \geqslant n$. Therefore the assumption that is selected at N must belong to C_n. Hence there is no children of N in T_{m+1} and hence in T. Contradiction to the assumption of the infinite length of p. Hence there is no infinite path in T. Hence T is finite. Therefore there exists a Fail-dispute derivation for S. □

References

[1] J.J. Alferes, P.M. Dung, L.M. Pereira, Scenario semantics of extended logic programs, in: A. Nerode, L. Pereira (Eds.), Proc. 2nd International Workshop on Logic Programming and Non-monotonic Reasoning, MIT Press, 1993, pp. 334–348.

[2] C. Baral, M. Gelfond, Logic programming and knowledge representation, Journal of Logic Programming 19/20 (1994) 73–148.

[3] A. Bondarenko, P.M. Dung, R.A. Kowalski, F. Toni, An abstract, argumentation-theoretic approach to default reasoning, Artificial Intelligence 93 (1–2) (1997) 63–101.

[4] A. Bondarenko, F. Toni, R.A. Kowalski, An assumption-based framework for non-monotonic reasoning, in: A. Nerode, L. Pereira (Eds.), Proc. 2nd International Workshop on Logic Programming and Non-Monotonic Reasoning, MIT Press, 1993, pp. 171–189.

[5] C. Cayrol, S. Doutre, J. Mengin, On decision problems related to the preferred semantics for argumentation frameworks, Journal of Logic and Computation 13 (3) (2003) 377–403.

[6] Y. Dimopoulos, B. Nebel, F. Toni, On the computational complexity of assumption-based argumentation for default reasoning, Artificial Intelligence 141 (2002) 57–78.

[7] P.M. Dung, On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games, Artificial Intelligence 77 (1995) 321–357.

[8] P.M. Dung, R.A. Kowalski, F. Toni, Dialectic proof procedures for assumption-based, admissible argumentation, Artificial Intelligence 170 (2006) 114–159.

[9] P.M. Dung, P. Mancarella, F. Toni, A dialectic procedure for sceptical, assumption-based argumentation, in: 1st International Conference on Computational Models of Argument (COMMA 2006), September 2006.

[10] P.E. Dunne, T.J.M. Bench-Capon, Two party immediate response disputes: properties and efficiency, Artificial Intelligence 149 (2003) 221–250.

[11] D. Gartner, F. Toni, CaSAPI: a system for credulous and sceptical argumentation, in: G. Simari, P. Torroni (Eds.), Proc. Workshop on Argumentation for Non-Monotonic Reasoning, 2007, pp. 80–95.

[12] D. Gartner, F. Toni, Conflict-free normative agents using assumption-based argumentation, in: Proc. of the Fourth International Workshop on Argumentation in Multi-Agent Systems, 2007, pp. 138–153.

[13] G. Governatori, M.J. Maher, D. Billington, G. Antoniou, Argumentation semantics for defeasible logics, Journal of Logic and Computation 14 (5) (2004) 675–702.

[14] A.C. Kakas, F. Toni, Computing argumentation in logic programming, Journal of Logic and Computation 9 (1999) 515–562.

[15] A.C. Kakas, P. Moraitis, Argumentation based decision making for autonomous agents, in: Proceedings of AAMAS '03, 2003, pp. 883–890.

[16] R.A. Kowalski, F. Toni, Abstract argumentation, Journal of Artificial Intelligence and Law (Special Issue on Logical Models of Argumentation) 4 (3–4) (1996) 275–296.

[17] M. Morge, P. Mancarella, The hedgehog and the fox. An argumentation-based decision support system, in: Proc. of the Fourth International Workshop on Argumentation in Multi-Agent Systems, 2007, pp. 55–68.

[18] J. Pollock, Defeasible reasoning, Cognitive Science 11 (4) (1987) 481–518.

[19] H. Prakken, G. Sartor, The role of logic in computational models of legal argument: a critical survey, in: A.C. Kakas, F. Sadri (Eds.), Computational Logic: Logic Programming and Beyond—Essays in Honour of Robert A. Kowalski, in: Lecture Notes Artif. Intell., vol. 2408, Springer-Verlag, 2002, pp. 342–381.

[20] F. Toni, A.C. Kakas, Computing the acceptability semantics, in: V.W. Marek, A. Nerode, M. Truszczynski (Eds.), Proc. 3rd International Workshop on Logic Programming and Non-Monotonic Reasoning, in: Lecture Notes Artif. Intell., vol. 928, Springer-Verlag, 1995, pp. 401–415.

[21] G. Vreeswijk, H. Prakken, Credulous and sceptical argument games for preferred semantics, in: Proc. JELIA, in: Lecture Notes Comput. Sci., vol. 1919, Springer-Verlag, 2000, pp. 224–238.

[22] G. Vreeswijk, H. Prakken, An algorithm to compute minimally grounded and admissible defence sets, in: 1st International Conference on Computational Models of Argument (COMMA 2006), 2006.