

A Unified Framework for Representation and Development of Dialectical Proof Procedures in Argumentation

P. M. Dung P. M. Thang
Department of Computer Science
Asian Institute of Technology
dung@cs.ait.ac.th thangfm@ait.ac.th

Abstract

We present an unified methodology for representation and development of dialectical proof procedures in abstract argumentation based on the notions of legal environments and dispute derivations. A legal environment specifies the legal moves of the dispute parties while a dispute derivation describes the procedure structure. A key insight of this paper is that the opponent moves determine the soundness of a dispute while the completeness of a dispute procedure depends on the proponent moves.

1 Introduction

Argumentation is a form of reasoning, that could be viewed as a dispute resolution, in which the participants present their arguments to establish, defend or attack certain propositions.

The probably most abstract among the well-known formalisms for argumentation is the abstract argumentation framework [Dung, 1995b] (or abstract argumentation for short) consisting simply of a set of "atomic" arguments together with a binary relation representing attack relation between them. There is extensive research on dialectical proof procedures for abstract argumentation [Cayrol,Doutre,Mengin, 2003; Dunne,Bench-Capon, 2003; Modgil,Caminada, 2009; Vreeswijk,Prakken, 2000; Jakobovits,Vermeir, 1999; Vreeswijk, 2006; Verheij, 2007]. Though clearly related, their formal and precise relationship remains unexplored. We address this problem by providing an unified framework for representing dialectical proof procedures based on the notions of legal environments and dispute derivation trees. A key insight of this paper is that the opponent moves determine the soundness of a dispute while the completeness depends on the proponent moves.

2 Abstract Argumentation: Preliminaries

An abstract argumentation framework [Dung, 1995b] is 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 ($att \subseteq \mathcal{A} \times \mathcal{A}$) with $(A, B) \in att$ meaning A attacks B . For simplicity, we restrict ourself on frameworks with finite sets of arguments. 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' .

A set S of arguments is *conflict-free* iff it does not attack itself. Argument A is *acceptable* with respect to S iff S attacks each argument attacking A . S is *admissible* iff S is conflict-free and each argument in S is acceptable with respect to S . S is a *preferred extension* iff S is maximally (wrt set inclusion) admissible.

The semantics of argumentation could also be characterized by a fixpoint theory of the characteristic function $\mathcal{F}(S) = \{A \in \mathcal{A} \mid A \text{ is acceptable wrt } S\}$. It is easy to see that S is admissible iff S is conflict free and $S \subseteq \mathcal{F}(S)$. As \mathcal{F} is monotonic, it follows that S is a preferred extension iff S is a maximal fixpoint of \mathcal{F} . The least fixed point of \mathcal{F} is defined as the *grounded extension*.

An argument A is said to be *credulously accepted* iff it is contained in at least one preferred extension and *groundedly accepted* iff it is contained in the grounded extension.

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. $Attacked_B = \{A \mid (B, A) \in att\}$.

3 Dispute derivations

In a dispute, the proponent starts by putting forward an initial argument and then the proponent and opponent alternate in attacking each other's previous arguments. The proponent wins if the opponent runs out of arguments to make a move. A dispute is represented by a dispute derivation in which dispute derivation trees are successively constructed by expanding the previous one by adding children to some frontier (or leaf) nodes following some legal move functions for the dispute parties. The initial partial derivation tree consists of only the root.

Definition 3.1 A dispute derivation tree (or simply derivation tree for short) \mathcal{T} for an argument A is defined as follows:

1. Every node of \mathcal{T} is labeled 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 from that of its parent. Arguments labeling children nodes attack arguments labeling their parent node.
2. The root is a proponent node labeled by A .

For example, various derivation trees are given in figure 1. Abusing the notation, for a node N labeled by A , we often refer to $Attack_A$ by $Attack_N$.

In the following, we introduce the notion of legal environments generalizing the well-known notions of legal move functions in the literature.

Definition 3.2 • A partial function F is called a *proponent* (resp *opponent*) legal move function if F assigns to pairs (N, T) of a frontier opponent (resp. proponent) node N in a derivation tree T an argument in (resp. a subset of) $Attack_N$. If $F(N, T)$ is not defined, we write $F(N, T) = \perp$.

- A pair of proponent and opponent legal move functions $\Phi = (\Phi_P, \Phi_O)$ is called a legal environment.

Examples of legal move functions are given in figure 1. For simplicity, if it is clear from context, we often write $\Phi(N, T)$ for $\Phi_P(N, T)$ if N is a proponent node, and for $\Phi_O(N, T)$ if N is an opponent node. We often refer to the leaf nodes in a derivation tree as frontier nodes to indicate that they are the ones to be expanded later.

Definition 3.3 Given a legal environment $\Phi = (\Phi_P, \Phi_O)$ and derivation trees T, T' . We say that T' is obtained from T by expansion of a frontier (leaf) node N in T wrt Φ if $\Phi(N, T)$ is defined and following condition holds:

- If N is an opponent node then a child node labeled by an argument selected in $\Phi_P(N, T)$ is added to N
- If N is proponent node then a set of child nodes labeled by arguments in $\Phi_O(N, T)$ are added to N . There is one-one correspondence between the children nodes of N and the arguments in $\Phi_O(N, T)$

In our model of dispute, when the opponent moves against a proponent argument, he delivers all attacks that are legal at once. This assumption simplifies the formal framework without affecting its generality. We discuss alternative strategies later.

Definition 3.4 • A dispute derivation wrt legal environment $\Phi = (\Phi_P, \Phi_O)$ is a sequence $T_0, \dots, T_i, \dots, T_n$ of derivation trees where T_0 consists of a single node and T_{i+1} is obtained from T_i by expansion of a frontier node N in T_i .

- A dispute derivation T_0, \dots, T_n terminates if for each frontier node N of the final tree T_n , $\Phi(N, T_n) = \emptyset$
- A terminating dispute derivation is said to be successful if all the frontier nodes of the final tree are proponent nodes.

The sequence T_0, T_1, T_2 in figure 1 is an example of a successful dispute derivation.

For a derivation tree T , $PRO(T)$, $OPP(T)$ denote respectively the sets of arguments labeling the proponent or opponent nodes in T . Arguments in $PRO(T)$ or $OPP(T)$ are often called respectively proponent or opponent arguments.

Definition 3.5 1. A derivation tree T is consistent if there is no argument labeling both a proponent node and an opponent node.

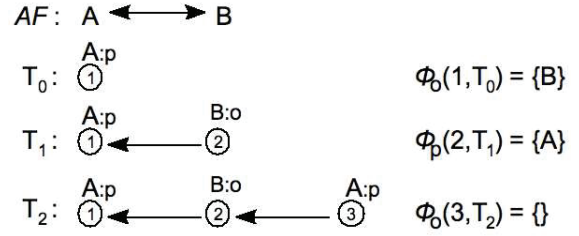


Figure 1:

2. A derivation tree is successful wrt legal environment Φ if it is the final tree of a successful dispute derivation wrt Φ .

Consider a frontier proponent node N in a derivation tree T such that $Attack_N \cap PRO(T) \neq \emptyset$. This implies immediately that the set of proponent arguments is not conflict-free as one of them attacks the argument labeling N . We certainly do not want to expand T at this node, i.e. $\Phi_O(N, T)$ should not be defined. Further in general, we want to consider only those legal environments guaranteeing that an expansion of consistent derivation trees leads to consistent ones. This requirement imposes that for frontier opponent node N , $\Phi_P(N, T) \cap OPP(T) = \emptyset$.

Definition 3.6 • A legal environment Φ is eligible iff following conditions are satisfied:

1. For each derivation tree T , and each frontier proponent node N of T , $\Phi_O(N, T) = \perp$ if $Attack_N \cap PRO(T) \neq \emptyset$.
2. If N is an opponent node then $\Phi_P(N, T) \subseteq Attack_N \setminus OPP(T)$

- An eligible legal environment is said to be fully defined if the "if" in the condition (1) above is replaced by "iff".

Examples of eligible legal environments are given in figure 2.

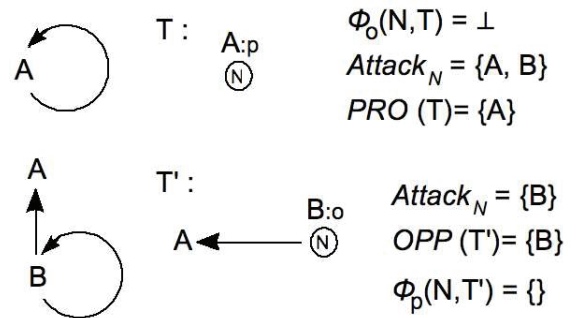


Figure 2:

It is not difficult to show that the eligibility of legal environment guarantees that only consistent derivation trees are generated in a dispute derivation.

Lemma 3.1 Let $\Phi = (\Phi_P, \Phi_O)$ be an eligible legal environment and T_0, \dots, T_n be a dispute derivation wrt Φ . Then for each i , T_i is consistent.

Proof By induction and left to the readers. \square

Definition 3.7 Let Φ be a legal environment.

1. Φ is said to be credulously (resp. groundedly) sound if the sets of proponent arguments of the successful derivation trees wrt Φ are admissible (resp. admissible subsets of the grounded extension).
2. Φ is said to be credulously (resp. groundedly) complete if for each credulously (resp. groundedly) accepted argument there is a successful derivation tree wrt Φ whose initial argument is A.
3. Φ is said to be terminating if there exists no infinite dispute derivation wrt Φ .

4 Legal Environments for Credulous Semantics

To ensure admissibility of proponent arguments, the legal environment should guarantee that all possible attacks against proponent arguments are accounted for. This is the case if for each proponent node N, $\Phi_O(N, T) = Attack_N$. But if some arguments in $Attack_N$ have been defeated by the proponent arguments then deploying them will not help the opponent winning the case. The opponent also does not need to deploy arguments already deployed by itself in previous steps. This insight is formalized in the following theorem in which a general class of credulously sound legal environments is introduced.

Theorem 4.1 Let $\Phi = (\Phi_P, \Phi_O)$ be an eligible legal environment such that for each derivation tree T, each proponent node N in T

$$Attack_N \setminus (Attacked_{PRO(T)} \cup OPP(T)) \subseteq \Phi_O(N, T)$$

whenever $\Phi_O(N, T)$ is defined.

Then Φ is credulously sound.

Proof Let T_0, \dots, T_n be a successful derivation wrt Φ . First we want to show that $PRO(T_n)$ counterattacks every attack against it. Let A be an argument attacking an argument B $\in PRO(T_n)$. Let N be a proponent node labelled by B. There are two cases:

1. N is a leaf node in T_n . Hence $\Phi_O(N, T_n) = \emptyset$. From $Attack_N \setminus (Attacked_{PRO(T)} \cup OPP(T_n)) \subseteq \Phi_O(N, T_n)$, it follows $Attack_N \setminus (Attacked_{PRO(T_n)} \cup OPP(T)) = \emptyset$, i.e. $Attack_N \subseteq (Attacked_{PRO(T_n)} \cup OPP(T))$. Hence $A \in (Attacked_{PRO(T_n)} \cup OPP(T))$. As all leaf nodes in T_n are proponent nodes, it follows that $OPP(T_n) \subseteq Attacked_{PRO(T_n)}$. A is hence attacked by $PRO(T_n)$.
2. N is an internal node N in T_n . Let T_i be the tree at which the opponent attacks N and expands T_i into T_{i+1} . If $A \in \Phi_O(N, T_i)$ then A labels one of the opponent child of N. As T_n is a successful tree, A has a proponent child in T_n . Hence A is attacked by $PRO(T_n)$. If $A \notin \Phi_O(N, T_i)$, then $A \in (Attacked_{PRO(T_n)} \cup OPP(T_n))$. As each opponent node in T_n has a proponent child, it follows that A is attacked by $PRO(T_n)$.

It remains to show that $PRO(T_n)$ is conflict-free. Suppose $PRO(T_n)$ is not conflict-free. An argument $A \in PRO(T_n)$ is said to be controversial if A attacks or is attacked by some argument in $PRO(T_n)$. Hence the set of controversial arguments in $PRO(T_n)$ is not empty.

We first show that no such argument labels a leaf node in T_n . Suppose the contrary that a controversial argument A labels a leaf node N in T_n . As $PRO(T_n)$ counterattacks every attack against it, it follows that there is some proponent argument in T_n attacking A. Hence $Attack_N \cap PRO(T_n) \neq \emptyset$. Therefore $\Phi_O(N, T_n) = \perp$ contradiction to the successfulness of T_n .

Therefore every controversial argument labels an internal node in T_n . Let m be the maximum number such that T_m is expanded at a proponent node N (by the opponent) labelled by controversial argument A. Further let $B \in PRO(T_n)$ such that B attacks A. From the definition of m, it follows that B labels a proponent node in T_m . Hence $Attack_N \cap PRO(T_m) \neq \emptyset$. Therefore $\Phi_O(N, T_n) = \perp$ contradiction to the successfulness of T_n .

Therefore the set of controversial arguments is empty. Contradiction. Hence $PRO(T_n)$ is conflict-free.

\square

Theorem 4.1 states formally that the soundness of a legal environment will not be affected if filtering mechanisms are deployed to filter out some or all arguments in $Attacked_{PRO(T)} \cup OPP(T)$. An important insight gained from this theorem is that the soundness of a legal environment depends only on the legal moves of the opponent. The following theorem shows that the legal moves of the proponent determine the completeness.

Theorem 4.2 Let $\Phi = (\Phi_P, \Phi_O)$ be an eligible, credulously sound and terminating legal environment. Further suppose that Φ is fully defined and for each derivation tree T, each opponent node N in T

$$Attack_N \setminus Conf(T) \subseteq \Phi_P(N, T)$$

where $Conf(T)$ consists of all self-attacking arguments and all arguments attacking or being attacked by arguments in $PRO(T)$.

Then Φ is credulously complete.

Proof Let A be a credulously accepted argument and S be an admissible set of arguments and $A \in S$. We construct by induction on i a successful derivation $T_0, \dots, T_i, \dots, T_n$ such that A labels the root of the trees and $PRO(T_i) \subseteq S$.

If T_i is successful then we are done, the theorem is proved. Suppose now that T_i is not successful. Select a frontier node N of T_i .

Suppose N is a opponent node. Then from the admissibility of S and $PRO(T_i) \subseteq S$, it follows that $Attack_N \cap S \neq \emptyset$. Because S is conflict-free and $PRO(T_i) \subseteq S$, it follows that $S \cap Conf(T_i) = \emptyset$. Therefore $(Attack_N \setminus Conf(T_i)) \cap S \neq \emptyset$. Hence there is an argument $B \in S \cap \Phi_P(N, T_i)$. Select B and expand T_i by adding a child to N labelled by B.

Suppose N is a proponent node. As the argument labelling N belongs to S and S is admissible and $PRO(T_i) \subseteq S$, it follows that $Attack_N \cap PRO(T_i) = \emptyset$. Since Φ is fully defined, it follows that $\Phi_O(N, T_i) \neq \perp$. If $\Phi_O(N, T_i) \neq \emptyset$ then expand T_i at N.

As Φ is terminating, the process must stop sometimes. It follows that all the frontier nodes are proponent nodes. The theorem is proved. \square

The following theorem shows that one can guarantee the termination of dispute by not allowing the opponent to repeat his moves.

Theorem 4.3 *Let $\Phi = (\Phi_P, \Phi_O)$ be an eligible legal environment such that*

$$\Phi_O(N, T) \subseteq \text{Attack}_N \setminus \text{AOPP}(N, T)$$

where $\text{AOPP}(N, T)$ contains all arguments labeling opponent nodes on the path from the root to N .

Then Φ is terminating.

Proof Suppose Φ is not terminating and $\Phi_O(N, T) \subseteq \text{Attack}_N \setminus \text{AOPP}(N, T)$. Let T_0, \dots, T_i, \dots be a nonterminating derivation. Let T be the limit of this sequence of trees. T is hence infinite. There is then an infinite path in T . There are hence infinite opponent nodes on this path. Since the set of arguments is finite, there are many opponent nodes labelled by the same argument. Contradiction. \square

As we have discussed, a key insight of theorems 4.1, 4.2 is that the credulous soundness of a legal environment is determined by opponent legal moves while its completeness depends on the legal moves of the proponent. This insight is best illustrated by the special case where $\Phi_O(N, T) = \text{Attack}_N$ whenever it is defined. It turns out that in this case, the legal environment is even groundedly sound.

Lemma 4.1 *Let Φ be an eligible legal environment such that $\Phi_O(N, T) = \text{Attack}_N$ whenever $\Phi_O(N, T)$ is defined. Then Φ is groundedly sound.*

Proof Let Φ' be a legal environment such that $\Phi'_O(N, T) = \text{Attack}_N$ whenever it is defined and $\Phi'_P(N, T) = \text{Attack}_N \setminus \text{OPP}(T)$. It is easy to see that each successful derivation wrt Φ is a successful derivation wrt Φ' . Let T be a successful derivation tree wrt Φ' . It is easy to show by induction on the height of T that $\text{PRO}(T)$ is a subset of the grounded extension. \square

5 Legal Environments for Grounded Semantics

We are now turning our attention to legal environments for grounded semantics. Lemma 4.1 points out that the opponent could do nothing wrong if he/she keeps bringing up all possible counter-evidences to whatever the proponent says. The legal environment is groundedly sound independent of whatever the proponent does. The best the proponent could do in such cases is to make sure that it can win if the initial argument is groundedly accepted, i.e. to ensure that its legal move function guarantees completeness. This is the case if $\Phi_P(N, T) = \text{Attack}_N \setminus \text{OPP}(T)$. But this environment is not terminating. It turns out that to ensure both termination and completeness, the proponent should not repeat arguments put forward by itself previously. Though the legal environment just described is groundedly sound, complete and terminating, it could be quite inefficient as the opponent could

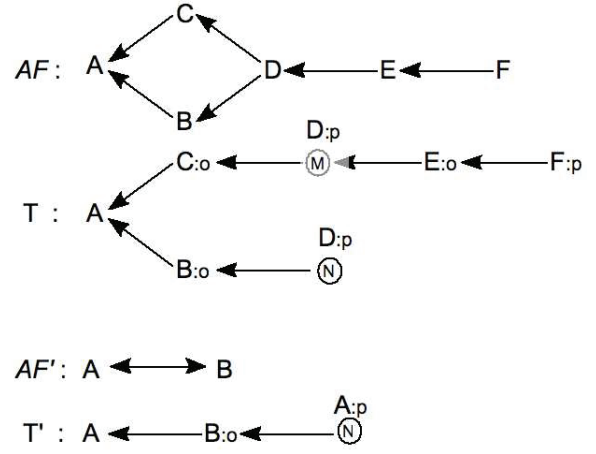


Figure 3:

repeat many arguments that have been defeated before as the following example illustrates.

Consider the derivation tree T in figure 3. It would be wasteful if argument D at node N should be defeated again. We hence expect from an efficient legal environment that $\Phi_O(N, T) = \emptyset$. But in contrast to the credulous case, not all previous opponent arguments could be filtered out as illustrated by derivation tree T' in figure 3. If the opponent argument B is filtered out at N i.e. $\Phi_O(N, T) = \emptyset$ then T' would be a successful derivation tree though A is not groundedly accepted.

The examples suggest that we should allow the opponent to filter out previous opponent arguments except those lying on the path from the root to the considered node, i.e. $\text{Attack}_N \setminus (\text{OPP}(T) \setminus \text{AOPP}(N, T)) \subseteq \Phi_O(N, T)$. Unfortunately, this constraint is not strong enough to guarantee the grounded soundness of the legal environment as the following example shows:

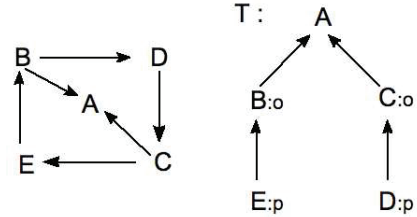


Figure 4:

Note that when an argument labels exactly one node in a derivation tree, we identify the argument and the node for simplicity.

In figure 4, $\text{AOPP}(E, T) = \{B\}$, $\text{Attack}_E = \{C\}$, $\text{OPP}(T) = \{B, C\}$. Hence $\text{Attack}_E \setminus (\text{OPP}(T) \setminus \text{AOPP}(E, T)) = \emptyset$. $\text{AOPP}(D, T) = \{C\}$, $\text{Attack}_D = \{B\}$. Hence $\text{Attack}_D \setminus (\text{OPP}(T) \setminus \text{AOPP}(D, T)) = \emptyset$. Hence $\Phi_O(E, T) = \Phi_O(D, T) = \emptyset$ satisfies the constraint. T would be a successful derivation tree though A is not groundedly accepted. The reason is that B, E, C, D, B is a cycle

in the argumentation framework and this cycle makes the filtering of B,C groundedly unsound. It turns out filtering is groundedly sound if it avoids filtering out previous opponent arguments lying in cycles like B, E, C, D, B . This motivates the following definition.

Definition 5.1 Let T be a derivation tree.

1. Let N be a proponent node and M be an opponent node and A be the argument of M . M is an adopted child of N in T if A attacks the argument of N and N has no child labelled by A ¹.
2. A generalized path in T is a sequence of nodes N_0, \dots, N_k , $k \geq 1$ such that N_{i+1} is either a child or an adopted child of N_i .

A generalized path is a cycle if the first and last nodes coincide.

In figure 3, E is an adopted child of N in T and the sequence B, N, E is a generalized path. B, A, B is a cycle in T' while T contains no cycle. In figure 4, the sequence B, E, C, D, B is a cycle in T .

Definition 5.2 Let N be a frontier proponent node in T , and A be an argument attacking N . A is said to be a taboo for filtering out by opponent at N if there is an cycle N_0, \dots, N_n in T such that $N = N_i$ and A labels N_{i+1} for some i .

The set of taboos for filtering out at N by opponent in T is denoted by $TABOO(N, T)$.

In figure 3, $TABOO(N, T) = \emptyset$. Hence it is possible to filter out the argument E attacking D at node N . In contrast, $TABOO(N, T') = \{B\}$. Hence it is not possible to filter out B at N . T' is hence not a successful derivation tree while T is. Similarly, in figure 4, $TABOO(E, T) = \{C\}$, $TABOO(D, T) = \{B\}$.

Theorem 5.1 Let $\Phi = (\Phi_P, \Phi_O)$ be an eligible legal environment such that for each derivation tree T , each frontier proponent node N labeled by A ,

$$Attack_N \setminus (OPP(T) \setminus TABOO(N, T)) \subseteq \Phi_O(N, T)$$

whenever $\Phi_O(N, T)$ is defined.

Then Φ is groundedly sound.

Proof Let T_0, \dots, T_n be a successful dispute derivation wrt Φ . A cycle is said to be internal if it does not contain any frontier node. We prove by induction that for each i , there is no internal cycle in T_i .

Suppose that there is no internal cycle in T_i but some internal cycles in T_{i+1} . Let $Cyc = N_0, \dots, N_n$ be an internal cycle in T_{i+1} labelled by A_0, \dots, A_n . As there is no internal cycle in T_i , Cyc must contain a frontier node in T_i . Since Cyc is an internal cycle in T_{i+1} , Cyc contains exactly one frontier node in T_i that is the frontier node N at which T_i is expanded. Let $N = N_k$. If N is an opponent node in T_i then N_{k+1} is a child of N . Hence N_{k+1} is a frontier node of T_{i+1} . Contradiction to the assumption that Cyc is an internal cycle in T_{i+1} .

Hence N must be a proponent node in T_i . As Cyc is an internal cycle, N_{k+1} is not a frontier node in T_{i+1} . Hence

¹Note that we do not define adopted children for opponent nodes

N_{k+1} is an adopted child of N in T_i and T_{i+1} . From definition 5.2, it follows that $A_{k+1} \in TABOO(N, T_i)$.

As N_{k+1} is an adopted child of N also in T_{i+1} , there is no child of N in T_{i+1} labelled by A_{i+1} . It follows $A_{i+1} \notin \Phi_O(N, T_i)$. From the properties of Φ , it follows that $A_{i+1} \notin Attack_N \setminus (OPP(T_i) \setminus TABOO(N, T_i))$. Therefore $A_{i+1} \in (OPP(T) \setminus TABOO(N, T_i))$. Hence $A_{i+1} \notin TABOO(N, T_i)$. Contradiction.

We prove now that there is no cycle in T_n . Suppose that there is a cycle Cyc in T_n . As T_n has no internal cycle, Cyc must contain a frontier node, say N . Therefore $TABOO(N, T_n) \neq \emptyset$. Hence $\Phi_O(N, T_n) \neq \emptyset$. Contradiction.

We prove now that for all nodes M in T_n , if M is a proponent (resp opponent) node then the argument labelling M is groundedly accepted (resp rejected, i.e. attacked by a groundedly accepted argument). The proof is by induction on the rank of the nodes in T_n defined as follows..

The rank of nodes without any kind of children is 0. The rank of a node M is equal 1 plus the maximum of the ranks of the children or adopted children of M .

We show by induction that the nodes of even (resp odd) ranks are proponent (resp. opponent) nodes. Obviously that nodes of rank 0 are proponent nodes. Suppose now that the property holds for all nodes of rank $\leq i$. Let M be a node of rank $i + 1$. As one of the children or adopted children of M is of rank i , M is a proponent (resp opponent) node if i is odd (resp even) from the induction hypothesis.

We could easily prove by induction now that nodes of even (resp odd) ranks are groundedly accepted (resp rejected) based on the following property:

Let N be proponent node in T_n . Then $Attack_N = \{A \mid A \text{ labels a child or an adopted child of } N\}$.

Let T_i be the tree at which N is selected for expansion. Suppose A is an argument attacking N and A does not label a child of N . Hence $A \notin \Phi_O(N, T_i)$. Therefore $A \in (OPP(T_i) \setminus TABOO(N, T_i))$. Hence there is an opponent node M in T_i labeled by A . As A does not label a child of N , M is an adopted child of N . \square

Theorem 5.2 Let $\Phi = (\Phi_P, \Phi_O)$ be an eligible and groundedly sound legal environment such that for each derivation tree T , each frontier opponent node N in T

$$Attack_N \setminus (Conf(T) \cup APRO(N, T)) \subseteq \Phi_P(N, T)$$

$$\subseteq Attack_N \setminus (OPP(T) \cup APRO(N, T))$$

where $APRO(N, T)$ contains all proponent arguments lying on the path from the root to N .

Then Φ is groundedly complete and terminating

Proof For groundedly accepted (resp rejected) argument A , define $rank(A)$ to be the smallest number i such that $A \in \mathcal{F}^i(\emptyset)$ (resp. A is attacked by $\mathcal{F}^i(\emptyset)$). We prove the theorem by induction on $rank(A)$.

Let A be a groundedly accepted argument. A successful derivation T_0, \dots, T_n of A is constructed by induction as follows:

If T_k is obtained from T_{k-1} by expanding an opponent node then select an argument belonging to $\mathcal{F}^j(\emptyset)$ where j is the rank of the argument labelling the node that is expanded.

It is not difficult to see that constructing T_i 's that way will produce a successful derivation wrt Φ . \square

6 Discussion and Conclusion

In [Cayrol,Doutre,Mengin, 2003], two proof procedures called ϕ_1, ϕ_2 are given for credulous semantics. ϕ_1 corresponds to a sequentialization of a legal environment with $\Phi_{1,O}(N, T) = Attack_N \setminus (Attacked_{PRO(T)} \cup OPP(T))$ and $\Phi_{1,P}(N, T) = Attack_N \setminus Conf(T)$. ϕ_2 corresponds to $\Phi_{2,O}(N, T) = Attack_N \setminus Attacked_{APRO(N,T)}$ and $\Phi_{2,P}(N, T) = Attack_N \setminus Conf(N, T)^2$. [Modgil,Caminada, 2009] gives also several proof procedures for credulous semantics. They are somewhat similar to the ϕ_2 procedure above. For example, one of them corresponds to the legal environment: $\Phi_{3,O}(N, T) = Attack_N \setminus AOPP(N, T)$ and $\Phi_{3,P}(N, T) = Attack_N \setminus Conf(N, T)$. While ϕ_1 is sound, complete and terminating, ϕ_2, Φ_3 may not be eligible and hence are sound only if their successful derivation trees are consistent. But no condition is given under what conditions is the consistency of the successful derivation trees is guaranteed. [Modgil,Caminada, 2009] gives also proof procedures for grounded semantics. The basic one corresponds to the sound and complete and terminating legal environment: $\Phi_O(N, T) = Attack_N$ and $\Phi_P(N, T) = Attack_N \setminus APRO(N, T)$.

In general, the opponent does not need to unleash all of his attacks at once. By dropping the "one-one-correspondence" condition in the second bullet of the definition 3.3 and requires an extra condition in theorem 4.1 and 5.1 that $\Phi_O(N, T)$ does not contain any arguments labelling children of N and allowing the opponent to attack at all proponent nodes as long as there is an option for him there, we obtain new dispute procedures with the same characteristics.

A possible application of the insight that the soundness of a dispute is determined by the opponent moves while its completeness by the proponent ones, is a modular methodology for designing dispute procedures. The design consists of four tasks: ensuring the eligibility of the legal environment, ensuring the soundness by looking at opponent moves and guaranteeing completeness by looking at proponent moves and enforcing terminating by forbidding one of the participant not to repeat his moves.

A possible interesting application of this paper lies in the design of proof procedures for logic-based argumentation systems [Dung,Kowalski,Toni, 2006; Dung,Mancarella,Toni, 2007; Eshghi,Kowalski, 1989; Kakas,Toni, 1999; Dung, 1995a] by translating automatically proof procedures in abstract argumentation into ones for the latter systems.

7 Acknowledgements

We thank the referees for constructive comments and criticisms. This work was partially funded by the Sixth Framework IST program of the European Commission under the 035200 ARGUGRID project.

²Conf(N,T) consists of all arguments that selfattack or attacked or being attacked by proponent arguments on the path from the root to N

References

- [Bondarenko,Dung,Toni, 1997] A Bondarenko, P M Dung, R A Kowalski, and F Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93, 1997.
- [Cayrol,Doutre,Mengin, 2003] Claudette Cayrol, Sylvie Doutre, and Jerome Mengin. On decision problems related to the preferred semantics for argumentation frameworks. *Journal of Logic and Computation*, 13, 2003.
- [Dung,Kowalski,Toni, 2006] Phan Minh Dung, Robert A. Kowalski, and Francesca Toni. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence*, 170, 2006.
- [Dung,Mancarella,Toni, 2007] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 2007.
- [Dung, 1995a] Phan Minh Dung. An argumentation theoretic foundation for logic programming. *Journal of Logic Programming*, 1995.
- [Dung, 1995b] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77, 1995.
- [Dunne,Bench-Capon, 2003] Paul E. Dunne and Trevor J M. Bench-Capon. Two party immediate response disputes: Properties and efficiency. *Artificial Intelligence*, 149, 2003.
- [Eshghi,Kowalski, 1989] K. Eshghi and R.A. Kowalski. Abduction compared with negation by failure. In *In Proceedings of the Sixth International Conference on LP*, 1989.
- [Jakobovits,Vermeir, 1999] H. Jakobovits and D. Vermeir. Dialectic Semantics for Argumentation Frameworks In *Proc ICAIL'99, ACM Press*, 1999.
- [Kakas,Toni, 1999] T Kakas and F Toni. Computing argumentation in logic programming. *Journal of Logic and Computation*, 1999.
- [Modgil,Caminada, 2009] S Modgil and M Caminada. *Argumentation in AI*, chapter Proof Theories and Algorithms for Abstract Argumentation. Springer-Verlag, 2009.
- [Prakken, 2005] Henry Prakken. Coherence and Flexibility in dialogue games for argumentation In *Journal of Logic and Computation*, 2005.
- [Verheij, 2007] Bart Verheij. A Labeling Approach to the Computation of Credulous Acceptance In *IJCAI*, 2007.
- [Vreeswijk,Prakken, 2000] Gerard Vreeswijk and Henry Prakken. Credulous and sceptical argument games for preferred semantics. In *JELIA*, 2000.
- [Vreeswijk, 2006] Gerard Vreeswijk. An Algorithm to compute minimally grounded and admissible defence sets in argument systems. In *COMMA*, 2006.