Journal of Logic and Computation Advance Access published June 26, 2009

Towards a Common Framework for Dialectical Proof Procedures in Abstract Argumentation

PHAN MINH THANG, PHAN MINH DUNG and NGUYEN DUY HUNG,
Department of Computer Science, Asian Institute of Technology, GPO Box 4, Klong Luang, Pathumthani 12120, Thailand.
E-mail: thangfm@ait.ac.th; dung@cs.ait.ac.th; nguyenduy.hung@ait.ac.th

Abstract

We present a common framework for dialectical proof procedures for computing credulous, grounded, ideal and sceptical preferred semantics of abstract argumentation. The framework is based on the notions of dispute derivation and base derivation. Dispute derivation is a dialectical notion first introduced for computing credulous semantics in assumption-based argumentation, and adapted here for computing credulous semantics and grounded semantics. Base derivation is introduced for two purposes: (i) to characterize all preferred extensions containing a given argument, and (ii) to represent backtracking in the search for a dispute derivation. We prove the soundness of the proof procedures for any argumentation frameworks and their completeness for general classes of finitary or finite-branching argumentation frameworks containing the class of finite argumentation frameworks as a subclass. We also discuss related results.

Keywords: Abstract argumentation, dialectical, proof procedure, dispute resolution

1 Introduction

Argumentation is a form of reasoning, that could be viewed as a dispute resolution, in which the agents present their arguments to establish, defend or attack certain propositions. Argumentation provides a basis for understanding non-monotonic and defeasible reasoning [11, 20, 33, 34, 36], a promising platform for investigating decision making, negotiation, legal reasoning, and dispute resolution [20, 25, 26, 38, 43, 45, 51, 52, 60].

Over the years, a number of argumentation systems have been introduced [11, 20, 40, 46, 49, 53, 54, 57]. The most abstract of them is the abstract argumentation framework of [20] defined by a set of 'atomic' arguments together with a binary relation representing the attack relationship between arguments. The semantics of abstract argumentation is based on the notion of *acceptability* of arguments: an argument A is *acceptable* w.r.t. a set S of arguments iff S attacks every argument attacking A. The *preferred* semantics [20] is based on preferred extensions, maximal (under set inclusion) conflict-free sets of arguments that can defend itself against all possible attacks. This semantics is *credulous* since an agent may hold a preferred extension conflicting with the ones held by others. In some applications such as in the legal domain, it is often more appropriate to accept an argument only if it represents a *consensus* among all agents. Several *sceptical* semantics were proposed for this purpose, notably the *sceptical preferred* semantics [20] under which only arguments acceptable w.r.t. *all* preferred extensions are accepted. More easily computable but also more sceptical are the *ideal* semantics [23] and the *grounded* semantics [18, 20, 50], both specify a subset of the set of arguments sceptically preferred accepted as their unique extension. Besides proposals of new semantics to overcome certain shortcomings of the semantics studied in [20], there

doi:10.1093/logcom/exp032

has been an extensive study to extend the abstract argumentation framework in other directions, for example, to deal with new features like preferences [1] and values [7, 8], or to deal with constructing arguments and identifying attacks from a set of rules in an underlying logic [11, 21, 23]. There is a need for a common framework to develop proof procedures for all these extensions.

Viewing ordinary, human debates and discussions as forms of proofs, dialectical proof procedures provide a formal account for informal reasoning in everyday argumentation. Imagine a group of agents whose beliefs are represented by different preferred extensions of some argumentation framework, a proof for the credulous acceptance of a given argument A is seen as the formal counterpart of a discussion about A that requires at least *one* of these agents to support. In contrast, a discussion about A that requires the support from *all* these agents corresponds to a proof for its skeptical acceptance. Dialectical proof procedures could be seen as a formal represention of these discussions. The TPI (*Two party Immediate Response*) procedure of Vreeswijk and Prakken [59] constructs disputes in which two fictitious players called the proponent and the opponent alternate in attacking each other's most recent argument whenever possible, until the opponent lost the dispute because it cannot present anything new. This argument game is shown to be a sound and complete proof for credulous acceptance [29]. Cayrol *et al*. in [14] also presented argument games for credulous acceptance. Their games improve on [59] by performing more filtering in order to finish earlier, and differ from [59] in that search strategies are left open. To our best knowledge, there are no dialectical procedures computing sceptical semantics in all but some cases. In [23], Dung *et al*. developed dialectical procedures computing the ideal and grounded semantics of assumption-based argumentation frameworks based on the dialectical procedure for the credulous semantics in [21]. In [42], Modgil and Caminada presented several argument games for the grounded semantics. The sceptical TPI dispute procedure of [59] and the procedure of [14], both for computing the sceptical preferred semantics, are proven to be sound and complete only in *coherent* argumentation frameworks [14, 29]. However, there are collections of procedures using non-dialectical methods. Vreeswijk in [58] developed a procedure for computing the grounded semantics and all admissible sets containing an examined argument. The procedures of Doutre and Mengin in [16], Cayrol *et al*. in [14], Caminada in [13], Modgil and Caminada in [42], and Verheij in [56] can construct all preferred extensions, and using these sceptical preferred acceptance can be checked. Besides differences in semantics and paradigms, existing proof procedures also differ in their origins. One of the early known dialectical procedures for logic programming was presented in [18]. Later proof procedures for assumption-based argumentation [21, 23] could be viewed as being inspired by both negation as failure approach in logic programming and by the dialectical view in [18].

We present a common framework capable of capturing dialectical proof procedures for computing the credulous, grounded, ideal and sceptical preferred semantics in general cases of abstract argumentation frameworks. The framework consists of only two notions of dispute derivations and base derivations. Dispute derivation is a dialectical notion originally introduced in [21] for computing the credulous semantics in assumption-based argumentation, and adapted here in several variants for abstract argumentation. Base derivation is introduced for two purposes: (i) to characterize all preferred extensions containing a given argument, and (ii) to represent backtracking in the search for a dispute derivation. We prove the soundness of obtained proof procedures for any argumentation frameworks and their completeness for a general class of finitary argumentation frameworks containing the class of finite argumentation frameworks as a subclass.

The structure of the article is as follows. In Section 2, we introduce proof procedures computing the credulous and grounded semantics. In Section 3, we develop proof procedures computing the sceptical preferred and ideal semantics. In Section 4, we present algorithmic issues. We conclude in Section 5.

This article is an extended and improved version of [24]. It extends its predecessor by introducing proof procedures for the grounded and ideal semantics, concrete algorithms for all proof procedures and analysis of related work.

2 A dialectical framework for proof procedures in abstract argumentation

Following [20], we define an argumentation framework as a pair $\mathcal{AF} = (\mathcal{A}, att)$, where $\mathcal{A}$ is a set of arguments, and att is a binary relation over $\mathcal{A}$ representing the attack relation between the arguments ($att \subseteq \mathcal{A} \times \mathcal{A}$). Given two arguments A and B, $(A, B) \in att$ means A attacks B. A set S of arguments attacks an argument A if some argument in S attacks A; S attacks another set S' if S attacks some argument in S'. The definitions of *conflict-free*, *admissible*, *complete* sets and *preferred, grounded, ideal* extensions are recalled from [20] as follows.

Let S be a set of arguments

1. Argument A is *acceptable* with respect to S iff S attacks every argument attacking A.
2. S is *conflict-free* iff it does not attack itself.
3. S is *admissible* iff S is conflict-free and each argument in S is acceptable with respect to S.
4. S is a *preferred extension* iff S is maximally (w.r.t. set inclusion) admissible.
5. S is *ideal* iff it is admissible and contained in every preferred extensions. S is an *ideal extension* iff it is maximally ideal.
6. S is *complete* iff it is admissible and contains every arguments acceptable w.r.t. to S.
7. S is a *grounded extension* iff it is minimally complete.

Let $\mathcal{F}(S) = \{A \in \mathcal{A} \mid \text{A is acceptable w.r.t. } S\}$, the *characteristic function* of $\mathcal{AF}$. The following properties were proven in [20, 23]:

- S is complete iff it is a conflict-free fixed point of $\mathcal{F}$,
- there is a unique grounded extension, which is the least fixed point of $\mathcal{F}$,
- there is a unique ideal extension, which is the union of all ideal sets,
- the ideal extension, the grounded extension and preferred extensions are complete and
- the ideal extension is a super set of the grounded extension and is a subset of the intersection of all preferred extensions.

In ascending order of scepticism, an argument A is said to be:

1. credulously accepted if it is contained in at least one preferred extension,
2. sceptically preferred accepted if it is contained in every preferred extensions,
3. ideally accepted if it is contained in the ideal extension and
4. groundedly accepted if it is contained in the grounded extension.

An overly confident agent may believe in an argument if it is credulously accepted, while an overly sceptical agent may only do so if the argument is groundedly accepted. In between these semantics, other rational agents may believe in ideally or sceptically preferred accepted arguments.

EXAMPLE 1
For the framework $(\mathcal{A}, att)$ borrowed from [23] depicted below,[1] the grounded extension is a proper subset of the ideal extension, which is a proper subset of the intersection of all preferred extensions.

---

[1] For purpose of reference, we often identify an $\mathcal{AF}$ with the graph representing it.

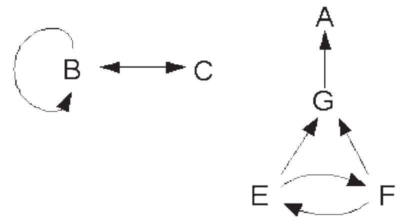

FIGURE 1. The framework of Example 2.1 in [23].

- $\{A, C, E\}$ and $\{A, C, F\}$ are preferred extensions.
- A, C are sceptically preferred accepted.
- $\{C\}$ is the ideal extension.
- $\{\}$ is the grounded extension.

Throughout the article we assume that an implicit argumentation framework $\mathcal{AF} = (\mathcal{A}, att)$ is given and only mention it explicitly if not clear by the context. Given an argument $B \in \mathcal{A}$, $Attack_B$ and $Attacked_B$ denote the set of arguments attacking B, i.e. $Attack_B = \{A \mid (A,B) \in att\}$, and attacked by B, i.e. $Attacked_B = \{A \mid (B,A) \in att\}$.

Argumentation provides a natural framework for representing dispute resolution. In informal terms, a dispute between a *proponent* and an *opponent* is a sequence of alternate moves where at each move a player puts forward an argument attacking arguments made in previous moves by the opposing player. An argumentation strategy of a player specifies when an argument should be presented. An argument is accepted credulously if the proponent has a winning strategy for defending it. Proof procedures have to address three questions:

1. what are the winning argumentation strategies?
2. how to search for such strategies?
3. how to filter both within and across disputes?

In the context of credulous acceptance, the framework of Modgil and Caminada [42] only addresses the first question by specifying what are the disputes won by the proponent, and partially addresses the third by discussing filtering within disputes. It leaves unanswered the second question, is thus not yet ready to be used procedurally. The TPI framework of Vreeswijk and Prakken [59] formalized by Dunne and Bench-Capon [29] mixes winning strategies and the search, and provides some filtering across disputes. The framework of Cayrol *et al.* [14] integrates winning strategies with filtering mechanisms. It leaves open search strategies, making it easier to understand. But it is unclear how these frameworks could be extended to deal with other semantics like grounded, ideal or sceptical preferred. It is also unclear whether they can be implemented in instances of the abstract argumentation framework, like logic programming, assumption-based argumentation [11, 21] or Prakken and Sartor's [49] defeasible argumentation system in which arguments need to be constructed on demand. Our approach provides a common framework for dialectical proof procedures for all these semantics. The framework is geared towards the abstract argumentation as well as its existing instances. It is based on the notions of dispute derivation and base derivation. Dispute derivation provides a means to construct winning argumentation strategies for both the credulous and grounded semantics. Base derivation incorporating different backtracking strategies provides, among others, also means to search for dispute derivations. In this section we present dispute derivation and describe the base derivation in Section 3.

2.1 Dispute trees

To prove a credulous/grounded acceptance, a dispute tree is constructed. Dispute tree is a conceptual tool for illustrating winning argumentation strategies. Under the credulous semantics, the proponent wins if he can attack every attacking argument of the opponent without self-attacking. The grounded semantics requires further requirements, which shall be presented later.

We recall below the definition of dispute tree from [21–23]:

DEFINITION 1
A dispute tree $\mathcal{T}$ for an argument A with respect to an argumentation framework $\mathcal{AF}$ is defined as follows:

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. The status of a child node is different to the status of its parent node. The argument labelling a child node attacks the argument labelling its parent node.
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 attacking B, there exists a child of N labelled by C.
4. For every opponent node N labelled by an argument B, there exists exactly one child of N labelled by an argument attacking B.
5. There are no other nodes except those given by rules 1–4 above.

The set of all arguments labelling the proponent nodes in a dispute tree $\mathcal{T}$ is called the defense set of $\mathcal{T}$.

A branch in a dispute tree may be finite or infinite. A finite branch represents a winning debate (also called dispute) that ends with an argument by the proponent against which the opponent is unable to attack. An infinite branch represents a winning debate in which the proponent counterattacks every attack of the opponent, *ad infinitum*. A dispute tree is finite iff all its nodes have a finite number of children, and all its branches are of finite length.

The definition of dispute tree requires that the proponent must counterattack every attack, but it does not guarantee that the proponent does not attack itself. This further requirement is incorporated in the definition of admissible dispute tree [21, 23].

DEFINITION 2
A dispute tree is said to be admissible if there is no argument that labels both a proponent node and an opponent node.

EXAMPLE 2
Figure 2 depicts an argumentation framework and two admissible dispute trees for argument A (a node is represented by x:Y where Y is the argument labelling the node and x is either p or o denoting its proponent or opponent status).

The following lemma is analogous to Theorem 5.1 of [21] and Theorem 3.2 of [23].

LEMMA 1
1. If $\mathcal{T}$ is an admissible dispute tree for an argument A then the defense set of $\mathcal{T}$ is admissible.
2. Let S be an admissible set of arguments and $A \in S$. Then there exists an admissible dispute tree $\mathcal{T}$ for A such that the defense set of $\mathcal{T}$ is a subset of S.

Similar to Theorem 5.2 of [21], Theorem 3.1 of [23] proves that all finite dispute trees are admissible. We prove further that these are admissible trees for groundedly accepted arguments in the

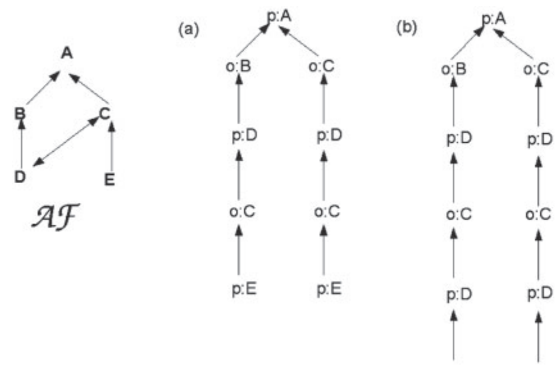

FIGURE 2. Depiction of an argumentation framework and two admissible dispute trees for argument A. (**a**) A finite admissible dispute tree with defense set {A,D,E}. (**b**) An infinite admissible dispute tree with defense set {A,D}.

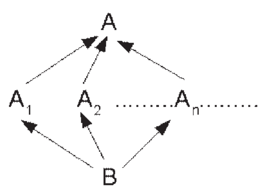

FIGURE 3. An infinite-branching framework.

following theorem. Moreover, the reverse also holds for finite-branching argumentation frameworks, i.e. frameworks where for any argument A, $attack_A$ is finite.

THEOREM 1
An argument A in a finite-branching argumentation framework is groundedly accepted iff there exists a finite dispute tree for A.

The proof of this theorem and of all other results in this section are given in Appendix A.

Theorem 1 underlines the strategies ϕ_{G_1}, ϕ_{G_2}, ϕ_{G_3} of [42], all of which essentially guarantee the finiteness of constructed dispute trees by preventing the proponent from repeating arguments. Since for each of these strategies there is an equivalent finite dispute tree but the reverse does not hold, Theorem 1 is more general than the related results of [43].

In general, the theorem does not hold for any arbitrary argumentation frameworks, as shown by the following example.

EXAMPLE 3
Argument A in the argumentation framework depicted in Figure 3 is groundedly accepted. The unique dispute tree for A is not finite because it is infinite in breath.

As elaborated in [21, 23], admissible dispute trees shorten the distance between proofs for credulous/grounded acceptance and procedures searching for these proofs because they show how to construct incrementally an admissible set in defense of a given, desired argument. However, it is inefficient to construct them faithfully because they may contain repeated redundant segments, or even are non-constructive because they may be infinite as shown by Example 2. For efficiency and completeness, a proof procedure needs concrete filtering mechanisms, like one avoiding re-defense of what has already been defended in order to fold infinite trees into finite trees, or one checking admissibility explicitly in order to reduce their sizes. Indeed, we will incorporate these mechanisms into our notions of dispute derivations, *credulous dispute* derivation and *ground dispute* derivation, which, respectively, provide efficient constructions of dispute trees for proving credulous acceptance and grounded acceptance. But first in the following, we present their preliminary form, *simple dispute* derivation, seen here as an initial step between dispute trees and our final dispute derivations, because it is correct, but highly inefficient for grounded acceptance, and not complete for credulous acceptance. It, however, presents an important feature of our framework for linking with practical argument-based systems. Our dispute derivations are defined in a spirit like the dispute derivation in [21, 23] (that are themself inspired by proof procedures in logic programming like SLDNF, the 'EK' procedure of [17, 19, 30], the 'KT' procedure of [37, 55]), the dialectical proof procedure for the grounded semantics in [18] and the argument games in [14, 29, 35, 47, 48].

2.2 Simple dispute derivations

A simple dispute derivation is a top-down construction of a dispute tree by a sequence of pairs $\langle P_i, O_i \rangle$, where P_i is a set of arguments presented by the proponent but not yet attacked by the opponent, and O_i is a set of arguments presented by the opponent but not yet counterattacked by the proponent. Trees emerging at any step i but the last one are called partial dispute trees, because they are like dispute trees with the exception that conditions 3, 4 of Definition 1 are not applied to their frontier nodes (i.e. nodes labelled by arguments in P_i or O_i). The initial partial dispute tree (i.e. $i=0$) consists of only one node labelled by the initial argument. Partial dispute trees are expanded stepwise into the final dispute tree. Each step represents the expansion of a partial dispute tree by expanding a frontier node labelled by argument B by a child node labelled by an argument in $Attack_B$ (if $B \in O_i$) or by a set of child nodes (if $B \in P_i$) labelled by arguments in $Attack_B$ (note that arguments are allowed to be repeated since there are no filtering mechanisms preventing so). Different selections give rise to different derivations, but do not affect completeness, because they simply represent different ways to construct the same dispute tree.

DEFINITION 3
Given a selection function, a partial simple dispute derivation for an argument A is a possibly infinite sequence of pairs $\langle P_0, O_0 \rangle \ldots \langle P_n, O_n \rangle \ldots$, where:

1. P_i, O_i are argument sets
2. $P_0 = \{A\}, O_0 = \emptyset$
3. Let B be the argument selected at step i

   (a) If $B \in P_i$ then
   $P_{i+1} = P_i \setminus \{B\}$
   $O_{i+1} = O_i \cup Attack_B$

   (b) If $B \in O_i$ then there exists some argument $C \in Attack_B$ such that
   $P_{i+1} = P_i \cup \{C\}$
   $O_{i+1} = O_i \setminus \{B\}$

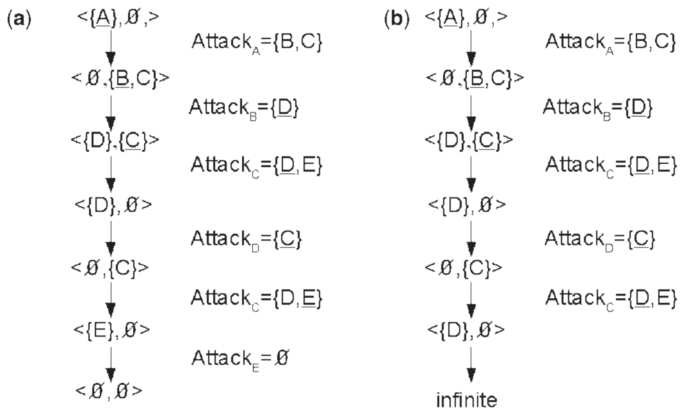

FIGURE 4. (**a**) A finite simple dispute derivation containing a wasteful re-defeat of C. (**b**) An infinite simple dispute derivation.

A full simple dispute derivation is a finite partial simple dispute derivation ended by pair $\langle \emptyset, \emptyset \rangle$.

The following theorem states that full simple dispute derivations represent sound and complete proofs for grounded acceptance.

THEOREM 2

1. An argument A is groundedly accepted if there is a full simple dispute derivation for it.
2. If an argument A is groundedly accepted in a finite-branching argumentation framework, then there is a full simple dispute derivation for it.

Simple dispute derivations not only answer the question of what are winning argumentation strategies, but also address partially the problem of filtering across disputes by collecting arguments not attacked yet in sets P_i and O_i instead of multi-sets.[2] Thus, a proof procedure based on simple dispute derivations for grounded acceptance needs to specify only a search strategy. Such a procedure clearly improves on procedures that faithfully construct dispute trees. However, it is still highly inefficient, and even may not terminate since simple dispute derivations may be redundant or infinite, as shown by the following example.

EXAMPLE 4 (Continue Example 2)

– The infinite simple dispute derivation in Figure 4b constructs the infinite tree in Figure 2b. There is no full simple dispute derivation for this infinite tree.
– In the full simple dispute derivation depicted in Figure 4a constructing the finite tree in Figure 2a, argument C is defeated twice. This is wasteful, although still better than the case of faithfully constructing the tree with three times of defeating C.

Simple dispute derivations present an important feature of our framework: to attack a proponent's argument selected in some step, the opponent puts forward all possible arguments simultaneously.

[2]An argument can have several occurrences in a multi-set since several nodes of a dispute tree can be labelled by the same argument.

Other existing frameworks, e.g. those of Cayrol *et al*. [14], allow only one argument in such a step; thus, the opponent needs many steps not necessarily consecutively to attack the same argument. In practical instances of the abstract argumentation where arguments need to be constructed, like logic programming, the assumption-based argumentation [11, 21] or the framework by Prakken and Sartor [49], generating all arguments attacking some given argument at once is much more efficient than generating them separately at different steps. This is because interleaving their construction makes it difficult to reuse their shared segments. For example, consider a logic program consisting of the following rules: $p \leftarrow q; q \leftarrow r; r \leftarrow nots; r \leftarrow nott$; and $g \leftarrow notp$ where $nots, nott$ and $notp$ are assumptions. The argument supporting g is represented by a backward deduction $\{g\}\{notp\}$ and denoted by α. There are two arguments attacking α, represented by backward deductions $\{p\}\{q\}\{r\}\{nots\}$ and $\{p\}\{q\}\{r\}\{nott\}$ sharing the initial segment, which will be constructed twice if there is no reuse. Allowing the opponent to attack simultaneously avoids the need to plan its attacks.

2.3 Credulous dispute derivations

A credulous dispute derivation is a top-down construction of a possibly infinite admissible dispute tree by a sequence of tuples $\langle P_i, O_i, SP_i, SO_i \rangle$, where P_i and O_i are defined exactly as in simple dispute derivations. Unlike simple dispute derivations, repeated arguments are distinguished from fresh arguments by two new components, SP_i denoting the set of arguments presented by proponent up to step i (so $P_i \subseteq SP_i$); and SO_i denoting the set of arguments presented by the opponent and already counterattacked by the proponent so far (so the opponent has presented $O_i \cup SO_i$). With regard to the partial dispute tree of step i, P_i (resp. O_i) is the set of arguments labelling its proponent (resp. opponent) frontier nodes; SP_i is the set of arguments labelling all its proponent nodes; and SO_i is the set of arguments labelling its opponent internal nodes. This richer representation of the tree makes it possible to apply a series of filtering mechanisms. Each step represents the expansion of its partial dispute tree by a selection of a frontier node labelled by argument B and its replacement by *some* of its children (because it is unnecessary to reconsider children already attacked). Furthermore, there are two conditions to ensure that the set of proponent's arguments is conflict-free: (i) if $B \in P_i$ then $Attack_B \cap SP_i = \emptyset$ (argument B of the proponent is not attacked by his other arguments in SP_i); (ii) if $B \in O_i$ then the proponent does not use arguments in $(SO_i \cup O_i)$ to avoid later conflicts in SP_j, j > i.

To prove the credulous acceptance of an argument A (or generally a set S of arguments), the first tuple $\langle P_0, O_0, SP_0, SO_0 \rangle$ is set to $\langle \{A\}, \emptyset, \{A\}, \emptyset \rangle$ (or $\langle S, \emptyset, S, \emptyset \rangle$, resp.) because the proponent starts the dispute by putting forwards A (or S, resp.). However, in some cases we also want to answer the *credulous refutation* question of 'Can the proponent admissibly attack a set O of arguments proposed by the opponent?' by setting $\langle P_0, O_0, SP_0, SO_0 \rangle$ to $\langle \emptyset, O, \emptyset, \emptyset \rangle$.

DEFINITION 4
Given a selection function, a partial credulous dispute derivation is a possibly infinite sequence of tuples $\langle P_0, O_0, SP_0, SO_0 \rangle \ldots \langle P_n, O_n, SP_n, SO_n \rangle \ldots$ where:

1. P_i, O_i, SP_i and SO_i are argument sets
2. $\langle P_0, O_0, SP_0, SO_0 \rangle$ is one of the forms

   (a) $\langle S, \emptyset, S, \emptyset \rangle$ for some non-empty set $S \subseteq \mathcal{A}$ or
   (b) $\langle \emptyset, O, \emptyset, \emptyset \rangle$ for some non-empty set $O \subseteq \mathcal{A}$.

3. Let B be the argument selected at step i

(a) If $B \in P_i$ and $Attack_B \cap SP_i = \emptyset$ then
$P_{i+1} = P_i \setminus \{B\}$
$O_{i+1} = O_i \cup (Attack_B \setminus SO_i)$
$SP_{i+1} = SP_i$
$SO_{i+1} = SO_i$

(b) If $B \in O_i$ then there exists some argument $C \in Attack_B \setminus (SO_i \cup O_i)$ such that
$P_{i+1} = P_i \cup \{C\}$ if C $\notin SP_i$, otherwise $P_{i+1} = P_i$
$O_{i+1} = O_i \setminus Attacked_C$
$SP_{i+1} = SP_i \cup \{C\}$
$SO_{i+1} = SO_i \cup (Attacked_C \cap O_i)$ (Note that $B \in Attacked_C \cap O_i$)

In Step 3a, it is required that $Attack_B \cap SP_i = \emptyset$ since otherwise the proponent arguments would be conflicting. The expression $O_{i+1} = O_i \cup (Attack_B \setminus SO_i)$ excludes arguments belonging to SO_i from O_{i+1}, thus every opponent's argument is counterattacked only once. The condition $C \in Attack_B \setminus (SO_i \cup O_i)$ of Step 3b allows the proponent to select an argument among possibly many arguments in $Attack_B$ to counterattack B while keeping the dispute tree admissible. Also in Step 3b, $O_{i+1} = O_i \setminus Attacked_C$ because the proponent does not need to counterattack those arguments in O_i that are attacked by C. These filtering mechanisms effectively reduce the cost of constructing the implicit dispute tree without affecting its correctness under the credulous semantics. We define several restricted forms of partial credulous dispute derivations with meaningful input–output behaviours.

DEFINITION 5

1. A full credulous dispute derivation is a finite partial credulous dispute derivation:
$\langle P_0, O_0, SP_0, SO_0\rangle \ldots \langle P_n, O_n, SP_n, SO_n\rangle$
such that $P_n = O_n = \emptyset$. SP_n is called the defense set of the dispute derivation.
2. A credulous dispute derivation for an argument A is a full credulous dispute derivation:
$\langle P_0, O_0, SP_0, SO_0\rangle \ldots \langle P_n, O_n, SP_n, SO_n\rangle$
such that $\langle P_0, O_0, SP_0, SO_0\rangle = \langle \{A\}, \emptyset, \{A\}, \emptyset\rangle$.
3. A credulous dispute derivation against a set O of arguments is a full credulous dispute derivation:
$\langle P_0, O_0, SP_0, SO_0\rangle \ldots \langle P_n, O_n, SP_n, SO_n\rangle$
such that $\langle P_0, O_0, SP_0, SO_0\rangle = \langle \emptyset, O, \emptyset, \emptyset\rangle$

EXAMPLE 5 (Continue Example 2)
A construction of a credulous dispute derivation for A is presented in Figure 5, where the notation $\underline{\text{X}}$ means that X is selected. It shows that A is credulously accepted and the set {A, D} is admissible. Let $t_i = \langle P_i, O_i, SP_i, SO_i\rangle$. Filtering $O_3 = O_2 \cup (Attack_D \setminus SO_2)$ in the transition from t_2 to t_3 prevents the opponent from presenting argument C again, thus folding the infinite dispute tree (on the right) into this finite derivation.

Since our purpose is to develop effective proof procedures, we want to restrict our attention to dispute derivations of finite length. Thus the following result is interesting.

THEOREM 3
There is no infinite partial credulous dispute derivation for finite argumentation frameworks.

Theorem 3 follows from the fact that in the Definition 4, SO component remains unchanged by Step 3a but monotonically increases by Step 3b; and any selection function must interleave between Step 3a and 3b (as step 3a cannot be applied infinitely because it reduces the P component). So the

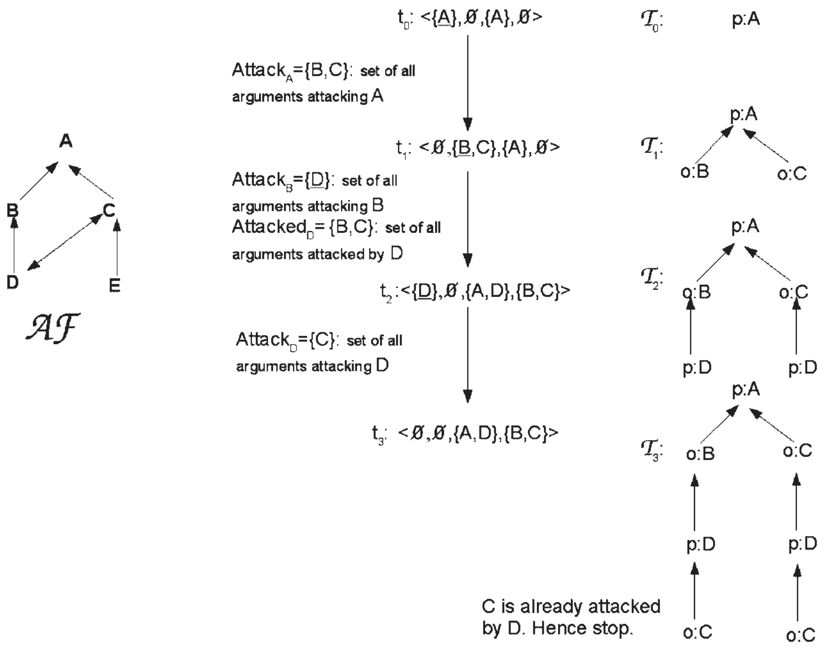

FIGURE 5. A credulous dispute derivation for A.

FIGURE 6. An infinite framework.

number of steps of any dispute derivation is finite if there is a finite upper bound for SO. Naturally for finite argumentation frameworks, this is the set of all arguments.

The proof procedures given in Definitions 4 and 5 are sound and complete not only for finite argumentation frameworks, but also for finitary argumentation frameworks that could be infinite. Consider the infinite argumentation framework in Figure 6. This framework admits a unique preferred extension consisting of arguments $A_0, A_2, \ldots, A_{2n}, \ldots$. It is obvious that for each argument A_{2n} there is a credulous dispute derivation for A_{2n} because all directed paths to A_{2n} are finite. In the following, we introduce the class of finitary argumentation frameworks generalizing this property.

Let $\mathcal{AF}$=($\mathcal{A}$, att) and $A \in \mathcal{A}$. The environment of A denoted by ENV_A is the set of all arguments B in $\mathcal{A}$ such that there is a directed path from B to A in the graph of $\mathcal{AF}$ (i.e. there is a sequence $B_1, B_2, \ldots, B_n$ such that B_i attacks B_{i+1} and $B = B_1$ and $A = B_n$). Let $\mathcal{AF}_A = (ENV_A, att_A)$, where att_A is the restriction of att to ENV_A.

DEFINITION 6
An argumentation framework is said to be *finitary* if for each argument A, $\mathcal{AF}_A$ is finite.[3]

LEMMA 2
1. If S is an admissible set of arguments in $\mathcal{AF}$, then for any argument A, $S \cap ENV_A$ is also admissible in both $\mathcal{AF}$ and $\mathcal{AF}_A$.

[3]It is not difficult to see that finitary argumentation frameworks are finite branching. But the reverse does not hold.

2. If $S \subseteq ENV_A$ is an admissible set in $\mathcal{AF}_A$ for some argument A, then S is also admissible in $\mathcal{AF}$.

From Lemma 2, it is obvious that:

COROLLARY 1
A is credulously accepted in $\mathcal{AF}$ iff A is credulously accepted in $\mathcal{AF}_A$.

A generalized version of Theorem 3 is as follows.

THEOREM 4
There is no infinite partial credulous dispute derivation for finitary argumentation frameworks if the sets in the initial tuple are finite.

The following theorem establishes the soundness and completeness of credulous dispute derivations for finitary argumentation frameworks.

THEOREM 5
1. Suppose $\langle P_0, O_0, SP_0, SO_0\rangle \ldots \langle P_n, O_n, SP_n, SO_n\rangle$ with $P_0 = \{A\}$ is a credulous dispute derivation for an argument A. Then SP_n is admissible and $A \in SP_n$.
2. Let $\mathcal{AF}$ be a finitary argumentation framework, and let A be an argument of $\mathcal{AF}$. If A belongs to an admissible set S, then for any selection function there is a credulous dispute derivation for A, whose defense set is a subset of S.

Similarly, credulous dispute derivation against a set of arguments is a sound and complete proof procedure for credulous refutation.

THEOREM 6
1. Suppose $\langle P_0, O_0, SP_0, SO_0\rangle \ldots \langle P_n, O_n, SP_n, SO_n\rangle$ with $O_0 = O$ is a credulous dispute derivation against a set O of arguments. Then SP_n is an admissible set attacking every elements of O.
2. Let $\mathcal{AF}$ be a finitary argumentation framework, and let O be a finite set of arguments of $\mathcal{AF}$. If there is an admissible set S attacking every elements of O, then for any selection function there is a credulous dispute derivation against O, whose defense set is a subset of S.

The *AB*-dispute derivation of Dung *et al.* [23] for computing admissible beliefs, which improves on the dispute derivation developed by Dung *et al.* in [21], could be viewed as an adaptation of our credulous dispute derivation to assumption-based argumentation frameworks [11].

A variant of credulous dispute derivations with weaker filtering drops all filtering at Step 3b of Definition 4, i.e. changes it to 'there exists some argument C such that $P_{i+1} = P_i \cup \{C\}; O_{i+1} = O_i \setminus \{B\}; SP_{i+1} = SP_i \cup \{C\}; SO_{i+1} = SO_i \cup \{B\}$'. In finitary frameworks, proof procedures based on this variant always terminate since Theorem 4 still holds.

A variant with stronger filtering modifies Definition 4 as follows. In 3a, $O_{i+1} = O_i \cup (Attack_B \setminus X)$, where X contains arguments attacked by SP_i. In 3b, $C \in Attack_B \setminus (SP_i \cup X \cup Y \cup Ref)$, where Y contains arguments attacking SP_i and Ref contains self-attacking arguments (note that SO_i and $SO_i \cup O_i$, respectively, approximate X and Y). These mechanisms, respectively, implement two rules: (i) the opponents are not allowed to present arguments that are attacked by arguments already presented by the proponent; and (ii) the proponents are not allowed to repeat or present arguments that induce conflicts between his arguments. This variant captures the filtering in the sequential argument game ϕ_1 of Cayrol *et al.* [14]. Though it is conceptually attractive, it is not clear how to compute sets X and Y in contexts where arguments are not given explicitly and need to be constructed. In contrast, the filtering in our framework does not have this problem since it involves only arguments already identified and cached in the data structure.

Filtering may improve efficiency. But stronger filtering does not necessarily lead to more efficiency since filtering may require extra data structure and hence may slow down the main computation. In logic programming, for example, Prolog does not have any filtering at all, but is efficient. But here termination is pushed towards users. In otherwords, users are responsible to 'filter' out the redundancies in their programs.

Instead of constructing dispute trees directly, other dialectical proof procedures may construct them via argument games. In informal terms, an argument game consists of disputes concerning a common argument. Different procedures differ mainly in their rules specifying legal moves in a dispute or capabilities of players to open a new dispute within a game. In TPI-dispute of Vreeswijk and Prakken [59], a legal move presents an argument attacking the argument presented in the most recent move. Thus its disputes correspond to branches in our dispute trees, and its games correspond to depth-first expansions of dispute trees. Since disputes interleave, TPI mechanisms for filtering across disputes are not as comprehensive as those in our credulous dispute derivations which, like their predecessors, integrate winning strategies and filtering mechanisms both within and across disputes. Moreover, while in TPI-dispute the search strategy is mixed with winning strategies, it is excluded from our credulous dispute derivation and modelled later by the notion of base derivation. In this aspect, our credulous dispute derivations are closer to the argument games of Cayrol *et al.* [14], which are based on the dialectical framework of Jakobovits and Vermeir [35]. In particular, their ϕ_1 game is a sequentialization of our variant. As discussed at the end of Section 2.2, such sequentialization is impractical for many argument-based systems where arguments have to be generated.

2.4 Ground dispute derivations

Theorem 1 characterizes grounded acceptance in terms of finite dispute tree. Theorem 2 reduces the problem of search for a finite dispute tree to the problem of searching for a full simple dispute derivations; unfortunately, as shown by Example 4 simple dispute derivations may be infinite, resulting in a possible non-termination proof procedure. On the other hand, credulous dispute derivations are complete (at least for finitary argumentation frameworks in which they are always finite) and sound for credulous acceptance, but in general cannot be used for grounded acceptance as shown below.

EXAMPLE 6
In the argumentation framework in Figure 7, it would be incorrect to infer from the depicted credulous dispute derivation that A is groundedly accepted as the grounded extension is empty. Note that the dispute tree for A is infinite.

Credulous dispute derivation is not sound for grounded acceptance. Since all dispute trees of an argument that is credulously accepted but not groundedly accepted are infinite (Theorem 1), we could detect this situation by checking if constructed dispute trees are infinite. This motivates us to introduce the following notion of non-redundant dispute trees.

DEFINITION 7
Given an admissible dispute tree $\mathcal{T}$, the graph $\mathcal{G}_{\mathcal{T}} = \{(A,B) \mid \text{there is a node labelled by } B \text{ with a child node labelled by } A \text{ in } \mathcal{T}\}$ is said to be induced by $\mathcal{T}$. $\mathcal{T}$ is said to be non-redundant iff $\mathcal{G}_{\mathcal{T}}$ is acyclic[4]. $\mathcal{T}$ is redundant if it is not non-redundant.

[4]There is no finite sequence of arguments $A_1, A_2, \ldots, A_n$ such that $(A_{i+1}, A_i) \in \mathcal{G}_{\mathcal{T}}, 1 \le i \le (n-1)$ and $A_1 = A_n$.

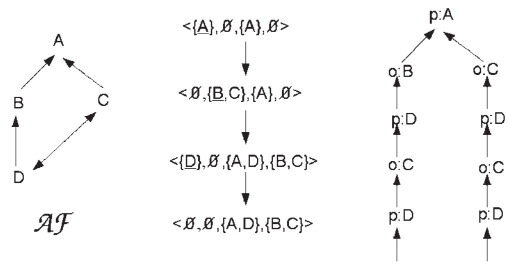

FIGURE 7. A credulous dispute derivation constructs an infinite dispute tree for A.

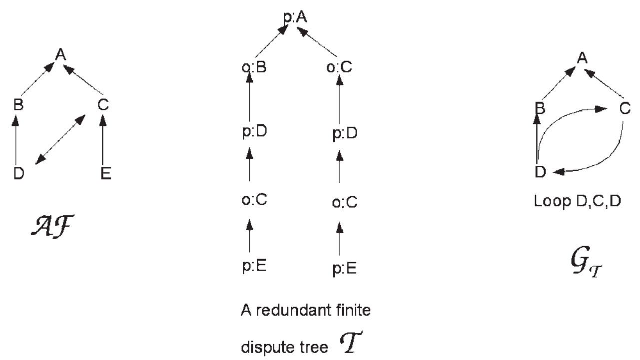

FIGURE 8. A redundant finite dispute tree $\mathcal{T}$.

It is easy to see that:

LEMMA 3
For finitary argumentation framework, infinite dispute trees are redundant.

It follows by contraposition that non-redundant dispute trees are all finite. The dispute tree in Figure 8 shows that the set of non-redundant dispute trees is a proper subset of the set of finite dispute trees. Non-redundant dispute trees still represent complete proofs for grounded acceptance for not only finitary but also finite-branching argumentation frameworks, as shown by the following lemma.

LEMMA 4
An argument A in a finite-branching argumentation framework is groundedly accepted iff there exists a non-redundant dispute tree for A.

Thus, Lemma 4 provides a method to modify credulous dispute derivation for grounded acceptance. This modification requires more than an incorporation of a check whether the induced graphs are acyclic since even credulous dispute derivation with such a check may not find non-redundant dispute trees although such trees exist. To illustrate this, suppose that Definition 4 is modified by inserting at the end of steps 3a and 3b, a check to ensure that the induced graph $\mathcal{G}_{\mathcal{T}}$ is acyclic. Consider the credulous dispute derivation in Example 5 recalled in Figure 9. It is easy to see that $\mathcal{T}_3$ is

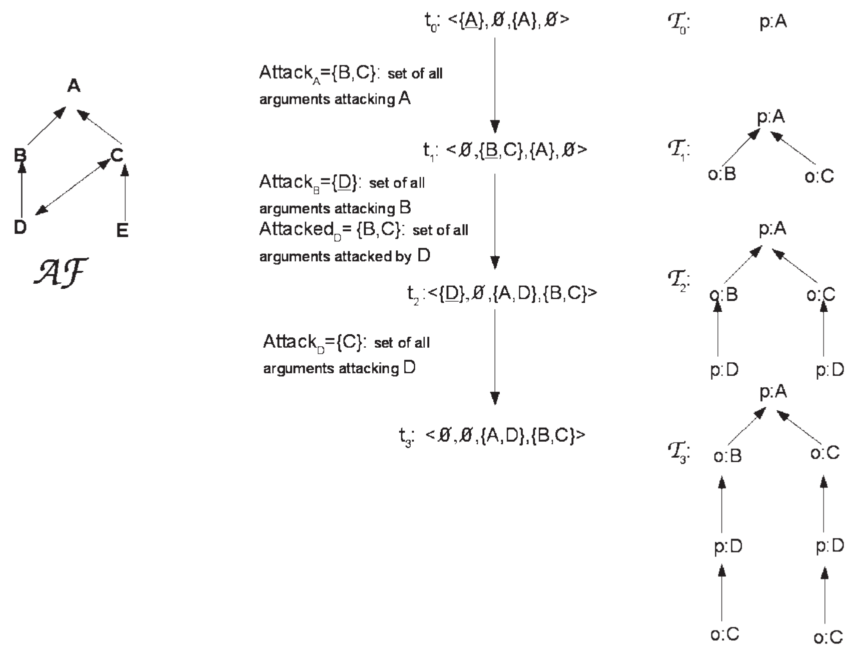

FIGURE 9. A credulous dispute derivation for A.

redundant as there is a loop between C and D. In fact, for any credulous dispute derivation whose selection function selects B instead C from t_1 like this one, the selection of D to counterattack B defeats C as well. This *unintended* defeat of C is due to the filtering at Step 3b of Definition 4: $O_{i+1} = O_i \setminus Attacked_D = O_i \setminus \{B,C\}$. While this defeat is good for credulous acceptance because it keeps the admissible set under construction small, it introduces a loop between D and C. Inserting a check for acyclicity will turn this derivation into a failure. In otherwords, the completeness of the procedure depends on the selection function, a bad feature. Thus, credulous dispute derivation needs to be modified at two points: (i) inserting a check for acyclicity; (ii) modifying filtering mechanisms to avoid introducing loops in induced graphs. Among those in our credulous dispute derivation, the filtering mechanism just observed is the only one that influences the construction of induced graphs. So it is the only one that needs modifying for our purpose.

DEFINITION 8
Given a selection function, a partial ground dispute derivation for an argument A in an argumentation framework $\mathcal{AF} = (\mathcal{A}, att)$ is a possibly infinite sequence of tuples
$\langle P_0, O_0, SP_0, SO_0, G_0\rangle, \ldots, \langle P_n, O_n, SP_n, SO_n, G_n\rangle \ldots,$
where:

1. P_i, O_i, SP_i, SO_i are argument sets, $G_i \subseteq \mathcal{A} \times \mathcal{A}$
2. $P_0 = SP_0 = \{A\},\ SO_0 = O_0 = G_0 = \emptyset$
3. Let B be the argument selected at step i

   (a) If $B \in P_i$ *and* $Attack_B \cap SP_i = \emptyset$ then
   $P_{i+1} = P_i \setminus \{B\}$
   $O_{i+1} = O_i \cup (Attack_B \setminus SO_i)$

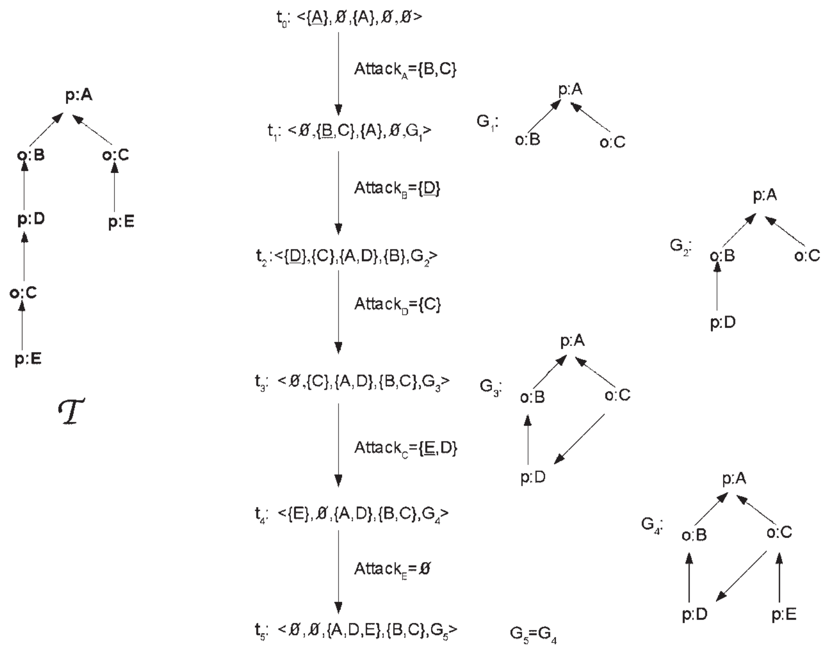

FIGURE 10. A ground dispute derivation for A.

$SP_{i+1} = SP_i$
$SO_{i+1} = SO_i$
$G_{i+1} = G_i \cup \{(C,B) \mid C \in Attack_B\}$ and G_{i+1} is acyclic

(b) If $B \in O_i$ then there exists some argument $C \in Attack_B \setminus (SO_i \cup O_i)$ such that
$P_{i+1} = P_i \cup \{C\}$ if $C \notin SP_i$, otherwise $P_{i+1} = P_i$
$O_{i+1} = O_i \setminus \{B\}$[5]
$SP_{i+1} = SP_i \cup \{C\}$
$SO_{i+1} = SO_i \cup \{B\}$
$G_{i+1} = G_i \cup \{(C,B)\}$ and G_{i+1} is acyclic.

A *full* ground dispute derivation for an argument A is a finite partial ground dispute derivation for A ended with tuple $\langle \emptyset, \emptyset, SP_n, SO_n, G_n \rangle$.

EXAMPLE 7 (Continue Example 5)
Figure 10 depicts a ground dispute derivation for argument A which constructs a non-redundant dispute tree by using the same selection function as in Example 5. Note that at t_3, $attack_C = \{E, D\}$, but selecting D would produce a redundant dispute tree, i.e. G_4 would be non-acyclic.

We obtain results similar to those in credulous dispute derivation.

THEOREM 7
1. Suppose $\langle P_0, O_0, SP_0, SO_0, G_0 \rangle \ldots \langle P_n, O_n, SP_n, SO_n, G_n \rangle$ with $P_0 = \{A\}$ is a full ground dispute derivation for A. Then SP_n is an admissible subset of the grounded extension and $A \in SP_n$.

[5]Note that here $Attacked_C$ is replaced by $\{B\}$ to exclude all *unintended* defeats.

2. Let $\mathcal{AF}$ be a finitary argumentation framework, and let A be an argument of $\mathcal{AF}$. If A belongs to the grounded extension of $\mathcal{AF}$, then there exists a full ground dispute derivation for A.

THEOREM 8
There is no infinite partial ground dispute derivation for finitary argumentation frameworks.

Our ground dispute derivation improves on the *GB*-dispute derivation of [23] computing grounded beliefs in assumption-based argumentation frameworks, by incorporating two more filtering mechanisms $O_{i+1} = O_i \cup (Attack_B \setminus SO_i)$ and $P_{i+1} = P_i \cup \{C\}$ if $C \notin SP_i$ (in 3a and 3b, respectively) to construct non-redundant dispute trees instead of finite dispute trees. As a result, our ground dispute derivation is complete for finitary argumentation frameworks, while the *GB*-dispute derivation is not so. For example, for an assumption-based argumentation framework consisting of two inference rules $\overline{p} \leftarrow q; \overline{q} \leftarrow p$ which corresponds to an argumentation framework with two arguments attacking each other, *GB*-dispute derivation is trapped in a loop while our ground dispute derivation can break it.

Argument games for grounded acceptance ϕ_{G_1}, ϕ_{G_2} and ϕ_{G_3} of [42] guarantee the finiteness of constructed dispute trees by preventing the proponent from repeating arguments. Like in TPI-disputes [29, 59], players in these games follow up in attacking each other's most recent argument. A new dispute is opened only if the existing one is *over*. Thus filtering across them is not possible. The proponent may also have wasteful re-attacks within a dispute since the opponent can repeat arguments already defeated. These games correspond to Definition 8 with weaker filtering.

The intuition of component G in our ground dispute derivations is that in order to prevent filtering from inducing unsoundness, it is necessary to examine the internal of disputes, not just their frontiers like in the first two versions of credulous dispute derivations. Further, it shows that filtering may affect the semantics to be captured. For credulous acceptance, filtering could be as comprehensive as possible, like the filtering of argument game ϕ_1 involves all attacks relating to presented arguments. But for grounded acceptance, it is necessary to exclude unintended defeats following the spirit of our ground dispute derivation.

3 Computing sceptical semantics

This section presents proof procedures for computing the ideal and sceptical preferred semantics by introducing the notion of base derivation.

3.1 Computing sceptical-preferred semantics

An argument A is sceptically preferred accepted iff every preferred extensions contains A. This motivates the following definitions, recalled from [24], of a base of an argument and the completion of such a base.

DEFINITION 9
Let A be an argument, and let $\mathcal{B}$ be a set of admissible sets of arguments such that each element of $\mathcal{B}$ contains A.

1. If for each preferred extension E such that A$\in E$, there exists an admissible set $S \in \mathcal{B}$ such that $S \subseteq E$ *then* $\mathcal{B}$ is called a *base* of A.
2. A base $\mathcal{B}$ of A is said to be *complete* if for each preferred extension E, there is a set $S \in \mathcal{B}$ such that $S \subseteq E$

LEMMA 5 (Sceptical lemma)
An argument A is sceptically accepted iff there exists a complete base $\mathcal{B}$ of A.

The proof of this lemma and of all other results in this section are given in Appendix B.

Thus the sceptical lemma suggests that a proof procedure for showing that A is sceptically preferred accepted, could proceed in two steps:

1. generate a base $\mathcal{B}$ for A and
2. verify that $\mathcal{B}$ is complete base of A.

3.1.1 Generating a base

The following definition of base derivation shows how to construct such a base incrementally.

DEFINITION 10
Given a selection function, a partial base derivation is a possibly infinite sequence $T_0, T_1, \ldots, T_n, \ldots$ where

1. T_i is a set of tuples of the form $\langle P, O, SP, SO\rangle$ defined as in Definition 4 Step 1.
2. For every i, one tuple $t = \langle P, O, SP, SO\rangle$ is selected from T_i and one argument B is selected from P or O.
   (a) If B is selected from P, then: $T_{i+1} = (T_i \setminus \{t\}) \cup \{t'\}$ where t' is computed from t as in Definition 4 Step 3a.
   (b) If B is selected from O then: $T_{i+1} = (T_i \setminus \{t\}) \cup \{t' \mid t'$ is computed from t as in Definition 4 Step 3b for some argument $C \in Attack_B \setminus (SO \cup O)\}$.

Like dispute derivations, we define some restricted forms of partial base derivations (in Definitions 11 and 13). Definition 11 presents two key uses of base derivation set out in the abstract and introduction: a successful base derivation represents backtracking in the search for a dispute derivation of a given argument, and a full base derivation characterizes all preferred extensions containing it.

DEFINITION 11
- A *partial* base derivation for an argument A is a partial base derivation $T_0, T_1, ..., T_n, ...$ such that $T_0 = \{\langle\{A\}, \emptyset, \{A\}, \emptyset\rangle\}$.
- A *successful* base derivation for an argument A is a finite partial base derivation $T_0, T_1, ..., T_n$ for A such that T_n contains at least one tuple of the form $\langle\emptyset, \emptyset, SP, SO\rangle$.
- A *full* base derivation for an argument A is a finite partial base derivation $T_0, T_1, ..., T_n$ for A such that T_n contains only tuples of the form $\langle\emptyset, \emptyset, SP, SO\rangle$.[6]

It follows immediately from Theorem 4.

THEOREM 9
For finitary argumentation frameworks, there is no infinite partial base derivation if the sets in the tuple of T_0 are finite and T_0 is finite.

Theorem 5 reduces the credulous acceptance problem to the problem of searching for a credulous dispute derivation. Part 1 of the following theorem reduces this search to the problem of constructing

[6]Note that T_n could be empty.

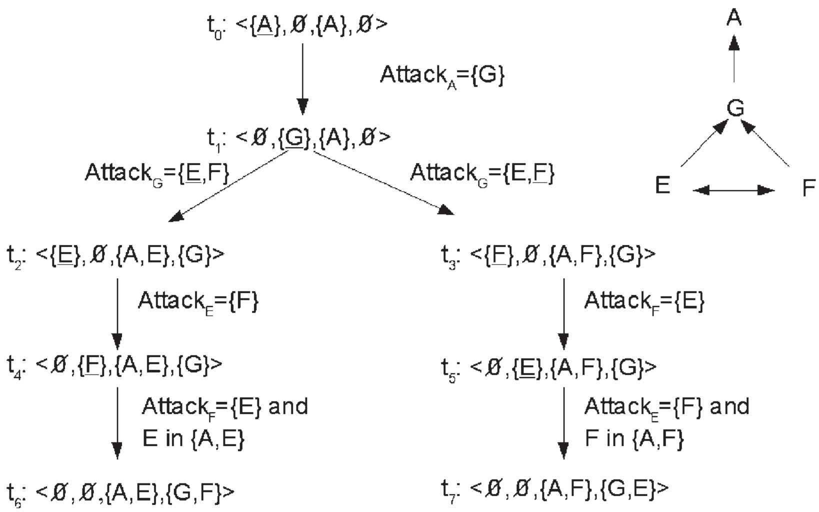

FIGURE 11. A full base derivation for A.

a successful base derivation for the given argument. Parts 2 and 3 provide a sound and complete procedure for generating bases.

THEOREM 10
Let A be an argument in an argumentation framework $\mathcal{AF}$

1. For finitary argumentation framework A is credulously accepted iff there exists a successful base derivation for A.
2. For finitary argumentation framework there exists a full base derivation for A.
3. If $T_0, T_1, ..., T_n$ is a full base derivation for A, then the set $\{SP \mid \langle \emptyset, \emptyset, SP, SO \rangle \in T_n\}$ is a base of A.

EXAMPLE 8
The argumentation framework $\mathcal{AF}$=$(\mathcal{A}, att)$ depicted on the right of Figure 11 has two preferred extensions {A,E} and {A,F}. The sequence $\{\underline{t_0}\}, \{\underline{t_1}\}, \{\underline{t_2}, t_3\}, \{t_4, \underline{t_3}\}, \{t_4, \underline{t_5}\}, \{\underline{t_4}, t_7\}$ and $\{t_6, t_7\}$ depicted on the left is a full base derivation for A. Hence $\{\{A, E\}, \{A, F\}\}$ is a base of A. The notation $\underline{t}$ means t is selected.

3.1.2 Verifying the completion of a base

Intuitively a base $\mathcal{B}$ is not complete iff there is a preferred extension E which is not superset of any admissible set $S \in \mathcal{B}$. The following lemma suggests a way to verify this condition.

LEMMA 6
If E is a preferred extension and S is a non-empty admissible set of arguments but not a subset of E, then E attacks S (and S also attacks E).

Thus it is easy to see that:

LEMMA 7
A base $\mathcal{B}$ for an argument A is complete iff it is not empty and there is no admissible set of arguments attacking every elements of $\mathcal{B}$.

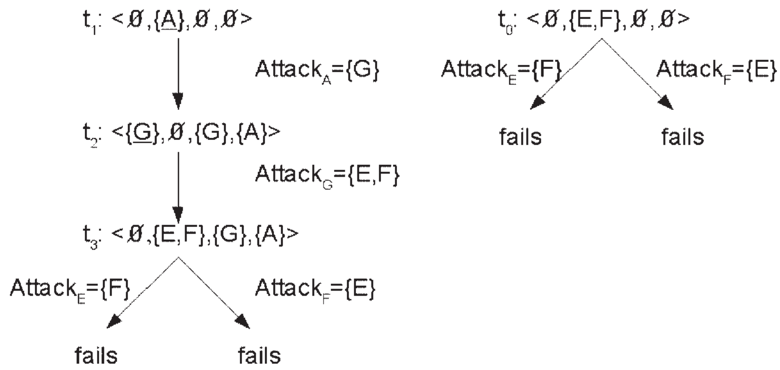

FIGURE 12. A full base derivation against {{A}, {E,F}}.

Lemma 8 re-states Lemma 7 by means of Definition 12.

DEFINITION 12
Given a base $\mathcal{B}$ for A, let $\mathcal{X}_{\mathcal{B}} = \{O \mid O$ is a minimal (under set inclusion) set of arguments such that $\forall S \in \mathcal{B},\ O \cap S \neq \emptyset\}$.

LEMMA 8
Let $\mathcal{B}$ be a base of A in an argumentation framework. $\mathcal{B}$ is a complete base of A iff $\mathcal{B}$ is not empty and for each $O \in \mathcal{X}_{\mathcal{B}}$ there is no admissible set attacking every elements of O.

To verify that there is no admissible set attacking every elements of a set O of arguments, we re-state Theorem 6 in terms of base derivation as follows.

DEFINITION 13
A full base derivation against a set $\mathcal{X}$ of argument sets is a partial base derivation $T_0, T_1, \ldots, T_n$ such that $T_0 = \{\langle \emptyset, O, \emptyset, \emptyset \rangle \mid O \in \mathcal{X}\}$, and $T_n = \emptyset$.

LEMMA 9
Let $\mathcal{AF}$ be a finitary argumentation framework, and O be a finite set of arguments of $\mathcal{AF}$. There is no admissible set attacking every elements in O iff there exists a full base derivation against $\{O\}$.

Combining Lemmas 8 and 9, the following lemma reduces the problem of verifying the completion of a base to an application of base derivation.

LEMMA 10
1. Suppose $\mathcal{B}$ is a base of A. If there is a full base derivation against $\mathcal{X}_{\mathcal{B}}$, then $\mathcal{B}$ is a complete base of A.
2. If $\mathcal{AF}$ is finitary and $\mathcal{B}$ is a finite complete base of A such that each element of $\mathcal{B}$ is finite, then there exists a full base derivation against $\mathcal{X}_{\mathcal{B}}$.

EXAMPLE 9 (Continue Example 8)
A base $\mathcal{B}$ of A is {{E,A},{F,A}}. Hence $\mathcal{X}_{\mathcal{B}}$ = {{A},{E,F}}. As depicted in Figure 12, the sequence: $\{t_0, \underline{t_1}\}, \{\underline{t_2}, t_0\}, \{\underline{t_3}, t_0\}, \{\underline{t_0}\}, \{\}$ is a full base derivation against $\mathcal{X}_{\mathcal{B}}$ (The notion of 'fails' means a partial dispute derivation *failed* to extend to a full dispute derivation). Hence $\mathcal{B}$ is a complete base of A.

Combining Theorem 10 parts 2, 3 and Lemma 10 leads to a sound and complete proof procedure for sceptical preferred acceptance in finitary argumentation frameworks.

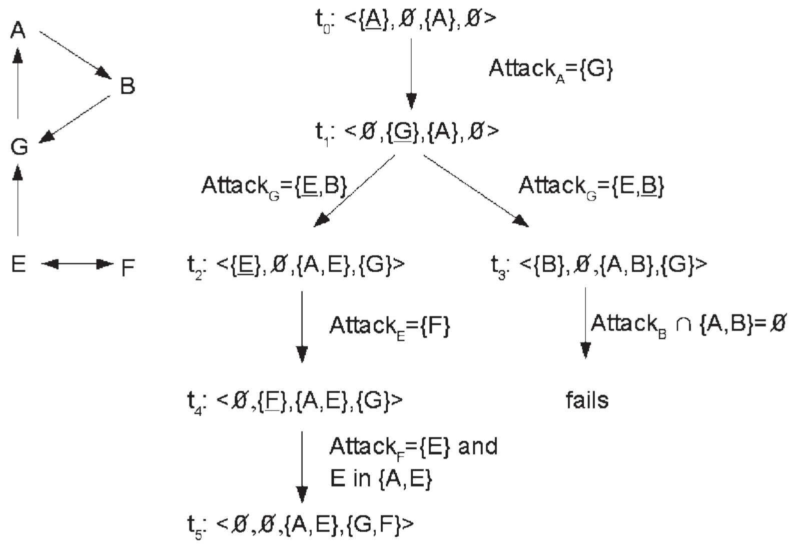

FIGURE 13. A full base derivation for A.

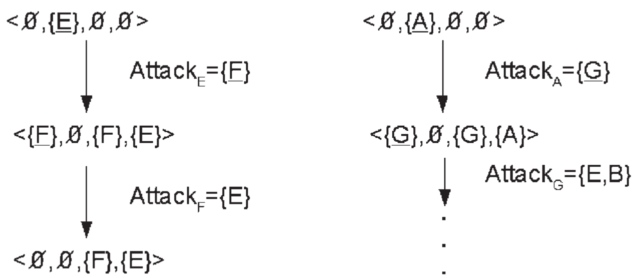

FIGURE 14. A full base derivation against {{A}, {E}}.

THEOREM 11

1. Let A be an argument in an argumentation framework $\mathcal{AF}$. If there exists a full base derivation for A that results in a non-empty base $\mathcal{B}$, and a full base derivation against $\mathcal{X}_\mathcal{B}$, then A is sceptically preferred accepted.
2. Let A be a sceptically preferred accepted argument in a finitary argumentation framework $\mathcal{AF}$. There exist a full base derivation for A resulting in a non-empty base $\mathcal{B}$, and a full base derivation against $\mathcal{X}_\mathcal{B}$.

EXAMPLE 10 (Continue Examples 8 and 9)
A is sceptically preferred accepted.

EXAMPLE 11
The argumentation framework depicted on the left of Figure 13 has two preferred extensions $\{F\}$, $\{A,E\}$. A construction of a full base derivation for A is presented on the right with $T_n = \{t_5\}$, resulting in a base $\mathcal{B}$={{A,E}} of A. Hence $\mathcal{X}_\mathcal{B} = \{\{A\},\{E\}\}$. A full base derivation against $\mathcal{X}_\mathcal{B}$ attempted in Figure 14 fails because of tuple $\langle \emptyset,\emptyset,\{F\},\{E\}\rangle$, which replaces tuple $\langle \emptyset,\{E\},\emptyset,\emptyset\rangle$ in the initial set after some steps. Thus A is not sceptically preferred accepted.

We are not aware of any dialectical proof procedures computing the sceptical preferred semantics in general cases as ours. However, there are a number of existing tools for argumentation frameworks satisfying specific constraints. For example, the sceptical TPI-dispute procedure [59] is defined in terms of the credulous TPI-dispute procedure, relying on the following proposition [14, 28, 59]: an argument A is sceptically accepted if A is credulously accepted and there exists no admissible set attacking A. This procedure is proven to be sound and complete for *coherent* frameworks [29, 59], i.e. frameworks where each preferred extension is also stable. In Example 11, although argument A is credulously accepted and there exists no admissible set attacking it, A is *not sceptically accepted*. However, the sceptical TPI-dispute procedure cannot give this answer.

The algorithm in [14] computes the sceptical preferred semantics 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. In the second step, it looks for an admissible set that cannot be extended into a bigger one containing A. Failure to find such a set implies that A is included in each preferred extension. In other words, the algorithm represents an indirect way of proving that A is sceptically preferred based on the idea that failure to show that A is not sceptical preferred implies the contrary. Though the idea is intuitive, no formal proof for the soundness of the algorithm is given.

3.2 Computing ideal semantics

An argument A is ideally accepted if it can be defended by an admissible set S that is subset of every preferred extensions. Hence there is no preferred extensions attacking S (Lemma 6). The reference [23] puts this formally.

LEMMA 11
An argument A is ideally accepted iff there is an admissible set S containing A such that for any argument in S there is no admissible set of arguments attacking it.

Thus the following theorem provides sound and complete proof procedure for ideal acceptance in finitary argumentation frameworks.

THEOREM 12
1. Let A be an argument in an argumentation framework $\mathcal{AF}$. If there exists a dispute derivation for A ended by $\langle \emptyset, \emptyset, SP_n, SO_n \rangle$, and a full base derivation against $\{\{B\} \mid B \in SP_n\}$, then A is ideally accepted.
2. Let A be an ideally accepted argument in a finitary argumentation framework $\mathcal{AF}$. For every selection function, there exist a dispute derivation for A ended by $\langle \emptyset, \emptyset, SP_n, SO_n \rangle$, and a full base derivation against $\{\{B\} \mid B \in SP_n\}$.

The reference [23] presents an algorithm computing the ideal extension both abstract and assumption-based argumentation frameworks based on Lemma 11. However, for abstract argumentation it is defined in terms of ideal dispute trees, i.e. admissible dispute trees that for no opponent node there exists an admissible dispute trees rooted at it. Hence, it involves two steps (not necessarily separated). In the first step, it searches for an admissible dispute tree and in the second step it verifies that this tree is ideal. Following Theorem 12, our proof procedure replaces the first step by a search for a credulous dispute derivation, and the second step by an application of base derivation.

4 Algorithmic issues

The vehicle for implementation is full/successful base derivations. For flexibility, we implement versions slightly modified from Definition 11 to allow the initial set to contain more than one tuple.

DEFINITION 14

- A *successful* base derivation for an initial set T_0 is a finite partial base derivation $T_0, T_1, ..., T_n$ such that T_n contains at least one tuple of the form $\langle \emptyset, \emptyset, SP, SO \rangle$.
- A *full* base derivation for an initial set T_0 is a finite partial base derivation $T_0, T_1, ..., T_n$ such that T_n contains only tuples of the form $\langle \emptyset, \emptyset, SP, SO \rangle$.

ALGORITHM 1 (*FullBaseDerivation*(T_0))
To construct a full base derivation for an initial set T_0, let
$T := T_0$
while not (all tuples of T have the form $\langle \emptyset, \emptyset, SP, SO \rangle$)
  select $t = \langle P, O, SP, SO \rangle \in T$, and $B \in P$ or $B \in O$
  remove t from T
  if $B \in P$, and $Attack_B \cap SP = \emptyset$, then
    add $\langle P \setminus \{B\}, O \cup (Attack_B \setminus SO), SP, SO \rangle$ to T
  if $B \in O$, then
    for each $C \in Attack_B \setminus (SO \cup O)$
      add $\langle P \cup \{C\}, O \setminus Attacked_C, SP \cup \{C\}, SO \cup (Attacked_C \cap O) \rangle$ to T
return T

ALGORITHM 2 (*SuccessfulBaseDerivation*(T_0))
To construct a successful base derivation for an initial set T_0, let
$T := T_0$
while not (a tuple of T has the form $\langle \emptyset, \emptyset, SP, SO \rangle$)[7]
  select $t = \langle P, O, SP, SO \rangle \in T$, and $B \in P$ or $B \in O$
  remove t from T
  if $B \in P$, and $Attack_B \cap SP = \emptyset$, then
    add $\langle P \setminus \{B\}, O \cup (Attack_B \setminus SO), SP, SO \rangle$ to T
  if $B \in O$, then
    for each $C \in Attack_B \setminus (SO \cup O)$
      add $\langle P \cup \{C\}, O \setminus Attacked_C, SP \cup \{C\}, SO \cup (Attacked_C \cap O) \rangle$ to T
return T

4.1 Computing credulous semantics

According to Theorem 10 part 1, the proof procedure for credulous acceptance needs to construct a successful base derivation for a given argument.

ALGORITHM 3 (*CredulousAcceptance*(A))
To answer the credulous acceptance of an argument A, let
$T := SuccessfulBaseDerivation(\{\langle \{A\}, \emptyset, \{A\}, \emptyset \rangle\})$
if $T \neq \emptyset$ return *true* else return *false*

[7] This algorithm and Algorithm 1 differ only in this line.

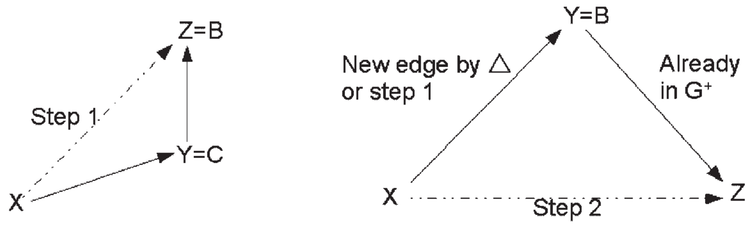

FIGURE 15. Two–step transitivity correction.

4.2 Computing grounded semantics

We define the notion of partial/successful ground base derivation similarly to the notion of *partial/successful credulous* base derivation in Definition 10 by applying ground dispute derivation (Definition 8) instead of credulous dispute derivation (Definition 4). Thus the following theorem is analogous to Theorem 9.

THEOREM 13
There is no infinite partial ground base derivation for finitary argumentation framework.

The proof of this theorem and of all other results in this section are given in Appendix C.

Theorem 7 reduces the grounded acceptance problem to the problem of searching for a ground dispute derivation. The following theorem reduces this search problem to the problem of constructing a successful ground base derivation.

THEOREM 14
Let $\mathcal{AF}$ be a finitary argumentation framework, and A be an argument of $\mathcal{AF}$. A is groundedly accepted iff there exists a successful ground base derivation for A.

To implement successful ground base derivation, we need to check if induced graphs are acyclic. As a graph is acyclic iff its transitive closure does not contain any self-loop, this check can rely on an algorithm computing transitive closures[8] stepwise as follows. Suppose at step i of a derivation, the dispute tree induces a graph G, and its transitive closure G^+ has been computed. The dispute tree at step $i+1$ induces a new graph $R = G \cup \triangle$ where $\triangle = \{(C,B) \mid C \in \mathcal{E}\}$ with $\mathcal{E}$ consists of newly presented arguments (i.e. $\mathcal{E} = Attack_B$ if B is selected from P_i and $\mathcal{E} = \{C\}$ if B is selected from O_i in the Definition 8). Given $G^+, B, \mathcal{E}$ as input, the algorithm computes R^+, by first setting $R^+ = G^+ \cup \triangle$, then correcting triples (X,Y,Z) that violate transitivity, i.e. $(X,Y),(Y,Z) \in R^+$ but $(X,Z) \notin R^+$. The correction proceeds in two steps, illustrated in Figure 15.

- Step 1: Connect X to Z due to new edge $(Y,Z) \in \triangle$.
- Step 2: Connect X to Z due to X and B have just been connected by Step 1 or by $\triangle$.

So the algorithm consists of two loops, respectively, performing two steps.

ALGORITHM 4 (*TransitiveClosure*$(G^+, B, \mathcal{E})$)
$R^+ := G^+ \cup \triangle_{\mathcal{E}}$;
for each $Y \in \mathcal{E}$
  for each node X appearing in R

[8] Our algorithm is motivated by the n^3 algorithm in [39, p. 36] computing transitive closures bottom-up.

if $(X,Y) \in R^+$ but $(X,B) \notin R^+$ then add (X,B) to R^+
for each node X of R
for each node Z of R
if $(X,B),(B,Z) \in R^+$ but $(X,Z) \notin R^+$ then add (X,Z) to R^+
return R^+

The correctness follows from the fact that the precise order in which triples (X,Y,Z) is examined in both loops is immaterial because the introduction of a new edge (X,Z) does not induce violations of triples already considered.

LEMMA 12
TransitiveClosure $(G^+, B, \mathcal{E})$ computes the transitive closure of $G \cup \{(B,C) \mid C \in \mathcal{E}\}$ in at most n^2 steps.

Using the above algorithm, the definition of successful ground base derivation can be rewritten as an algorithm as follows.

ALGORITHM 5 (*SuccessfulGroundBaseDerivation*(T_0))
To construct a successful ground base derivation from an initial set T_0, let
$T := T_0$
while not(a tuple of T has the form $\langle \emptyset, \emptyset, SP, SO, G^+ \rangle$)
select $t = \langle P, O, SP, SO, G^+ \rangle \in T$, and $B \in P$ or $B \in O$
remove t from T
if $B \in P$, and $Attack_B \cap SP = \emptyset$, then
$R^+ := TransitiveClosure(G^+, B, Attack_B)$
if R^+ is acyclic, then
add $\langle P \setminus \{B\}, O \cup (Attack_B \setminus SO), SP, SO, R^+ \rangle$ to T
if $B \in O$, then
for each $C \in Attack_B \setminus (SO \cup O)$
$R^+ := TransitiveClosure(G^+, B, \{C\})$
if R^+ is acyclic, then
add $\langle P \cup \{C\}, O \setminus \{B\}, SP \cup \{C\}, SO \cup \{B\}, R^+ \rangle$ to T
return T

According to Theorem 14, the proof procedure for grounded acceptance needs to construct a successful ground base derivation for the argument given.

ALGORITHM 6 (*GroundedAcceptance*(A))
To answer the grounded acceptance of an argument A, let
$T := SuccessfulGroundBaseDerivation(\{\langle \{A\}, \emptyset, \{A\}, \emptyset, \emptyset \rangle\})$
if $T \neq \emptyset$ return *true* else return *false*

4.3 Computing sceptical-preferred semantics

According to Theorem 11, the proof procedure answering the sceptical-preferred acceptance of argument A needs to construct a full base derivation for A, which results in some base $\mathcal{B}$, and a full base derivation against $\mathcal{X}_{\mathcal{B}}$. Failure to construct either full base derivations confirms that A is not sceptically preferred accepted.

ALGORITHM 7 (*ScepticalPreferredAcceptance*(A))
To answer the sceptical preferred acceptance of an argument A, let
$T := FullBaseDerivation(\{\langle \{A\}, \emptyset, \{A\}, \emptyset \rangle\})$

if $T = \emptyset$ then return false
else
    $\mathcal{B} := \{SP \mid \langle \emptyset, \emptyset, SP, SO \rangle \in T\}$
    $J_0 := \{\langle \emptyset, O, \emptyset, \emptyset \rangle \mid O \in \mathcal{X}_{\mathcal{B}}\}$
    $J := FullBaseDerivation(J_0)$
    if $J = \emptyset$ then return *true* else return *false*

4.4 Computing ideal semantics

According to Theorem 10 part 1 and Theorem 12, the proof procedure answering the ideal acceptance of argument A needs to construct a successful base derivation for A to find an admissible set SP defending A, then constructs a full base derivation against $\{\{B\} \mid B \in SP\}$. If it fails in the second step, it needs to resume the first step to find another admissible set defending A. If the first step collectively constructs a full base derivation for A but no full base derivation is successfully constructed in the second step, then A is not ideally accepted.

ALGORITHM 8 (*IdealAcceptance*(A))
To answer the ideal acceptance of an argument A, let
$T_0 := \{\langle \{A\}, \emptyset, \{A\}, \emptyset \rangle\}$
while $T_0 \neq \emptyset$
    $T := SuccessfulBaseDerivation(T_0)$
    if $T = \emptyset$ then return false
    else
        find a tuple $t = \langle \emptyset, \emptyset, SP, SO \rangle \in T$
        $J_0 := \{\langle \emptyset, \{B\}, \emptyset, \emptyset \rangle \mid B \in SP\}$
        $J := FullBaseDerivation(J_0)$
        if $J = \emptyset$ then return *true*
        else $T_0 := T \setminus \{t\}$

For finitary argumentation frameworks, all algorithms terminate due to there being no infinite partial base derivations as stated by Theorems 9, 13.

THEOREM 15
For finitary argumentation frameworks, algorithms 3, 6, 7, 8 terminate.

5 Conclusions

In the last decade, the theory of abstract argumentation has been extended in three directions. A number of authors develops systems that are instances of the abstract argumentation framework to deal with the question of how to build arguments and identify attacks. For example, in assumption-based argumentation framework [11], arguments are obtained by reasoning backwards using a set of rules in an underlying logic from conclusions to assumptions, and attacks are defined in terms of a notion of contrary of assumptions. In the second line of work, new semantics are proposed to overcome certain shortcomings of the semantics studied in [20]. Notably are the CF_2 semantics introduced by Baroni *et al.* [3, 6] which deals with some problematic behaviours of preferred semantics in occurrences of odd-length cycles; the semi-stable semantics of Caminada [12] which improves on the stable semantics in that semi-stable extensions always exist, and are stable if stable extensions exist; and the prudent semantics of Coste-Marquis *et al.* [15] which forbids indirect attacks within its extensions. Readers are referred to papers of Baroni and Giacomin, such as [4] proposing a set of criteria for evaluating proposals of new semantics, [5] classifying argumentation frameworks where different semantics coincide. In the third line of work, the framework is extended to deal with new features. Work done by Bench-Capon in [7, 8] dealt with social values that arguments promote, providing a natural basis for legal case-based reasoning [2, 9, 10]. Amgoud and Cayrol in [1] augmented the abstract argumentation framework with a preference relation between arguments, resulting in a preference-based argumentation framework in which an attack (A,B) only succeeds if B is not preferred to A. Recently Modgil in [41] extends [20] to provide a framework accommodating both values and preferences.

Therefore there is a need for a common framework to develop proof procedures for all three kinds of extensions. The common framework presented in this article captures dialectical proof procedures for four different semantics for abstract argumentation frameworks. The framework consists of only two notions of dispute derivations and base derivations. Dispute derivation is a dialectical notion for computing the credulous and grounded semantics. Base derivation, introduced in this article for the first time, characterizes a preferred extension containing some given argument by constructing a dispute derivation for it. However, a novel role of base derivation is to represent backtracking in the search for dispute derivations, and represent proofs for argument acceptance under the ideal/sceptical-preferred semantics. Thus, this notion becomes the centre of our framework, in that it not only shows the relationship between different proof procedures, but also offers itself a vehicle for implementation. We prove the soundness of obtained proof procedures for any argumentation frameworks and their completeness for a general class of finitary argumentation frameworks containing the class of finite argumentation frameworks as a subclass. We obtained algorithmic forms of all proof procedures, independent from selection functions. This is a nice feature because it leaves the room for players to engage in any active debate, opening an opportunity to make use of such heuristics as focusing on arguments with less attacks.

It remains a question of how the proposed framework could be used to develop proof procedures for extensions of the abstract argumentation framework. Some work has been done for the assumption-based argumentation framework. Concretely, dispute derivation has been used to compute admissible, ideal and grounded beliefs [21, 23] and has been implemented in CASAPI system [31]. Since CASAPI system (version v4.3 [32]) supports frameworks that are hybrid between abstract and assumption-based argumentation, it could be naturally augmented with base derivation for computing sceptical-preferred beliefs. The proposed framework could also be naturally applied to the extension of Modgil [41]. Though this extended framework does not bring any new power since it can be translated back to the abstract framework, this application would be appealling to those who find that defining problems in [41] is more convenient.

In contrast to the dialectical approach, Vreeswijk [58] presents an algorithm computing simultaneously the relevant part of the grounded extension and all relevant parts of the preferred extensions defending a given argument. The algorithm works by enforcing a labelling (*in, out, undecided*) on each argument encountered during computation and performs filtering in order to terminate early. The algorithm of Verheij in [56] for credulous acceptance, though presented in terms of labelling, can be clearly analysed by our framework because its two mutual recursive functions, *ExtendByAttack* and *ExtendByDefence*, respectively, can be seen as concrete implementations of steps 3a and 3b of our credulous dispute derivation. The labelling approach also includes the algorithm of Caminada [13] for finding all preferred extensions. Nieves *et al.* in [44] provide a mapping that constructs a disjunctive logic program P from an argumentation framework $\mathcal{AF}$ such that the preferred extensions of $\mathcal{AF}$ corresponds to the stable models of P. Then they show how to infer preferred extensions by using UNSAT algorithms and disjunctive stable model solvers.

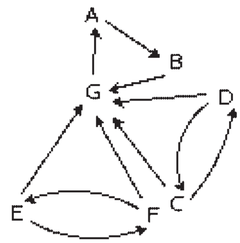

FIGURE 16. A framework with strongly connected components.

As noted by [58], argumentation systems where arguments are constructed dynamically can only rely on query-based algorithms like ours.

It is a well-known result from [27, 28] that credulous acceptance is NP, ideal acceptance is CO-NP-hard, and sceptical-preferred acceptance is $\prod_2^{(p)}$-complete. Therefore in worst cases, computing sceptical acceptance is not polynomial. Concrete complexity analysis of our presented algorithms is still open, however to certain extent, is reflected in how frequent base derivation is used to represent proofs for acceptance under corresponding semantics. Hence, optimization could focus around base derivation. For example, a further transformation could be applied in step 2a of Definition 10 to increase efficiency as follows: if T_{i+1} contains any tuple t of which the O component contains the selected argument B, then apply 2b recursively to T_{i+1} as if B was selected from t until there is no such tuple. However, this transformation requires an *occur* check which could decrease efficiency if such occurrences are only infrequent. To see another opportunity for optimization, consider the argumentation framework in Figure 16.

Using a full base derivation for A, we would be able to generate a base $\mathcal{B}$ = {{A,E,C}, {A,E,D},{A,F,C},{A,F,D}}. Looking at the subgraph consisting of only E, F, we could realize that if there is any attack against E or F, it should come from within this subgraph. Similarly for C, D. Hence, it would be enough if in the second part of the proof showing the sceptically preferred acceptance of A, we consider only a full base derivation against {A},{E,F},{C,D}. Structuring argumentation frameworks into strongly connected component like in [6] would facilitate optimizing the computation of sceptical preferred acceptance in this direction.

We plan to expand the presented framework to other semantics like semi-stable or CF_2.

Acknowledgements

We thank the referees for constructive comments and criticisms.

Funding

Sixth Framework IST program of the European Commission under the 035200 ARGUGRID project.

References

[1] L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, **34**, 197–215, 2002.

[2] K. Atkinson and T. J. M. Bench-Capon. Legal case-based reasoning as practical reasoning. *Artificial Intelligence and Law*, **13**, 93–131, 2005.

[3] P. Baroni and M. Giacomin. Solving semantic problems with odd-length cycles in argumentation. In *ECSQARU*, vol. 2711 of *Lecture Notes in Computer Science*, T. D. Nielsen and N. L. Zhang, eds, pp. 440–451. Springer, 2003.

[4] P. Baroni and M. Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, **171**, 675–700, 2007.

[5] P. Baroni and M. Giacomin. A systematic classification of argumentation frameworks where semantics agree. In *Computational Models of Argument: Proceedings of COMMA 2008, Toulouse, France, May 28–30, 2008*, vol. 172 of *Frontiers in Artificial Intelligence and Applications*, P. Besnard, S. Doutre and A. Hunter, eds, pp. 37–48. IOS Press, 2008.

[6] P. Baroni, M. Giacomin and G. Guida. Scc-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, **168**, 162–210, 2005.

[7] T. J. M. Bench-Capon. Agreeing to differ: modelling persuasive dialogue between parties with different values. *Informal Logic*, **22**, 2002, 2003.

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

[9] T. J. M. Bench-Capon, K. Atkinson and A. Chorley. Persuasion and value in legal argument. *Journal of Logic and Computation*, **15**, 1075–1097, 2005.

[10] T. J. M. Bench-Capon and G. Sartor. A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence*, **150**, 97–143, 2003.

[11] A. Bondarenko, P. M. Dung, R. A. Kowalski and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, **93**, 63–101, 1997.

[12] M. Caminada. Semi-stable semantics. In *COMMA*, vol. 144 of *Frontiers in Artificial Intelligence and Applications*, P. E. Dunne and T. J. M. Bench-Capon, eds, pp. 121–130. IOS Press, 2006.

[13] M. Caminada. An algorithm for computing semi-stable semantics. *Technical Report UU-CS-2007-010*. Department of Information and Computing Sciences, Utrecht University, 2007.

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

[15] S. Coste-Marquis, C. Devred and P. Marquis. Prudent semantics for argumentation frameworks. In *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2005)*, pp. 568–572. IEEE Computer Society, 2005.

[16] S. Doutre and J. Mengin. Preferred extensions of argumentation frameworks: query answering and computation. In *First International Joint Conference Automated Reasoning (IJCAR 2001)*, vol. of 2083 *Lecture Notes in Artificial Intelligence*, pp. 272–288, Springer, 2001.

[17] P. M. Dung. Negations as hypotheses: an abductive foundation for logic programming. In *Proceedings of the Eighth International Conference on Logic Programming*, pp. 3–17, 1991.

[18] P. M. Dung. Logic programming as dialogue games. *Technical report*. Division of Computer Science, Asian Institute of Technology, Bangkok, 1993.

[19] P. M. Dung. An argumentation theoretic foundation for logic programming. *Journal of Logic Programming*, **22**, 151–177, 1995.

[20] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, **77**, 321–358, 1995.

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

[22] P. M. Dung, R. A. Kowalski and F. Toni. Assumption-based argumentation. In *Argumentation in AI*, I. Rahwan and G. Simari, eds. Springer, 2009.

[23] P. M. Dung, P. Mancarella and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, **171**, 642–674, 2007.

[24] P. M. Dung and P. M. Thang. A sound and complete dialectical proof procedure for skeptical preferred argumentation. In *proceedings LPNMR-Workshop on Argumentation and Non-monotonic Reasoning (ArgNMR07)*, Arizona, 2007.

[25] P. M. Dung and P. M. Thang. Towards an argument-based model of legal doctrines in common law of contracts. In *9th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA)*, September 2008.

[26] P. M. Dung, P. M. Thang and F. Toni. Towards argumentation-based contract negotiation. In *International Conference on Computational Models of Argument*, pp. 134–146. 2008.

[27] P. E. Dunne. The computational complexity of ideal semantics I: Abstract argumentation frameworks. In *Computational Models of Argument: Proceedings of COMMA 2008, Toulouse, France, May 28–30, 2008*, vol. 172 of *Frontiers in Artificial Intelligence and Applications*, P. Besnard, S. Doutre and A. Hunter, eds, pp. 147–158. IOS Press, 2008.

[28] P. E. Dunne and T. J. M. Bench-Capon. Coherence in finite argument systems. *Artificial Intelligence*, **141**, 187–203, 2002.

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

[30] K. Eshghi and R. A. Kowalski. Abduction compared with negation by failure. In *Logic Programming: Proceedings of the Sixth International Conference*, pp. 234–254. The MIT Press, 1989.

[31] D. Gaertner and F. Toni. CaSAPI - a system for credulous and sceptical argumentation. In *First International Workshop on Argumentation and Nonmonotonic Reasoning, Arizona, USA*, 2007.

[32] D. Gaertner and F. Toni. Hybrid argumentation and its properties. In *Proceedings of the 2nd International Conference on Computational Models of Argument (COMMA08)*, pp. 183–195. IOS Press, 2008.

[33] A. J. García and G. R. Simari. Defeasible logic programming: an argumentative approach. *Theory and Practice of Logic Programming*, **4**, 95–138, 2004.

[34] G. Governatori, M. J. Maher, G. Antoniou and D. Billington. Argumentation semantics for defeasible logic. *Journal of Logic and Computation*, **14**, 675–702, 2004.

[35] H. Jakobovits and D. Vermeir. Dialectic semantics for argumentation frameworks. In *Proceedings of the 7th International Conference on Artificial Intelligence and Law*, pp. 53–62. 1999.

[36] A. Kakas, R. Kowalski and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, **6**, 719–770, 1993.

[37] A. Kakas and F. Toni. Computing argumentation in logic programming. *Journal of Logic and Computation*, **9**, 515–562, 1999.

[38] S. Kraus, K. P. Sycara and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, **104**, 1–69, 1998.

[39] H. R. Lewis and C. Papadimitriou. *Elements of the Theory of Computation*. Prentice-Hall, 1998.

[40] F. Lin and Y. Shoham. Argument systems: a uniform basis for nonmonotonic reasoning. In *Proceedings First International Conference on Principles of Knowledge Representation and Reasoning*, pp. 245–255. Morgan Kaufmann Publishers Inc., SanFrancisco, 1989.

[41] S. Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, **173**, 901–934, 2009.

[42] S. Modgil and M. Caminada. Proof theories and algorithms for abstract argumentation frameworks. In *Argumentation in AI*, I. Rahwan and G. Simari. eds. Springer, 2009.

[43] M. Morge. Computing argumentation for decision making in legal disputes. In *Computable Models of the Law: Languages, Dialogue, Games, Ontologies*, vol. 4884 of *Lecture Notes in Artificial Intelligence*, pp. 203–216. Springer, 2008.

[44] J. C. Nieves, U. Cortés and M. Osorio. Preferred extensions as stable models. *Theory and Practice of Logic Programming*, **8**, 527–543, 2008.

[45] S. Parson, C. Sierra and N. Jennings. Agent that reason and negotiate by arguing. *Journal of Logic and Computation*, **3**, 261–292, 1998.

[46] J. L. Pollock. Defeasible reasoning. *Cognitive Science*, **11**, 481–518, 1987.

[47] H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, **127**, 2001.

[48] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, **15**, 1009–1040, 2005.

[49] H. Prakken and G. Sartor. A system for defeasible argumentation, with defeasible priorities. In *Proceedings of the International Conference on Formal and Applied Practical Reasoning*, pp. 510–524. 1996.

[50] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, **7**, 25–75, 1997.

[51] I. Rahwan, L. Sonenberg and F. Dignum. Towards interest-based negotiation. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pp. 773–780. 2003.

[52] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons and L. Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, **4**, 343–375, 2003.

[53] N. Rescher. *Dialectics: A Controversy-Oriented Approach to the Theory of Knowledge*. State University of New York Press, 1978.

[54] G. R. Simari and R. P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, **53**, 125–157, 1992.

[55] F. Toni and A. C. Kakas. Computing the acceptability semantics. *Lecture Notes in Computer Science*, **928**, 401–415, 1995.

[56] B. Verheij. A labeling approach to the computation of credulous acceptance in argumentation. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6–12, 2007*, M. M. Veloso, ed., pp. 623–628. 2007.

[57] G. Vreeswijk. *Studies in defeasible argumentation*. PhD thesis, Vrije University, Amsterdam, Holland, 1993.

[58] G. Vreeswijk. An algorithm to compute minimally grounded and admissible defence sets in argument systems. In *International Conference on Computational Models of Argument*, pp. 109–120. 2006.

[59] G. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *Proceedings of the European Workshop on Logics in Artificial Intelligence*, pp. 239–253. 2000.

[60] M. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons, 2002.

Appendix

A Credulous and grounded semantics

A.1 Proof of Theorem 1

1. If part: the height of a finite dispute tree is $2k$. We prove that (in any frameworks) the defense set of a finite dispute tree is a subset of the grounded extension GE by induction on k, using two observations: (i) if S_1, S_2 are admissible subsets of GE, then so is $S_1 \cup S_2$; (ii) if argument A is acceptable w.r.t. S_1, then $S_1 \cup \{A\}$ is also an admissible subset of GE.
   $k=0$ corresponds to dispute trees containing a single node labelled by an argument that is attacked by no arguments. So its defense set containing only that argument is a subset of GE.
   Assume the assertion holds for all finite dispute trees of height smaller than $2k$. Let $\mathcal{T}$ be a dispute tree of height $2k$ with root labelled by argument A. For each argument B attacking a child of A, let $\mathcal{T}_B$ be the subtree of $\mathcal{T}$ rooted at B. Obviously $\mathcal{T}_B$ is a finite dispute tree with height smaller than $2k$. The union of defense sets of all $\mathcal{T}_B$ is a subset of GE, and admissibly defends A. So the defense set of $\mathcal{T}$ is an admissible subset of GE.
2. Only if part: to construct a finite dispute tree for a groundedly accepted argument in a finite-branching argumentation framework, we need the following lemma.

   LEMMA 13
   In a finite-branching argumentation framework, the grounded extension equals $\emptyset \cup \mathcal{F}(\emptyset) \cup \mathcal{F}^2(\emptyset) \cup \ldots$. This lemma follows immediately from two features, proven in [20], of the characteristic function $\mathcal{F}$:

   (a) (Lemma 19 [20]): $\mathcal{F}$ is monotonic w.r.t. set inclusion.
   (b) (Lemma 28 [20]): If $\mathcal{AF}$ is finite-branching, then $\mathcal{F}$ is ω-continuous.

   Thus each argument $A \in GE$ could be ranked by a natural number $r(A)$ such that $A \in \mathcal{F}^{r(A)}(\emptyset) \setminus \mathcal{F}^{r(A)-1}(\emptyset)$. So, $\{A \in GE \mid r(A)=1\}$ represents the set of arguments that are attacked by no other arguments, $\{A \in GE \mid r(A)=2\}$ represents the set of arguments defended by $\{A \in GE \mid r(A) \leq 1\}$, and so on. For each argument B attacking GE, let $r(B)=min\{r(A) \mid A \in GE$ and A attacks $B\}$. Clearly, B does not attack any argument in GE of rank smaller than $r(B)$.
   Now given any argument $A \in GE$, we can build a dispute tree $\mathcal{T}$ for A as follows. The root of $\mathcal{T}$ is labelled by A. For each $B \in Attack_A$, we select argument C to counterattacks B such that $r(C)=r(B)$, then for each arguments $D \in Attack_C$, we selects argument E to counterattack D such that $r(D)=r(E)$, and so on. So for each branch of $\mathcal{T}$, the rank of a proponent node is equal to that of its opponent parent node, but the rank of an opponent node is smaller than that of its parent proponent node. In otherwords, ranking decreases downwards. So all branches of $\mathcal{T}$ are of finite length. Since $\mathcal{AF}$ is finite-branching, $\mathcal{T}$ is finite in breath. Thus $\mathcal{T}$ is finite.

A.2 Proof of Theorem 2

1. If part: let $\mathcal{SD}=\langle P_0,O_0\rangle \ldots \langle P_n,O_n\rangle$ with $P_0=\{A\}$ and $O_0=O_n=P_n=\emptyset$ be a full simple dispute derivation for A. A finite dispute tree $\mathcal{T}$ for A can be built by a sequence of partial dispute trees $\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_n$ with $\mathcal{T}=\mathcal{T}_n$ as follows.

   (a) $\mathcal{T}_0$ is a partial dispute tree consisting of only one node labelled by A

(b) let B be the argument selected at step i in $\mathcal{SD}$
   i. if $B \in P_i$ then $\mathcal{T}_i$ is expanded into $\mathcal{T}_{i+1}$ by adding, for each $C \in Attack_B$, an opponent child labelled by C, to any frontier proponent node labelled by B
   ii. if $B \in O_i$ and C is selected from $Attack_B$, then $\mathcal{T}_i$ is expanded into $\mathcal{T}_{i+1}$ by adding a proponent child labelled by C, to any frontier opponent node labelled by B

   Clearly $\mathcal{T}_n$ is a finite dispute tree for A. From Theorem 1 A is groundedly accepted.
2. Only if part: let A be a groundedly accepted argument in a finite argumentation framework $\mathcal{AF}$. So there exists a finite dispute tree $\mathcal{T}$ for A (Theorem 1). It is easy to see that for any selection function sl, a full simple dispute derivation for A can be built by a traversal on $\mathcal{T}$ using sl to select the next node to visit.

A.3 Proof of Theorem 3

As elaborated in the paragraph following this theorem, we just need to prove that in 3b of Definition 4, $SO_i \subsetneq SO_{i+1}$. This follows from two observations:

1. $Attacked_C \cap O_i \neq \emptyset$. Obviously since $B \in Attacked_C \cap O_i$
2. $(Attacked_C \cap O_i) \cap SO_i = \emptyset$. We indeed prove that for any step j of a derivation, $O_j \cap SO_j = \emptyset$ as follows

   (a) if $j = 0$, it is clear that $O_0 \cap SO_0 = \emptyset \cap \emptyset = \emptyset$
   (b) Suppose the assertion holds for all $j \leq k$. If step $k+1$ of the derivation is computed from step k by 3a, then $O_{k+1} \cap SO_{k+1} = (O_k \cup (Attack_B \setminus SO_k)) \cap SO_k = (O_k \cap SO_k) \cup ((Attack_B \setminus SO_k) \cap SO_k) = \emptyset \cup \emptyset = \emptyset$. Otherwise, $O_{k+1} \cap SO_{k+1} = (O_k \setminus Attacked_C) \cap (SO_k \cup (Attacked_C \cap O_k)) \subseteq (O_k \setminus Attacked_C) \cap (SO_k \cup Attacked_C) = ((O_k \setminus Attacked_C) \cap SO_k) \cup ((O_k \setminus Attacked_C) \cap Attacked_C) = \emptyset \cup \emptyset = \emptyset$.

A.4 Proof of Lemma 2

1. Let $R = S \cap ENV_A$. S is conflict-free thus so is R. Let B be an argument attacking R. There is $C \in S$ such that C attacks B because B also attacks S. Obviously $B \in ENV_A$, thus $C \in ENV_A$. So $C \in R$. Hence R is admissible in both $\mathcal{AF}$ and $\mathcal{AF}_A$.
2. It is clear that S is conflict-free in $\mathcal{AF}$. Let B be an argument attacks S in $\mathcal{AF}$. Obviously $B \in ENV_A$. Hence S attacks B in $\mathcal{AF}_A$. So S attacks B in $\mathcal{AF}$.

A.5 Proof of Theorem 4

Let $\mathcal{D} = \langle P_0, O_0, SP_0, SO_0 \rangle \ldots \langle P_n, O_n, SP_n, SO_n \rangle \ldots$, where $X = P_0 \cup O_0$ is finite set, be a partial credulous dispute derivation in a finitary argumentation framework $\mathcal{AF}$,

For any $B \in \mathcal{A} \setminus (\bigcup_{A \in X} \{ENV_A\})$, there is no direct path from B to A in the graph of $\mathcal{AF}$. Thus $\mathcal{D}$ is also a partial credulous dispute derivation in $\mathcal{AF}_X = (\mathcal{A}_X, att_X)$, where $\mathcal{A}_X = \bigcup_{A \in X} \{ENV_A\}$, $att_X = \bigcup_{A \in X} \{att_A\}$ (att_A is the restriction of att to ENV_A). Because $\mathcal{AF}_X$ is finite, $\mathcal{D}$ is finite (Theorem 3).

A.6 Proof of Theorem 5

1. By *Step 2* (of *Definition 4*) $A \in SP_0$. By *Step 3b* $SP_0 \subseteq SP_1 \subseteq \ldots \subseteq SP_n$. Thus $A \in SP_n$. To prove SP_n is admissible, we shall prove (in a) that it is conflict-free and (in b) that for all $B \in SP_n$, B is acceptable w.r.t. SP_n.

   (a) $SP_0 = \{A\}$ is conflict-free. Suppose SP_i is conflict-free and C is an argument added to SP at step i (of the derivation), that is $SP_{i+1} = SP_i \cup \{C\}$ by *Step 3b*.
      i. C does not attack SP_i. This is because if C attacks SP_i, it must attack some $D \in P_i$ (Note that $C \notin SO_i \cup O_i$, the set of all arguments attacking $SP_i \setminus P_i$). Since $P_n = \emptyset$, D is selected at some step $l > i$ of the derivation. Then $C \in Attack_D \cap SP_l$, contradicting with the condition that $Attack_D \cap SP_l = \emptyset$ (according to *Step 3a*).
      ii. SP_i does not attack C. Assume the contrary, then there exists $D \in SP_i$ such that D attacks C. Because $C \in P_{i+1}$, it will be selected in some step $j \geq i+1$. Then $D \in Attack_C \cap SP_j$, contradicting with the condition that $Attack_C \cap SP_j = \emptyset$.

      Thus SP_n is conflict-free.

   (b) Let $B \in SP_n$. B is admitted to SP_n in some step $i < n$ by *Step 3b*. So $B \in P_{i+1}$. B is selected in some step $j > i+1$. Let D be an argument attacking B
      i. if $D \in SO_j$, then D is attacked by SP_j (because by *Step 3b* SP_j attacks every element of SO_j). So D is attacked by SP_n
      ii. if $D \notin SO_j$, then by *Step 3a*. $D \in O_{j+1}$. Since $O_n = \emptyset$, D is selected in some step $k > j+1$. By *Step 3b* an argument $C \in Attack_D$ is selected to be admitted to SP_{k+1}. So SP_{k+1} attacks D. SP_n attacks D accordingly.

      Thus B is acceptable w.r.t. SP_n

2. By Lemma 1 there is an admissible dispute tree $\mathcal{T}$ for A with defense set S' such that $S' \subseteq S$. Given a selection function sl, we can build a credulous dispute derivation for A: $\langle\{A\}, \emptyset, \{A\}, \emptyset\rangle \ldots \langle\emptyset, \emptyset, SP_n, SO_n\rangle$ such that $SP_n \subseteq S'$ by a traversal on $\mathcal{T}$ using sl to select the next node to visit. Furthermore for each i, constructed tuple t_i satisfies the following properties:

   (a) $SP_i \subseteq \mathcal{P}$, where $\mathcal{P}$ is the defense set of $\mathcal{T}$
   (b) $SO_i \subseteq \mathcal{O}$, where $\mathcal{O}$ is the set of arguments labeling opponent nodes of $\mathcal{T}$
   (c) for each $X \in P_i$, there is a proponent node of $\mathcal{T}$ labelled by X
   (d) for each $X \in O_i$, there is an opponent node of $\mathcal{T}$ labelled by X

   For the first tuple $t_0 = \langle\{A\}, \emptyset, \{A\}, \emptyset\rangle$, properties (a)–(d) clearly hold. Suppose we have constructed the derivation until step i, and above properties holds for t_i. If $P_i \cup O_i \neq \emptyset$, we can construct $t_{i+1} = \langle P_{i+1}, O_{i+1}, SP_{i+1}, SO_{i+1}\rangle$ as follows
   – If $B \in P_i$ is selected. From property (c) $Attack_B \subseteq \mathcal{O}$. Thus $Attack_B \cap SP_i = \emptyset$ because otherwise $\emptyset \subsetneq (Attack_B \cap SP_i) \subseteq \mathcal{O} \cap \mathcal{P}$, making $\mathcal{T}$ not admissible. So t_{i+1} can be constructed from t_i by Step 3a of Definition 4. It is easy to verify that properties (a)–(d) also hold for t_{i+1}
   – If $B \in O_i$ is selected. From property (d) there is an argument C labelling the proponent child node of the opponent node labelled by B. It is clearly that t_{i+1} is constructable from t_i by Step 3b of Definition 4 and properties (a)–(d) hold for t_{i+1}.

   Because there is no infinite credulous dispute derivation (shown by Theorem 4) the construction must finish after a finite number of steps.

A.7 Proof of Theorem 6

Given a finitary argumentation framework $\mathcal{AF} = (\mathcal{A}, att)$ and a finite set of arguments $O \subseteq \mathcal{A}$, let $\mathcal{AF}_O$ be the finitary framework formed by adding a new argument T_O to $\mathcal{A}$, and a set of attacks $\{(B, T_O) \mid B \in O\}$ to att. In $\mathcal{AF}_O$, for each credulous dispute derivation against O: $\mathcal{D} = \langle P_0, O_0, SP_0, SO_0\rangle \ldots \langle P_n, O_n, SP_n, SO_n\rangle$ with $O_0 = O$, there is a corresponding credulous dispute derivation for T_O:

$\mathcal{D}' = \langle \{T_O\}, \emptyset, \{T_O\}, \emptyset\rangle \langle P_0 \cup \{T_O\}, O_0, SP_0 \cup \{T_O\}, SO_0\rangle \ldots \langle P_n \cup \{T_O\}, O_n, SP_n \cup \{T_O\}, SO_n\rangle$. Note that $\mathcal{D}$ is also a credulous dispute derivation against O in $\mathcal{AF}$.

1. Only if part: let $\mathcal{D} = \langle P_0, O_0, SP_0, SO_0\rangle \ldots \langle P_n, O_n, SP_n, SO_n\rangle$ with $O_0 = O$ be a credulous dispute derivation against O.
   By Theorem 5 the credulous dispute derivation $\mathcal{D}'$ formed from $\mathcal{D}$ as above results in an admissible set $SP_n \cup \{T_O\}$ defending T_O. Since T_O is attacked by every elements of O and T_O attacks no arguments, SP_n attacks every elements of O.
2. If part: let S be admissible and attack every elements of O
   $S \cup T_O$ is admissible set in $\mathcal{AF}_O$ because T_O is acceptable w.r.t. S. Also by Theorem 5 there is a credulous dispute derivation $\mathcal{D}'$ for T_O in $\mathcal{AF}_O$. Then D formed from $\mathcal{D}'$ is a credulous dispute derivation against O in $\mathcal{AF}$.

A.8 Proof of Lemma 3

Assume there is $\mathcal{T}$, a non-redundant and infinite dispute tree in a finitary argumentation framework $\mathcal{AF}$. So either $\mathcal{T}$ has some node with an infinite number of children or has some branch of infinite length. Due to $\mathcal{AF}$ is finitary, the first scenario is not possible. Let $A_1, A_2, \ldots, A_n, \ldots$ be an infinite branch of $\mathcal{T}$. It is obvious that $A_i \in ENV_{A_1}$, which is a finite set. So an argument must be repeated in this sequence, contradicting with the assumption that $\mathcal{T}$ is non-redundant. So if a dispute tree is infinite, it must be redundant.

A.9 Proof of Lemma 4

1. If part: let $\mathcal{T}$ be a non-redundant dispute tree for an argument A. From Lemma 3, $\mathcal{T}$ is finite. From Theorem 1, A is groundedly accepted.
2. Only if part: let A be a groundedly accepted argument in a finite-branching argumentation framework. Consider the finite dispute tree $\mathcal{T}$ for A constructed in the proof of Theorem 1. The graph induced by $\mathcal{T}$ is acyclic, since otherwise there would be a finite sequence of arguments $A_1, A_2, \ldots, A_n$ with $A_1 = A_n$ labelling a path in this graph, resulting in a contradiction that two nodes labelled by A_1 on a corresponding branch of $\mathcal{T}$ are ranked equally. Thus $\mathcal{T}$ is also non-redundant.

A.10 Proof of Theorem 7

1. Let $\mathcal{GD} = \langle P_0, O_0, SP_0, SO_0, G_0\rangle, \ldots, \langle P_n, O_n, SP_n, SO_n, G_n\rangle$ be a full ground dispute derivation for an argument A. Following the method in the proof of Theorem 5, we can prove that SP_n is admissible and $A \in SP_n$.
   For each node $X \in G_n$, we define a dispute tree T_X as follows.

(a) the root of T_X is a node labelled by $p\!:\!X$ (resp., $o\!:\!X$) if X is presented by the proponent (resp., opponent).

(b) for each Y such that $(Y,X) \in G_n$, T_Y is a subtree of T_X. If X is presented by the proponent (resp., opponent), then T_Y is rooted as $o:Y$(resp., $p:Y$) which is a child node of $p\!:\!X$ (resp., $o\!:\!X$).

For all X presented by the proponent, it is easy to verify that T_X is finite. The defense set of T_A is SP_n. So SP_n is an subset of the grounded extension GE (see the proof of Theorem 1)

2. Let A be a groundedly accepted argument. By Lemma 4 there is a non-redundant dispute tree $\mathcal{T}$ for A . Like in the proof for Theorem 5, given a selection function, a ground dispute derivation for A can be constructed by a traversal on $\mathcal{T}$ using the selection function to select nodes to visit. For any step i during construction, G_i is acyclic because it is just a part of the acyclic graph induced by $\mathcal{T}$.

A.11 Proof of Theorem 8

Similar to proofs of Theorems 3, 4 (with $Attacked_C$ replaced by $\{B\}$)

B Sceptical semantics

B.1 Proof of Lemma 5

1. If part
   Let $\mathcal{B}$ be a complete base of A. For any preferred extension E there exists a set $S \in \mathcal{B}$ such that $S \subseteq E$. Since $A \in S$, $A \in E$. In otherwords, A is sceptically accepted.
2. Only if part
   The set of all preferred extensions is a complete base of A.

B.2 Proof of Theorem 9

Let $\mathcal{AF}$ be the finitary argumentation framework we are considering. A tuple t is said to be a descendant of a tuple s if there exists a partial dispute derivation $t_1,..,t_n$ such that $t_1 = s$, $t_n = t$. If $n = 2$, then t is said to be a child of s. Let $D(t)$ denote the set of all descendants of t. Because there is no infinite partial dispute derivations in $\mathcal{AF}$, and for any tuple there is a finite number of children, $D(t)$ is finite for any t.

Let $\mathcal{BD} = T_0, T_1, \ldots, T_n, \ldots$ be a partial base derivation. It is easy to see that each step i of $\mathcal{BD}$ represents a selection of a tuple $t \in T_i$ and its replacement by its children. So if a tuple t is selected sometimes in $\mathcal{BD}$, then $t \in \bigcup_{s \in T_0} \{D(s)\}$, a finite set. Moreover, there is a finite number of occurrences of t in $\mathcal{BD}$ since $t \notin D(t)$. In otherwords, $\mathcal{BD}$ is finite.

B.3 Proof of Theorem 10

1. (a) If part: let A be a credulously accepted argument in a finitary argumentation framework $\mathcal{AF}$. By Theorem 5 there is a credulous dispute derivation for A: $t_0, t_1, \ldots, t_n$ with $t_0 = \langle \{A\}, \emptyset, \{A\}, \emptyset \rangle$, $t_n = \langle \emptyset, \emptyset, \_, \_ \rangle$. Consider a partial base derivation for A containing just one set $T_0 = \{t_0\}$. By Theorem 9 there is a finite base derivation $T_0, \ldots, T_m$ that cannot be expanded into a longer derivation. Since no tuples in T_m could be selected as required by Definition 10,

either $T_m = \emptyset$ or T_m contains only (more than one) tuples of the form $\langle\emptyset,\emptyset,SP,SO\rangle$. It must be the later case since $t_n \in T_m$. In otherwords, $T_0,...,T_m$ is a successful base derivation for A.

   (b) Only if part: let $T_0,T_1,...,T_m$ be a successful base derivation for A. There is $\langle\emptyset,\emptyset,SP,SO\rangle \in T_m$. Thus there is a credulous dispute derivation for A. By Theorem 5, A is credulously accepted.

2. A finite partial base derivation specified as in part 1 (If part) $T_0,...,T_m$ is a full base derivation for A since either $T_m = \emptyset$ or T_m contains only tuples of the form $\langle\emptyset,\emptyset,SP,SO\rangle$.
3. Let E be any preferred extension containing A. E is an admissible set itself, so by Theorem 5 there is a credulous dispute derivation for A:
   $\langle\{A\},\emptyset,\{A\},\emptyset\rangle...\langle\emptyset,\emptyset,SP_n,SO_n\rangle$ such that $SP_n \subseteq E$. By definition 10 and 11, T_n contains $\langle\emptyset,\emptyset,SP_n,SO_n\rangle$. Thus $SP_n \in \mathcal{B}$. So $\mathcal{B}$ is a base of A.

B.4 Proof of Lemma 6

It is clear that if E attacks S then S also attacks E and vice versa. Assume that S,E do not attack each other. Then $C = S \cup E$ is conflict-free. For each argument $A \in C$ if there is an argument B attacking A then B is attacked by S or by E since A is in either S or E. So B is attacked by C. Hence each argument in C is acceptable w.r.t. C. Then C is admissible and contains E and there exists an argument G in C which is not in E because S is not subset of E. Contradiction since E is a preferred extension.

B.5 Proof of Lemma 7

1. If part: let $\mathcal{B}$ be a base of A such that there is no admissible set attacking every elements of $\mathcal{B}$. Let E be a preferred extension. Hence there is an admissible set $S \in \mathcal{B}$ that is not attacked by E. From Lemma 6, $S \subseteq E$. So $\mathcal{B}$ must be complete.
2. Only if part: let $\mathcal{B}$ be a complete base of A. If there is an admissible set attacking every elements of $\mathcal{B}$, then it can be extended to a preferred extension E attacking every elements of B. So for any $S \in \mathcal{B}$, E attacks S. From Lemma 6 $S \not\subseteq E$, contradicting with the completion of $\mathcal{B}$.

B.6 Proof of Lemma 8

This lemma follows immediately from Lemma 7 and the fact that an admissible set attacks every elements of some $O \in \mathcal{X}_{\mathcal{B}}$ iff it attacks every elements of $\mathcal{B}$.

B.7 Proof of Lemma 9

Let $\mathcal{A}F$ be a finitary argumentation framework, and O be a finite set of arguments in $\mathcal{AF}$.

1. If part: suppose there is no admissible set attacking every elements of O. Consider a partial base derivation containing just one set $T_0 = \{\langle\emptyset,O,\emptyset,\emptyset\rangle\}$. By Theorem 9 there is finite base derivation $T_0,T_1,...,T_n$ that cannot be expanded into a longer derivation. So either $T_n = \emptyset$ or T_n contains only (more than one) tuples of the form $\langle\emptyset,\emptyset,SP,SO\rangle$. The later case cannot happen since there is no credulous dispute derivation against O (Theorem 6). Thus $T_n = \emptyset$, in otherwords $T_0,T_1,...,T_n$ is a full base derivation against $\{O\}$
2. Only if part: suppose $T_0,T_1,...,T_n$ with $T_0 = \{\langle\emptyset,O,\emptyset,\emptyset\rangle\}$, $T_n = \emptyset$ is a full base derivation against $\{O\}$. So there is no dispute derivation against O. By Theorem 6, there is no admissible set attacking every elements of O.

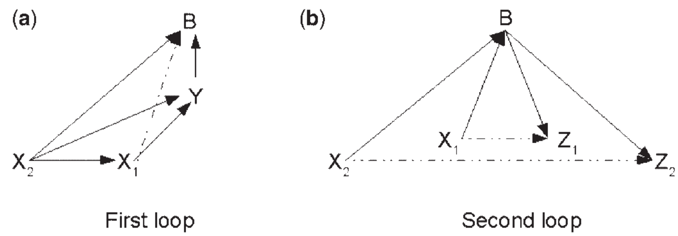

FIGURE C1.

B.8 Proof of Lemma 10

1. Let $\mathcal{B}$ be base of A and $\mathcal{BD} = T_0, ..., T_n$ be a full base derivation against $\mathcal{X}_\mathcal{B}$. For each $O \in \mathcal{X}_\mathcal{B}$ there is no admissible set attacking every elements of O because otherwise there is no full base derivation against $\{O\}$ (Lemma 9), contradicting with the existence of $\mathcal{BD}$. By Lemma 8, $\mathcal{B}$ is complete.
2. Let $\mathcal{B}$ be a finite complete base of A in a finitary $\mathcal{AF}$. So $\mathcal{X}_\mathcal{B}$ is finite and all its elements are also finite. Let O be an element of $\mathcal{X}_\mathcal{B}$. From the completion of $\mathcal{B}$, there is no admissible set attacking every elements of O (Lemma 8). From Lemma 9 there is a full base derivation against $\{O\}$. It follows that there is a full base derivation against $\mathcal{X}_\mathcal{B}$.

B.9 Proof of Theorem 11

1. From Lemma 10, $\mathcal{B}$ is complete base of A. Hence A is sceptically accepted (*Lemma 5*).
2. Since A is sceptically accepted, there is a full base derivation for A (Theorem 10) that results in a complete base $\mathcal{B}$. Due to Lemma 10, there is a full base derivation against $\mathcal{X}_\mathcal{B}$.

B.10 Proof of Theorem 12

1. It can be inferred from the credulous dispute derivation that SP_n is admissible and $A \in SP_n$ (Theorem 5). From the full base derivation against $\{\{B\} \mid B \in SP_n\}$ it follows that for any $B \in SP_n$ there is a full base derivation against $\{\{B\}\}$. From Lemma 9 there is no admissible set attacking B. Thus it follows from Lemma 11 that A is ideally accepted.
2. Because A is ideally accepted, by Lemma 11 there is an admissible set S containing A (property 1) such that there is no admissible set attacking any argument of S (property 2). From (1) and Theorem 5, for any selection function there is a credulous dispute derivation for A ended by $\langle \emptyset, \emptyset, SP_n, SO_n \rangle$ such that $SP_n \subseteq S$. From (2) and Lemma 9, there is a full base derivation against $\{\{B\} \mid B \in SP_n\}$.

C Algorithmic issue

C.1 Proof of Theorem 13

This proof is the same as that of Theorem 9.

C.2 *Proof of Theorem 14*

Similarly to Theorem 10 by applying Theorem 7 instead of Theorem 5.

C.3 *Proof of Lemma 12*

It is obvious that the complexity of the algorithm is n^2. To prove its correctness, we prove that after each loop, no violations could be found if the loop is run again. Figure C1 illustrates our proof.

- First loop:
  Let (X_1, Y, B) be a violation, which is found at some step of this loop. Suppose after this loop there is a violated triple (X_2, X_1, B) due to the introduction of (X_1, B). So $X_1 \neq B$ because otherwise (X_2, X_1, B) is not a violation. This leads to $(X_2, X_1) \in G^+$ because this loop only adds edges coming to B. From triple (X_2, X_1, Y) it must be that $(X_2, Y) \in G^+$. So triple (X_2, Y, B) must be found as a violation at some step of the loop, and (X_2, B) is introduced accordingly. So (X_2, X_1, B) cannot be a violation after the loop.
- Second loop:
  Suppose X_1 is connected to Z_1 due to triple (X_1, B, Z_1) which is found to be a violation at some step of this loop. Consider tuple (X_2, B, Z_2). If X_2 needs to be connected to Z_2 due to the introduction of (X_1, Z_1), then there is a path from X_2 to Z_2 going through B exactly one time. In otherwords, there is a path from X_2 to B, and a path from B to Z_2, both not going through B any times. So edges $(X_2, B), (B, Z_2)$ must be available after the first loop. Thus the violation of triple (X_2, B, Z_2) could also be detected in this second loop.

C.4 *Proof of Theorem 15*

Obviously.