Book Title
Book Editors
IOS Press, 2003

A dialectic procedure for sceptical, assumption-based argumentation

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

Abstract. We present a procedure for computing the sceptical "ideal semantics" for argumentation in assumption-based frameworks. This semantics was first proposed for logic programming in [1], extending the well-founded semantics. The proof procedure is defined by means of a form of dispute derivations, obtained by modifying the dispute derivations given in [2] for computing credulous admissible argumentation. The new dispute derivations are sound for the "ideal semantics" in all cases where the dispute derivations of [2] are complete for admissible argumentation. We prove that this is the case for the special kind of assumption-based frameworks with a finite underlying language and with the property of being "p-acyclic".

Keywords. Argumentation frameworks, Tools for argumentation

1. Introduction

We present a novel procedure for computing argumentation in the abstract, assumption-based frameworks of [3]. In these frameworks, arguments are built by means of deductions from assumptions, which are the components of the argument to be disputed by counter-arguments. These frameworks have been originally proposed for modelling default and legal reasoning [3,4], but have been equipped with powerful machinery for general-purpose argumentation in [2]. This machinery amounts to a procedure, in terms of *dispute derivations*, for computing arguments deemed acceptable according to the semantics of admissible sets of assumptions. This procedure uses *tight arguments*, which can be computed effectively by backward deductions.

The semantics of admissible sets of assumptions 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 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 (arguments) for achieving goals, and, when negotiating

[1]Correspondence to: P. Mancarella, Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo, 3 I-56127 Pisa (Italy). Tel.: +39 050 2212710; Fax: +39 050 2212726; E-mail: paolo.mancarella@unipi.it

resources, they may decide to give away a resource only if that resource is not needed to support *any* of their plans.
Procedures for the computation of the sceptical semantics exist, e.g. the TPI procedure [12] for *coherent* argumentation frameworks [13]. However, to the best of our knowledge, no procedure exists for computing sceptical reasoning for non-coherent cases.
The procedure in this paper computes the sceptical *ideal semantics* for assumption-based frameworks. This is adapted from a corresponding semantics for logic programming, presented in [1]. The ideal semantics has the advantage of being easily computable, by a simple modification of the dispute derivations of [2], but without being overly sceptical. We prove that our procedure is sound for assumption-based frameworks with a finite underlying language and with no positive cycles (we call such frameworks p-acyclic). The proofs are omitted for lack of space, and are given in the accompanying report [5].

2. Background

In this section we briefly review the notion of assumption-based framework [3,4,6], how it applies to argumentation [2], the semantics of admissible sets of assumptions [3,9], and various possible sceptical semantics [3,9].
Any logic, viewed as a deductive system, can be extended to an assumption-based argumentation framework.

Definition 2.1 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 $\alpha \leftarrow \alpha_1, \ldots, \alpha_n$ where $\alpha, \alpha_1, \ldots, \alpha_n \in \mathcal{L}$ and $n \geq 0$. [1] α is called the **conclusion** and $\alpha_1, \ldots, \alpha_n$ are called the **premises** of the inference rule.

If $n = 0$, then the inference rule represents an axiom. For notational convenience, we simply write α instead of $\alpha \leftarrow$.

Definition 2.2 A **deduction of a conclusion** α **based on a set of premises** P is a sequence $\beta_1, \ldots, \beta_m$ of sentences in $\mathcal{L}$, where $m > 0$ and $\alpha = \beta_m$, such that, for all $i = 1, \ldots, m$,

- $\beta_i \in P$, or
- there exists $\beta_i \leftarrow \alpha_1, \ldots, \alpha_n \in \mathcal{R}$ such that $\alpha_1, \ldots, \alpha_n \in \{\beta_1, \ldots, \beta_{i-1}\}$.

If there is a deduction of a conclusion α based on a set of premises P, we say that the deduction is **supported by** or **based upon** P.

Deductions are the basis for the construction of arguments, but to obtain an argument from a deduction its premises are restricted to ones that are acceptable as *assumptions*. In this paper, as in [2], we restrict ourselves to *flat* frameworks [3], whose assumptions do not occur as conclusions of inference rules. To specify when one argument attacks another, we need to determine when a sentence is the *contrary* of an assumption.

Definition 2.3 An **assumption-based framework** is a tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ where

- $(\mathcal{L}, \mathcal{R})$ is a deductive system.

[1] [2] uses the equivalent notation $\frac{\alpha_1, \ldots, \alpha_n}{\alpha}$ for inference rules.

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

Notice that, given an assumption α, $\overline{\alpha}$ may or may not be an assumption in general. Throughout the paper, following [2], we will illustrate our computational techniques by means of examples within *simplified* frameworks of the form $\langle\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{}\rangle$where:

- All sentences in $\mathcal{L}$ are atoms or negations of atoms (i.e. $\mathcal{L}$ is a set of **literals**).
- The contrary of any assumption p is $\neg p$; the contrary of any assumption $\neg p$ is p.

Definition 2.4 An **argument** is a deduction whose premises are all assumptions.

The only way to attack an argument is to attack one of its assumptions.

Definition 2.5

- An **argument** a **attacks an argument** b if and only if a attacks an assumption in the set of assumptions on which b is based.
- An **argument** a **attacks an assumption** α if and only if the conclusion of a is the contrary $\overline{\alpha}$ of α.
- A **set of assumptions** A **attacks a set of assumptions** B if and only if there exists an argument a based upon a set of assumptions $A' \subseteq A$ which attacks an assumption in B.

Note that the attack relationship between arguments depends solely on sets of assumptions. In some other approaches, however, such as that of Pollock [7] and Prakken and Sartor [8], an argument can attack another by contradicting its conclusion. Here, instead, such "rebuttal" attacks are reduced to "undermining" attacks, as described in [4,2].
The attack relationship is the basis of the *admissibility* semantics, first introduced in [9].

Definition 2.6

- A **set of assumptions** A is **admissible** if and only if
  A attacks every set of assumptions that attacks A, and
  A does not attack itself.
- A **belief** α is **admissible** if and only if there exists an argument for α based on a set of assumptions A_0, and A_0 is a subset of an admissible set A.

This semantics provides a non-constructive specification for which we need a practical, constructive and efficient, procedure. A major source of the non-constructivity of the specification is the monotonicity of deductive systems, implying that for every superset A' of the set of assumptions A that supports an argument a attacking another argument b, there exists an argument a' supported by A' that also attacks b. Thus, in general, there can be infinitely many arguments against another argument b. Moreover, for each such attack, there is the need to search among infinitely many candidate counter-attacks to find one that is successful. [2] proposes an alternative notion of argument, that lends itself to effective computation while maintaining correctness. This notion relies upon the use of a *selection function*, which, taken a (multi)set of sentences, returns a sentence in it.

Definition 2.7 Given a selection function:

- A **tight deduction** of a conclusion α is a (possibly infinite) sequence of multisets $S_1, \ldots, S_m, \ldots$, where $S_1 = \{\alpha\}$ and for every $1 \leq i < m$, where σ is the selected sentence occurrence in S_i:
  1. If σ is not an assumption then $S_{i+1} = S_i - \{\sigma\} \cup S$ for some inference rule of the form $\sigma \leftarrow S \in \mathcal{R}$ [2].
  2. If σ is an assumption then $S_{i+1} = S_i$.
- A **tight argument** of a conclusion α based on (or supported by) a set of assumptions A is a finite tight deduction $S_1, \ldots, S_m$ where $S_m = A$.

Tight arguments and ordinary arguments (as given in definition 2.4) are equivalent, as:
- for every tight argument of a conclusion α supported by a set of assumptions A there exists an argument of α supported by A;
- for every argument for a conclusion α supported by a set of assumptions A and for every selection function, there exists a tight argument of α supported by some subset $A' \subseteq A$.

Then, to show that a set of assumptions A is admissible, it suffices to consider only tight attacks against A and tight counter-attacks supported by assumptions in A. Indeed:

Theorem 2.1 A set of assumptions A is admissible if and only if
for every tight argument a that attacks A there exists a tight argument supported by $A' \subseteq A$ that counter-attacks a, and
no $A' \subseteq A$ supports a tight argument that attacks an assumption in A.

This theorem is the basis of the abstract procedure for argumentation via admissibility proposed in [2]. Intuitively, this is based on constructing dispute derivations between two players, the proponent $\mathcal{P}$ and the opponent $\mathcal{O}$. Here, given a sentence α to be proven to be an admissible belief, $\mathcal{P}_i$ intuitively corresponds to a multiset in a tight argument for α or counter-attacking an attack against the argument for α being constructed. Also, $\mathcal{O}_i$ corresponds to a set of multisets, each representing an argument potentially attacking the proponent's arguments. A_i is the set of all assumptions currently needed by the proponent to support its arguments. C_i is the set of all assumptions used by the opponent currently chosen to be counter-attacked by the proponent. Formally:

Definition 2.8 Given a selection function, a **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

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

and for every $0 \leq i < n$, 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

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

[2]We use the same symbols for multiset membership, union etc as for ordinary sets.

(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

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

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 $\sigma \notin A_i$ and $\sigma \notin C_i$ and [3]

(b.1) if $\overline{\sigma}$ is not an assumption, then

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

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

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

(c) or $\sigma \notin A_i$ and $\sigma \in C_i$ [4]

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

(ii) if σ is not an assumption, then

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

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

Then, [2] proves that if there exists a dispute derivation for a sentence, then that sentence is an admissible belief (and the defence set computed by the derivation is admissible). The admissibility semantics is credulous, in that it deems a belief to be admissible whenever there exists *one* admissible set of assumptions supporting one argument for it. There are many applications where a credulous semantics is not appropriate, though. Many sceptical semantics for argumentation could be adopted, including

- the *grounded* semantics [3], defined in terms of all *complete extensions*. Complete extensions are admissible sets of assumptions A containing all assumptions α such that A counter-attacks all attacks against α;
- the *sceptical preferred* semantics [3], defined in terms of all *preferred extensions*, namely maximally admissible sets of assumptions.

[3]In [2], the condition $\sigma \notin C_i$ in case (b) and the case (b.2) were missing. Our new case here provides an additional filtering of culprits by culprits without affecting the correctness of the procedure. Moreover, case (b.2) takes into account the situation in which the contrary of the chosen culprit is an assumption in turn.

[4]In [2], this case (c) was missing. Our new case here provides an additional filtering of culprits by culprits without affecting the correctness of the procedure.

These semantics are sceptical in that they deem a belief to be held only if this belief is "agreed upon" by *all* extensions sanctioned by the semantics. In [10], we give abstract proof procedures for computing the grounded extension and the sceptical preferred semantics of a given assumption-based framework. The procedure for the sceptical preferred semantics works as follows, given a sentence α:

1. determine whether α is an admissible belief, by determining an admissible set Δ supporting an argument for α (this can be achieved by a dispute derivation);
2. let $\mathcal{D}$ be the set of all admissible sets of assumptions attacking Δ; check that, for each element E of $\mathcal{D}$, there exists an admissible set of assumptions $E' \supseteq E$ such that E' supports an argument for α;
3. if all tests at step 2. are successful, then succeed.

This abstract procedure is very expensive in practice, due to the need to compute $\mathcal{D}$ at step 2. In [10] we attempt to optimise the search for $\mathcal{D}$ by considering only tight attacks against Δ, namely by replacing step 2. above by

$2'$ let $\mathcal{T}$ be the set of all tight attacks against Δ; check that, for each element E of $\mathcal{T}$, there exists an admissible set of assumptions $E' \supseteq E$ such that E' supports an argument for α.

However, this optimisation is not correct in general, as shown by the following example.

Example 2.1 Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be the assumption-based framework:

- $\mathcal{L} = \{a, b, c, f, \neg a, \neg b, \neg c, \neg f\}$
- $\mathcal{R}$ consists of

$$\neg a \leftarrow f \qquad \neg a \leftarrow b \qquad \neg b \leftarrow c \qquad \neg c \leftarrow b \qquad \neg f \leftarrow a$$

- $\mathcal{A} = \{a, b, c, f\}$
- $\overline{a} = \neg a, \overline{b} = \neg b, \overline{c} = \neg c, \overline{f} = \neg f$.

It is easy to see that $\neg a$ does not hold in all preferred extensions of $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, as it does not hold in the preferred extension $\{c, a\}$. If we apply the optimised algorithm above, though, this is not detected. Indeed, assume that $\Delta = \{b\}$ at step 1. (it is easy to see that this is admissible). b is attacked by the tight argument supported by the admissible $\{c\}$, thus $\mathcal{T}$ is $\{\{c\}\}$ at step $2'$. Since this set can be extended to the admissible set $\{c, f\}$ in which $\neg a$ holds, the algorithm succeeds, giving an incorrect answer.

This example suggests that the sceptical preferred semantics is hard to compute in general. On the other hand, the grounded semantics is efficiently computable, but may be too sceptical, and thus not useful, in many cases, as illustrated by the following example.

Example 2.2 Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be the assumption-based framework:

- $\mathcal{L} = \{a, b, c, \neg a, \neg b, \neg c\}$
- $\mathcal{R}$ consists of

$$a \leftarrow b \qquad a \leftarrow c \qquad \neg b \leftarrow c \qquad \neg c \leftarrow b$$

- $\mathcal{A} = \{b, c\}$
- $\overline{b} = \neg b, \overline{c} = \neg c$.

There are two complete extensions, $\{b\}$ and $\{c\}$, both supporting a. But the grounded extension is $\{\}$ and does not support a.

In this paper, we consider an alternative sceptical semantics, defined in the next section.

3. The ideal semantics for argumentation

In [1] the ideal sceptical semantics for extended logic programs was introduced, generalising the well-founded semantics. Here, we adopt a similar approach to extend the grounded semantics for argumentation frameworks. Intuitively, the *ideal* sceptical semantics approximates better than the grounded semantics the intersection of all preferred extensions.

Definition 3.1

- An admissible set S of assumptions is **ideal** if and only if it is a subset of every preferred extension.
- A set of assumptions Δ is an **ideal extension** if and only if it is a maximal ideal set of assumptions.
- A **belief** α is **ideal** if and only if there exists an argument for α based on a set of assumptions Δ_0 and Δ_0 is a subset of an ideal extension Δ.

The ideal extension is unique and is a superset of the grounded extension. Thus, the ideal semantics is a good sceptical compromise. Moreover, as we will prove, it can be computed effectively by a simple modification of dispute derivations for admissibility.
In example 2.1, $\neg a$ is not an ideal belief. Consider the following additional example.

Example 3.1 Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be the assumption-based framework:

- $\mathcal{L} = \{a, b, c, d, \neg a, \neg b, \neg c, \neg d\}$
- $\mathcal{R}$ consists of
  $\neg a \leftarrow a \qquad \neg a \leftarrow b \qquad \neg b \leftarrow a \qquad \neg c \leftarrow d \qquad \neg d \leftarrow c$
- $\mathcal{A} = \{a, b, c, d\}$
- $\overline{a} = \neg a$, $\overline{b} = \neg b$, $\overline{c} = \neg c$, $\overline{d} = \neg d$.

There are two preferred extensions of $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$: $\{b, c\}$, $\{b, d\}$. Hence b, and thus $\neg a$, hold in the sceptical preferred semantics. b and $\neg a$ are also ideal beliefs, as $\{b\}$ is the ideal extension. Instead, the grounded extension is empty.

The following results are the basis for our proof procedure for the ideal semantics.

Theorem 3.1 α is an ideal belief if and only if the following conditions are both satisfied:

1. there is an admissible set Δ such that Δ supports an argument for α;
2. there is no admissible set of assumptions S such that S attacks Δ.

Theorem 3.2 α is an ideal belief if and only if the following conditions are both satisfied:

1. there is an admissible set Δ such that Δ supports an argument for α;
2. for each tight argument A attacking Δ there exists no admissible set S such that $S \supseteq A$.

A straightforward implementation of this result, whose correctness follows directly from theorem 3.2, is the following abstract procedure:

Algorithm 3.1 Given a sentence α:

1. Determine whether α is an admissible belief, by determining an admissible set Δ supporting an argument for α (this can be achieved by constructing a dispute derivation for α).

2. For each tight argument A attacking Δ check that there is no admissible sets S such that $S \supseteq A$.
3. If all tests at step 2. are successful, then succeed (α is an ideal belief).

We will use this procedure to provide a computational technique for sceptical argumentation with the ideal semantics, in terms of a form of dispute derivations defined next.

4. IS-Dispute Derivation

Before we introduce a dispute derivation for the ideal semantics (*IS-dispute derivation*) let us give a few new notations.
The notion of dispute derivation in definition 2.8 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 4.1 Let S be a set of sentences in $\mathcal{L}$. By *Fail(S)*, we mean that there exists no dispute derivation for S.

IS-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 tight attacks A in step 2. of procedure 3.1);
- $\mathcal{P}_i, \mathcal{O}_i, A_i, C_i$ are as in ordinary dispute derivations, except that sentences occurring in the multisets in $\mathcal{O}_i$ may be *marked*.

Notation 4.2 Given a set of sentences S:

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

Intuitively, IS-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.

Definition 4.1 Given a selection function, an **IS-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 $0 \leq i < n$, 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 $\mathcal{O}_i$ and σ is selected in S_u then

(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 $\sigma \notin C_i$ and

(b.1) if $\overline{\sigma}$ is not an assumption, then

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

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

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

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

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

(ii) if σ is not an assumption, then

$$\mathcal{P}_{i+1} = \mathcal{P}_i \qquad A_{i+1} = A_i \qquad C_{i+1} = C_i \qquad \mathcal{F}_{i+1} = \mathcal{F}_i$$
$$\mathcal{O}_{i+1} = \mathcal{O}_i - \{S\} \cup \big\{S - \{\sigma\} \cup R \,|\, \sigma \leftarrow R \in \mathcal{R}\big\}$$

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

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

Example 4.1 Consider the assumption-based framework in example 3.1. An IS-dispute derivation for $\neg a$ 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

$$\mathcal{P}_0 = \{\neg a\} \qquad A_0 = \{\} \qquad \mathcal{O}_0 = C_0 = \mathcal{F}_0 = \{\},$$

applying step (1.ii), with the second rule, we have

$$\mathcal{P}_1 = \{b\} \qquad A_1 = \{b\} \qquad \mathcal{O}_1 = C_1 = \mathcal{F}_1 = \{\},$$

applying step (1.i), we have

$$\mathcal{P}_2 = \{\} \qquad A_2 = \{b\} \qquad \mathcal{O}_2 = \{\{\neg b\}\} \qquad C_2 = \mathcal{F}_2 = \{\},$$

applying step (2.ii), we have

$$\mathcal{P}_3 = \{\} \qquad A_3 = \{b\} \qquad \mathcal{O}_3 = \{\{a\}\} \qquad C_3 = \mathcal{F}_3 = \{\},$$

applying step (2.i.b.1), we have

$$\mathcal{P}_4 = \{\neg a\} \qquad A_4 = \{b\} \qquad \mathcal{O}_4 = \{\} \qquad C_4 = \{a\} \qquad \mathcal{F}_4 = \{\{a\}\},$$

applying step (1.ii) using the second rule, we have

$$\mathcal{P}_5 = \{\} \qquad A_5 = \{b\} \qquad \mathcal{O}_5 = \{\} \qquad C_5 = \{a\} \qquad \mathcal{F}_5 = \{\{a\}\},$$

applying step (3), $Fail(\{a\})$ is called (giving rise to a *Fail-dispute derivation* of $\{a\}$, given in example 6.1) and

$$\mathcal{P}_6 = \{\} \qquad A_6 = \{b\} \qquad \mathcal{O}_6 = \{\} \qquad C_6 = \{a\} \qquad \mathcal{F}_6 = \{\},$$

Hence, $\{b\}$ is the computed ideal support for $\neg a$.

5. Soundness of IS-dispute derivations

IS-dispute derivations can be guaranteed to be sound for the ideal semantics if dispute derivation (for the computation of $Fail$) are complete for the admissibility semantics. As discussed in [2], 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 IS-dispute derivations. For simplicity, we will restrict ourselves to the simplified frameworks used throughout the paper for the examples (see page 3).

Notation 5.1 Let $\mathcal{AF}$ be an assumption-based framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$. By $\mathcal{AF}^+$, we will denote the framework obtained by deleting all assumptions appearing in the premises of the inference rules of $\mathcal{R}$.

Below, given $\mathcal{AF}$, we use the notion of *dependency graph* of $\mathcal{AF}^+$, defined in a way similar to the *atom dependency graph* for logic programming (see, e.g. the review in [11]). The dependency graph of $\mathcal{AF}^+$ is a directed graph where:

- the nodes are the atoms occurring in $\mathcal{AF}^+$;
- 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 $\mathcal{AF}^+$ such that q occurs in B.

Definition 5.1 An assumption-based framework $\mathcal{AF}$ is **positively acyclic** (or **p-acyclic** for short) if the dependency graph of $\mathcal{AF}^+$ is acyclic.

Lemma 5.1 Given a p-acyclic framework, there exists no infinite tight deduction.

In the case of p-acyclic frameworks with a finite underlying language $\mathcal{L}$ the dispute derivations of definition 2.8 are complete, in the following sense:

Theorem 5.1 Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be an 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 supports an argument for α.

We can then prove the correctness of IS-dispute derivation, for p-acyclic assumption-based frameworks with a finite underlying language.

Theorem 5.2 Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be p-acyclic with a finite $\mathcal{L}$. Suppose that there exists an IS-dispute derivation for α. Then α is an ideal belief.

6. Computing Fail(S)

$Fail(S)$ at step 3 of IS-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 [2].

Definition 6.1 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

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

and, for every $0 \leq i < n$, 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 S from $\mathcal{O}$ is selected, then
   (a) If $S = \{\}$ then $\mathcal{D}_{i+1} = \mathcal{D}_i - \{Q\}$
   (b) If $S \neq \{\}$ then let $\sigma \in S$ be the selected sentence in S:
      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 2.8;
      ii. if σ is an assumption 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 are obtained from Q as in steps (2.i.b) or (2.i.c) (as applicable) of definition 2.8;
2. If an $\sigma \in \mathcal{P}$ 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 2.8;
   (b) if σ is not an assumption then $\mathcal{D}_{i+1} = \mathcal{D}_i - \{Q\} \cup \{Q' \mid$ there is a rule $\sigma \leftarrow R$ such that Q' is obtained from Q as in step (1.ii) of definition 2.8$\}$.

Theorem 6.1 There exists a Fail-dispute derivation for a multiset of sentences S if and only if there is no dispute derivation for S.

Example 6.1 Consider the assumption-based framework in example 3.1. We show here a Fail-dispute derivation of $\{a\}$.

$\mathcal{D}_0 = \{ \langle \{a\}, \{\}, \{a\}, \{\} \rangle \}$	applying step 2, we have:
$\mathcal{D}_1 = \{ \langle \{\}, \{\{\neg a\}\}, \{a\}, \{\} \rangle \}$	applying step (1.b), we have:
$\mathcal{D}_2 = \{ \langle \{\}, \{\{a\}, \{b\}\}, \{a\}, \{\} \rangle \}$	applying step (1.b) by selecting $S = \{a\}$ in $\{\{a\}, \{b\}\}$ we have:
$\mathcal{D}_3 = \{ \langle \{\}, \{\{\}, \{b\}\}, \{a\}, \{\} \rangle \}$	applying step (1.a) by selecting $S = \{\}$, we have:[5]
$\mathcal{D}_4 = \{\}$.	

7. Conclusions

We have proposed a new proof procedure for computing the ideal semantics for argumentation in assumption-based frameworks, adapted from [1]. We have argued that this

[5] Notice that step (2.i.b) and (2.i.c) are not applicable in this case.

is a good semantics for performing sceptical argumentation, as it is easily computed and is not overly sceptical.

The proof procedure is defined in terms of IS-dispute derivations and Fail-dispute derivations, both adapted from the dispute derivations of [2]. All these derivations extend and generalise standard SLD-based derivations in logic programming, as discussed in [2].

We have proven that it is sound for assumption-based frameworks with a finite underlying language and p-acyclic. In order to prove this soundness result, we have proven a novel completeness result, for p-acyclic frameworks, for the proof procedure proposed in [2].

There are a number of existing tools for computing sceptical argumentation, notably [12], [13] and [14]. These tools are proven to be sound and complete for *coherent* frameworks [9], i.e. frameworks for which the preferred and stable semantics coincide. Instead, our procedure is sound for non-coherent frameworks too, as soon as they are p-acyclic.

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 fellowship.

References

[1] José Júlio Alferes, Phan Minh Dung, and Luís Moniz Pereira. Scenario semantics of extended logic programs. In A. Nerode and L. Pereira, editors, *Proc. 2nd International Workshop on Logic Programming and Non-monotonic Reasoning*, pages 334–348. MIT Press, 1993.

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

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

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

[5] P.M. Dung, P. Mancarella, and F. Toni. A dialectic proof procedure for sceptical assumption-based argumentation. Technical report, AIT, 2006.

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

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

[8] Henry Prakken and Giovanni Sartor. The role of logic in computational models of legal argument: A critical survey. In *Computational Logic: Logic Programming and Beyond*, volume 2408 of *Lecture Notes in Computer Science*, pages 342–381. Springer, 2002.

[9] P.M. Dung. The acceptability of arguments and its fundamental role in non-monotonic reasoning and logic programming and n-person game. *Artificial Intelligence*, 77:321–357, 1995.

[10] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Argumentation-based proof procedures for credulous and sceptical non-monotonic reasoning. In *Computational Logic: Logic Programming and Beyond*, volume 2408 of *Lecture Notes in Computer Science*, pages 289–310. Springer, 2002.

[11] Chitta Baral and Michael Gelfond. Logic programming and knowledge representation. *Journal of Logic Programming*, 19/20:73–148, 1994.

[12] G. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *Proc. JELIA*, pages 224–238. Springer Verlag LNCS 1919, 2000.

[13] P.E. Dunne and T.J.M. Bench-Capon. Coherence in finite argument systems. *Artificial Intelligence*, 149:221–250, 2003.

[14] C. Cayrol, S. Doutre, and J. Mengin. On decision problems related to preferred semantics a of argumentation frameworks. *Journal of Logic and Computation*, 13(3):377–403, 1003.