Argument & Computation 13 (2022) 121–157
DOI 10.3233/AAC-200546
IOS Press

Infinite arguments and semantics of dialectical proof procedures

Phan Minh Thang [a,*], Phan Minh Dung [b] and Jiraporn Pooksook [c]
[a] *International College of Burapha University, Burapha University, Thailand*
E-mail: thangpm@gmail.com
[b] *Department of ICT, Asian Institute of Technology, Thailand*
E-mail: dung.phanminh@gmail.com
[c] *Department of Electrical and Computer Engineering, Naresuan University, Thailand*
E-mail: jirapornpook@nu.ac.th

Abstract. We study the semantics of dialectical proof procedures. As dialectical proof procedures are in general sound but not complete wrt admissibility semantics, a natural question here is whether we could give a more precise semantical characterization of what they compute. Based on a new notion of infinite arguments representing (possibly infinite) loops, we introduce a stricter notion of admissibility, referred to as strict admissibility, and show that dialectical proof procedures are in general sound and complete wrt strict admissibility.

Keywords: Dialectical proof procedure, infinite arguments, admissibility semantic, soundness and completeness

1. Introduction

Argumentation is a reasoning model in which reasons for conclusions that are drawn for resolving conflicts are given explicitly. While abstract argumentation studies the acceptance of arguments, structured argument systems like assumption-based argumentation or defeasible logic programming provide frameworks for structuring arguments based on assumptions and rules [5,6,13,23,24,29,31]. Argument-based systems are becoming increasingly popular due to their intuitive appeal to the ways humans perform their practical and daily reasoning [2,3,19,32,38].

Dialectical proof procedures for argumentation have been developed both for abstract argumentation [7,18,20,28,37,39] and for rule-based instances of it like logic programming [12,14,21] or assumption-based argumentation [15–17,22,35,36]. A proof procedure for assumption-based argumentation could be viewed as an integration of the dialectical procedures of abstract argumentation with the process of argument constructions where the later could get into a non-terminating loop leading to the incompleteness wrt the admissibility semantics.

A natural question here is: can we give a precise semantical characterization of what dialectical proof procedures compute?

The following example illustrates this point (as logic programming is an instance of assumption-based argumentation where the contrary of a negation-as-failure assumption not_δ is δ, we will represent our examples in logic programming for convenience).

[*] Corresponding author. E-mail: thangpm@gmail.com.

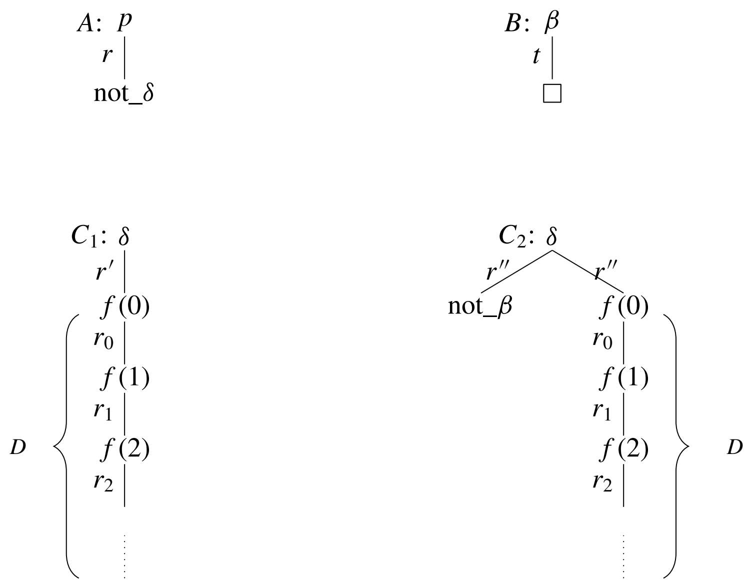

Fig. 1. Arguments of $\mathcal{F}_1$ and $\mathcal{F}_2$.

Example 1. Consider two logic programs below.

$$\mathcal{F}_1 : r : p \leftarrow \text{not\_}\delta \qquad r' : \delta \leftarrow f(0) \qquad r_n : f(n) \leftarrow f(n+1), \quad n \geqslant 0$$
$$t : \beta \leftarrow$$
$$\mathcal{F}_2 : r : p \leftarrow \text{not\_}\delta \qquad r'' : \delta \leftarrow \text{not\_}\beta, f(0) \qquad r_n : f(n) \leftarrow f(n+1), \quad n \geqslant 0$$
$$t : \beta \leftarrow$$

The semantics of $\mathcal{F}_1$ and $\mathcal{F}_2$ are determined by the arguments illustrated in Fig. 1.

The unique argument supporting p in both frameworks is A supported by assumption not_δ. It is obvious that δ is not supported as there is no (finite) argument supporting it. Hence there are no attacks (by finite arguments) against both A and B in both frameworks. $\{A, B\}$ is consequently admissible wrt both frameworks. But dialectical reasoning engines like the proof procedures for logic programming or assumption-based argumentation in [14–17,21,22,26,35,36] fail to deliver A wrt the first framework $\mathcal{F}_1$ as they could not overcome the non-termination of the process to construct an argument supporting δ due to the "infinite-loop" represented by D, though all of them deliver A wrt the second framework $\mathcal{F}_2$ despite the presence of the same loop D.

The distinct behavior of the dialectical proof procedures wrt the two frameworks $\mathcal{F}_1$, $\mathcal{F}_2$ suggests that to fully understand their semantics, we need to consider the effects of "infinite loops" on their behavior. In this paper, we accomplish this by introducing a new notion of infinite arguments to represent such "infinite loops".

For framework $\mathcal{F}_1$, there is an infinite argument for δ represented by the infinite proof tree C_1. This argument can not be attacked as it is not based on any assumption.

For framework $\mathcal{F}_2$, there is an infinite argument for δ represented by the infinite proof tree C_2. This argument is based on assumption not_β and hence is attacked by the argument B.

Therefore, if infinite arguments are taken into account, the set $\{A, B\}$ is not admissible wrt $\mathcal{F}_1$ while it is still admissible wrt $\mathcal{F}_2$.

The example suggests that a stronger notion of admissibility taking into account also the effects of infinite arguments, is needed to characterize the semantics of dialectical proof procedures.

As argument C_1 can not be attacked by any other argument wrt $\mathcal{F}_1$, should it be accepted and δ concluded?

As infinite loops represent a kind of unfinished, inconclusive business, C_1 can not be viewed as a support of δ.

In general, a finite argument has two roles: As a direct support for its conclusion and as an attacker against certain arguments. In contrast, infinite arguments have only one role as attacker against some other arguments. They can not support their conclusion.

To capture this peculiar character of infinite arguments, we model infinite arguments as self-attacking arguments. This very simple idea presents a declarative view of the "spoiling role" of infinite loops: A presence of infinite loops in a computation prevents it to come to a conclusion.

We make three key contributions in this paper:

- The introduction of infinite arguments and a new stricter notion of admissibility;
- We show the soundness and completeness of dialectical proof procedures wrt the new notion of strict admissibility for general assumption based frameworks. To accomplish this goal, we introduce a new dialectical proof procedure that are based explicitly on proof trees (i.e. arguments and partial arguments).[1]
- Last but not least, we introduce a new view of proof trees (and arguments) as sets of the partial proofs represented by paths from the root to their nodes. This view of "trees as sets" of partial proofs allows simple characterizations of dispute derivation and simplifies in no small amount the technical machinery needed in the proofs of soundness and completeness of dialectical procedures.

The paper also offers an interesting conceptual insight about the "spoiling" role of arguments that are unacceptable (like the infinite arguments in our case or more abstractly the self-attacking arguments in abstract argumentation). As the problem of non-termination is not decidable, such arguments can not be dismissed and practical rule-based argumentation systems need to take care of them. In the context of assumption-based argumentation, these "spoiler (infinite) arguments" lead to strict admissibility as the semantics of dialectical procedures.

The paper is organized in 8 sections including the Introduction. In the following section, we recall the basic notions of abstract and assumption-based argumentation as well as introduce the infinite arguments together with a new notion of strict admissibility. We then present a proof-tree based dialectical proof procedure in Section 3. We show the soundness of the proof-tree based procedure in Section 4 and its completeness in Section 5. We introduce a flattened version of the proof-tree based procedure together with its soundness and completeness in Section 6. We then conclude the paper with a short discussion in Section 7. The last section is the Appendix.[2]

---

[1] The new procedure could be viewed as a full embracement of the idea of explainable AI in dialectical procedures in which we not only state which assumptions provide support and defence for a conclusion, but also explicitly state which arguments are employed to accomplish such tasks.

[2] An extended abstract of this paper has been published in [34].

2. Assumption-based argumentation with infinite arguments

2.1. Abstract argumentation

An argumentation framework [13] is a pair AF = (AR, att), where AR is a set of arguments, and *att* is a binary relation over AR representing the attack relation between the arguments where $(A, B) \in$ att means that A attacks B. A set S of arguments attacks an argument A if some argument in S attacks A.

A set S of arguments is *conflict-free* iff it does not attack itself. S is *self-defensible* iff S attacks each argument attacking S. S is *admissible* iff S is conflict-free and self-defensible. S is a *preferred extension* iff S is maximally (wrt set inclusion) admissible.

An argument A is said to be credulously accepted iff it is contained in at least one preferred extension. It follows that an argument is credulously accepted iff it belongs to an admissible set of arguments.

2.2. Assumption-based argumentation

Given a logical language $\mathcal{L}$, a *standard assumption-based argumentation (ABA) framework* [5] is a triple $\mathcal{F} = (\mathcal{R}, \mathcal{A}, \overline{})$ where $\mathcal{R}$ is a set of inference rules of the form $l_0 \leftarrow l_1, \ldots l_n$ ($n \geqslant 0$ and $l_1, \ldots, l_n \in \mathcal{L}$), and $\mathcal{A} \subseteq \mathcal{L}$ is a set of assumptions, and $\overline{}$ is a (total) one-one mapping from $\mathcal{A}$ into $\mathcal{L} \setminus \mathcal{A}$, where $\overline{x}$ is referred to as the *contrary* of x, and assumptions in $\mathcal{A}$ do not appear in the heads of rules (see Remark 1).[3]

An ABA framework $\mathcal{F} = (\mathcal{R}, \mathcal{A}, \overline{})$ is *finitary* if for each sentence $\sigma \in \mathcal{L}$, the set of rules of the form $\sigma \leftarrow l_1, \ldots l_n$ ($n \geqslant 0$ and $l_1, \ldots, l_n \in \mathcal{L}$), is finite.

Logic programming is a well-known instance of standard ABA where the contrary of a negation-as-failure assumption not_p is p.

Remark 1. For each rule r of the form $l_0 \leftarrow l_1, \ldots l_n$, l_0 and the set $\{l_1, \ldots, l_n\}$ are referred respectively as the head and the body of r and denoted by $\text{hd}(r)$, $\text{bd}(r)$.

Convention 1. From now on until the end of the paper,

- we restrict our consideration to standard ABA. Hence whenever we refer to an ABA framework, we mean a *standard one*, and
- we assume an arbitrary but fixed finitary standard assumption-based argumentation framework $\mathcal{F} = (\mathcal{R}, \mathcal{A}, \overline{})$.

Definition 1 (Partial proof)**.** Given an ABA $\mathcal{F}$, a partial proof supporting σ_0 (wrt $\mathcal{F}$) is a finite sequence of the form

$$(\text{root}, \sigma_0).(r_1, \sigma_1). \ldots .(r_n, \sigma_n)$$

where $r_i \in \mathcal{R}$, $i \geqslant 1$ such that $\sigma_{i-1} = \text{hd}(r_i)$ and $\sigma_i \in \text{bd}(r_i)$. If $\text{bd}(r_i) = \emptyset$ then $\sigma_i = \text{true}$.

[3] In non-standard ABA frameworks [22], the contrary $\overline{\alpha}$ of an assumption α could be a set. Such non-standard frameworks could be translated into equivalent standard ones by introducing a new atom α' for each assumption α and i) define α' as the contrary of α; and ii) for each $\delta \in \overline{\alpha}$, add a new rule: $\alpha' \leftarrow \delta$ to $\mathcal{R}$.

It is not difficult to see that the new framework is equivalent to the old one. Hence focusing on standard ABA does not cause any loss of generality while simplifying the technical machinery in no small amount.

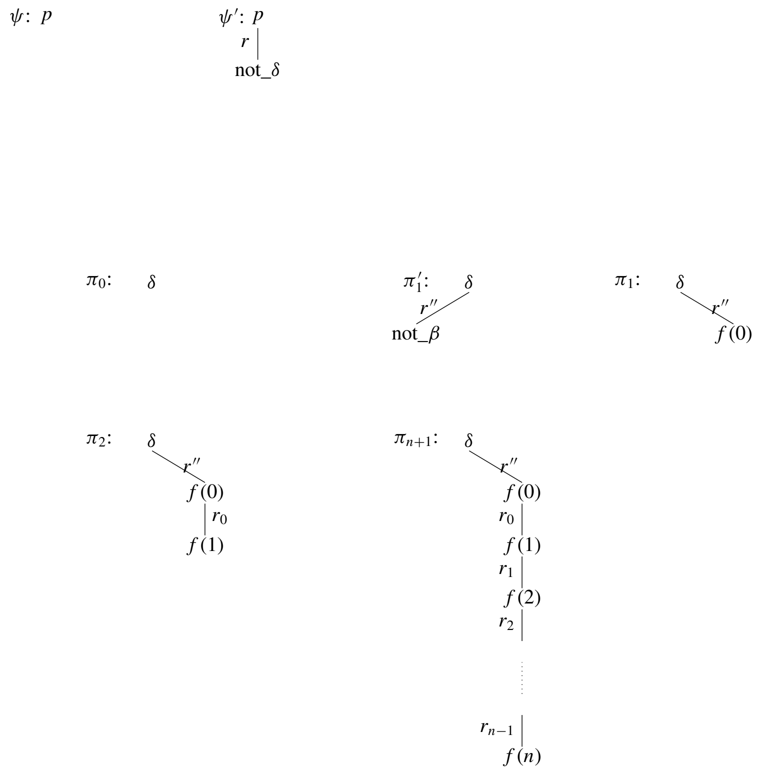

Fig. 2. A graphical representation of partial proofs of Example 2.

Example 2. Consider the argumentation framework $\mathcal{F}_2$ in Example 1.

Some partial proofs supporting p and δ are given below and illustrated in Fig. 2.

$$\psi = (\text{root}, p)$$

$$\psi' = (\text{root}, p).(r, \text{not\_}\delta)$$

$$\pi_0 = (\text{root}, \delta)$$

$$\pi'_1 = (\text{root}, \delta).\big(r'', \text{not\_}\beta\big)$$

$$\pi_1 = (\text{root}, \delta).\big(r'', f(0)\big)$$

$$\pi_2 = (\text{root}, \delta).\big(r'', f(0)\big).\big(r_0, f(1)\big)$$

$$\ldots$$

$$\pi_{n+1} = (\text{root}, \delta).\big(r'', f(0)\big).\big(r_0, f(1)\big)\ldots\big(r_{n-1}, f(n)\big)$$

We next define partial proof trees where we identify the nodes in a proof tree with the partial proofs representing the unique paths from the root to them. This allows us to treat proof trees as sets simplifying the technical machinery for presenting and understanding dialectical procedures significantly.

Definition 2 (Partial proof trees)**.** A *partial proof tree* (or just *proof tree* for simplification) T for a sentence σ_0 wrt $\mathcal{F}$ is a non-empty set of partial proofs supporting σ_0 wrt $\mathcal{F}$ such that for each partial proof

$$\pi \equiv (\text{root}, \sigma_0).(r_1, \sigma_1). \dots .(r_n, \sigma_n), \quad n > 0$$

from T, the following properties hold:

- the partial proof $\pi' \equiv (\text{root}, \sigma_0).(r_1, \sigma_1). \dots .(r_{n-1}, \sigma_{n-1})$ also belongs to T and is referred to as the unique parent of π whereas π is referred to as a child of π';
- Every partial proof of the form $\pi'.(r_n, \sigma')$ with $\sigma' \in \text{bd}(r_n)$, also belongs to T and is a child of π';
- π' has no other children.

σ_0 is often referred to as the *conclusion* of T, denoted by $\text{Cl}(T)$ while the partial proof (root, σ_0) is referred to as the *root* of T.

Example 3. Continue from Example 2. Some partial proof trees for p and δ are given below and illustrated in Fig. 3.

$$A_0 = \{\psi\}$$
$$A = \{\psi, \psi'\}$$
$$C_0 = \{\pi_0\}$$
$$C_{20} = \{\pi_0, \pi_1, \pi_1'\}$$
$$C_{21} = \{\pi_0, \pi_1', \pi_1, \pi_2\}$$
$$C_2 = \{\pi_0, \pi_1', \pi_1, \pi_2, \pi_3, \dots, \pi_n, \dots\}$$

Note that $\{\pi_0, \pi_1\}$ and $\{\pi_0, \pi_1'\}$ are not partial proof trees.

Remark 2. For convenience, we often refer to a partial proof tree without mentioning its conclusion if there is no possibility for misunderstanding.

Notation 1 (Nodes in partial proof trees)**.** Abusing the notation for convenience, we often refer to a partial proof $(\text{root}, \sigma_0).(r_1, \sigma_1). \dots .(r_n, \sigma_n)$ in a proof tree T as a *node labeled by* σ_n in T.

Notation 2. Let T be a partial proof tree and S be a set of partial proof trees,

- A node N in T is said to be a leaf of T if N has no children.
  A leaf N of T is said to be final if N is labeled by true or by an assumption. N is non-final if it is not final.
- The *support* of T, denoted by $\text{Sp}(T)$, is the set of all sentences labeling the leaves of T and different to true.
  The *union of supports* of proof trees in S is denoted by $\text{Sp}(S)$.
- The *set of all assumptions* appearing in T is denoted by $\text{Ass}(T)$.
  The *set of all assumptions appearing* in proof trees in S is denoted by $\text{Ass}(S)$.
- The set of conclusions of proof trees in S is denoted by $\text{Cl}(S)$.

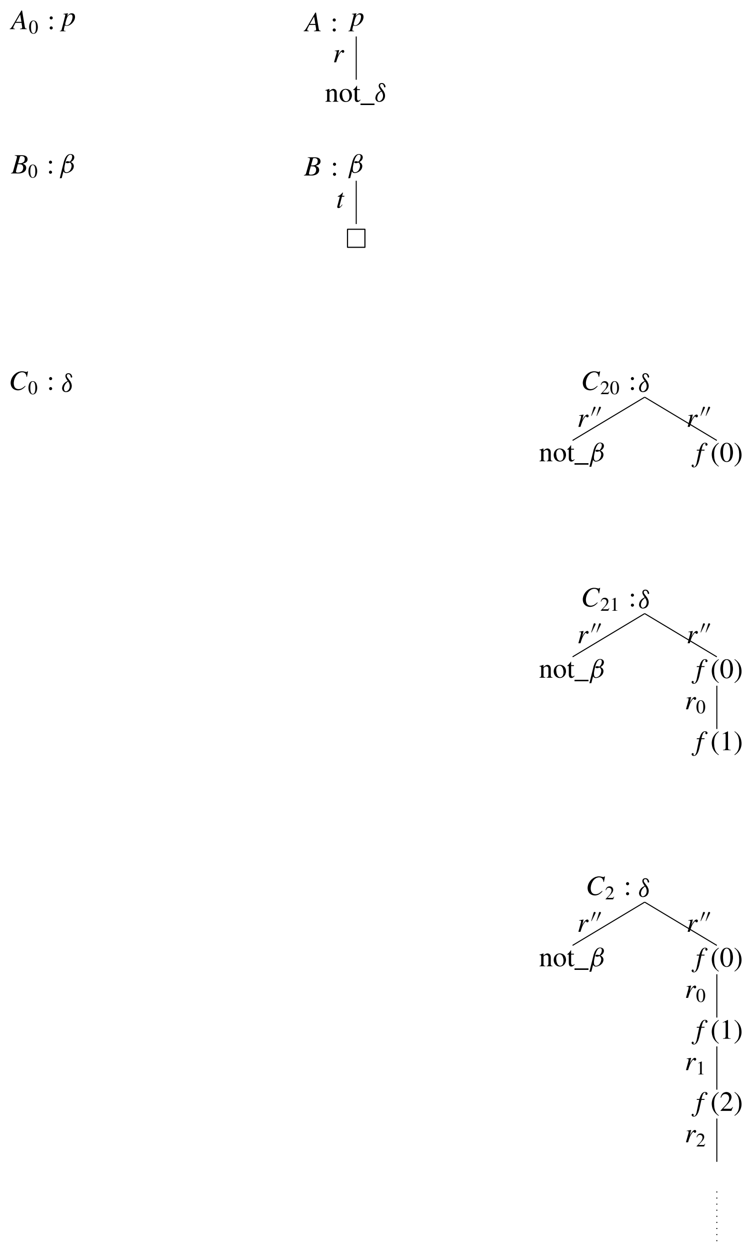

Fig. 3. Some partial proof trees from Example 3.

Consider the partial proof tree C_{20} in Fig. 3, the partial proof $\pi_0 = (\text{root}, \delta)$ is the parent of the node $\pi_1 = (\text{root}, \delta).(r'', f(0))$ and the node $\pi_1' = (\text{root}, \delta).(r'', \text{not\_}\beta)$. Also, π_1 and π_1' are the children of π_0.

The support of A is $\{\text{not\_}\delta\}$ while the supports of C_{20}, C_{21}, C_2 are $\{\text{not\_}\beta, f(0)\}$, $\{\text{not\_}\beta, f(1)\}$ and $\{\text{not\_}\beta\}$ respectively.

Definition 3 (Arguments)**.**

- A *full proof tree* is a partial proof tree whose support consists only of assumptions.[4]
- An *argument* for α is a full proof tree for α.
- The set of all arguments wrt the ABA framework $\mathcal{F}$ is denoted by $AR_{\mathcal{F}}$ while the set of all finite arguments is denoted by $AR_{fin,\mathcal{F}}$.

Example 4. Continue from Example 3. A, B, C_2 are full proof trees and their supports are assumptions which are $\{not\_\delta\}$, $\emptyset$, and $\{not\_\beta\}$ respectively. Hence they all are arguments.

If the opponent in a dialectical computation is constructing an infinite argument to attack some proponent argument (like argument C_1 in Fig. 1), the computation may not terminate and the admissibility of the proponent arguments can not be established. Declaratively, we model this situation as an attack of the infinite argument against some proponent arguments.

As infinite arguments do not provide support for their conclusions, they can not be accepted as an admissible argument. We model this intuition as self-attacks of infinite arguments.

Definition 4 (Attacks)**.**

- An argument A *attacks* an argument B iff one of the following conditions holds:
  (1) The conclusion of A is the contrary of some assumption in the support of B.
  (2) A and B are identical and infinite.
- The attack relation between arguments in $AR_{\mathcal{F}}$ is denoted by $att_{\mathcal{F}}$ while the attack relations between finite arguments is denoted by $att_{fin,\mathcal{F}}$. Define

$$AF_{\mathcal{F}} = (AR_{\mathcal{F}}, att_{\mathcal{F}}) \quad \text{and} \quad AF_{fin,\mathcal{F}} = (AR_{fin,\mathcal{F}}, att_{fin,\mathcal{F}})$$

Example 5. Consider again Example 1, we have $AF_{\mathcal{F}_1} = (AR_{\mathcal{F}_1}, att_{\mathcal{F}_1})$ where $AR_{\mathcal{F}_1} \supseteq \{A, B, C_1\}$ and $att_{\mathcal{F}_1} \supseteq \{(C_1, A), (C_1, C_1)\}$.

Further $AF_{\mathcal{F}_2} = (AR_{\mathcal{F}_2}, att_{\mathcal{F}_2})$ where $AR_{\mathcal{F}_2} \supseteq \{A, B, C_2\}$ and $att_{\mathcal{F}_2} \supseteq \{(C_2, A), (B, C_2), (C_2, C_2)\}$.

If we consider only finite arguments, $AR_{fin,\mathcal{F}_1} = AR_{fin,\mathcal{F}_2}$ and $att_{fin,\mathcal{F}_1} = att_{fin,\mathcal{F}_2}$. Hence $AF_{fin,\mathcal{F}_1} = AF_{fin,\mathcal{F}_2}$.

The graphical representation of attacks among arguments can be seen in Fig. 4.

(a) Attacks between arguments in $\mathcal{F}_1$. (b) Attacks between arguments in $\mathcal{F}_2$.

Fig. 4. A graphical representation of attacks among arguments.

[4]I.e. all leaves of any full proof tree T are final and $Ass(T) = Sp(T)$.

Due to the fact that the infinite arguments always attack themself, the following lemma holds obviously.

Lemma 1. *Let $S \subseteq \mathrm{AR}_{\mathcal{F}}$ be admissible wrt $\mathrm{AF}_{\mathcal{F}} = (\mathrm{AR}_{\mathcal{F}}, \mathrm{att}_{\mathcal{F}})$. Then S contains only finite arguments.*

Definition 5 (Admissibility and strict admissibility)**.** Abusing the notation slightly for simplicity, we say that a set of arguments $S \subseteq \mathrm{AR}_{\mathcal{F}}$ is

- *strictly admissible* iff it is admissible wrt the argumentation framework $\mathrm{AF}_{\mathcal{F}} = (\mathrm{AR}_{\mathcal{F}}, \mathrm{att}_{\mathcal{F}})$, and
- *admissible* iff it is admissible wrt the argumentation framework $\mathrm{AF}_{\mathrm{fin},\mathcal{F}} = (\mathrm{AR}_{\mathrm{fin},\mathcal{F}}, \mathrm{att}_{\mathrm{fin},\mathcal{F}})$.

It holds obviously.

Theorem 1. *If S is strictly admissible then S is also admissible.*

In Fig. 1, the set of arguments $\{A, B\}$ is strictly admissible wrt $\mathrm{AF}_{\mathcal{F}_2}$ but not wrt $\mathrm{AF}_{\mathcal{F}_1}$ as B attacks C_2 but not C_1.

We define accordingly the notions of admissible and strictly admissible sets of assumptions.

Definition 6 (Admissible and strictly admissible sets of assumptions)**.** Let H be a set of assumptions and AR_H be the set of all finite arguments whose supports are subsets of H.

(1) We say H is *admissible (resp strictly admissible)* iff there is subset $S \subseteq \mathrm{AR}_H$ such that $\mathrm{Ass}(S) = H$ and S is admissible (resp. strictly admissible).

(2) A sentence σ is said to be *credulously derived* (resp. *strictly credulously derived*) from H if H is admissible (resp. strictly admissible) and there is $A \in \mathrm{AR}_H$ such that $\mathrm{Cl}(A) = \sigma$.

(3) We write $H \mid\!\sim \sigma$ to denote that σ is strictly credulously derived from H.

Example 6. Consider again Example 5 and let $H = \{\mathrm{not\_}\delta\}$. Hence, $\mathrm{AR}_H \supset \{A, B\}$ (wrt both $\mathcal{F}_1, \mathcal{F}_2$). Since C_1 attacks A wrt $\mathcal{F}_1$ but there is no attack against C_1 wrt $\mathcal{F}_1$, H is not strictly admissible wrt $\mathcal{F}_1$. In contrast, H is strictly admissible wrt $\mathcal{F}_2$ as B attacks C_2. Hence p is strictly credulously derived from H wrt $\mathcal{F}_2$ but not wrt $\mathcal{F}_1$.

3. Introducing proof-trees-based dialectical proof procedures

Dialectical proof procedures could be viewed as games between a proponent who is trying to construct an argument for some conclusion and defend it from the attacking arguments constructed by an opponent. Both players construct their arguments by expanding partial proof trees stepwise to the full proof trees. In [15–17,35,36], the constructed proof trees are implicit, acting more or less as intuitive guidances. The procedures only present a flattened view of the proof trees represented by multisets of their supports. [22] introduces more structure by representing proof trees as a pair of support and conclusion.

We will present two procedures. In one, proof trees are fully and explicitly represented. The explicit representation of proof trees (or partial arguments) allows deeper structural insights into process of argument construction by incorporating the concept of expansion of partial arguments into the procedures and hence making the task of tracing the construction of proof trees simpler and more natural. It simplifies the associated technical machinery in no small amount. The second procedure is a result of flattening the first.

We first present some key insights into the structure of proof trees that are needed to understand the procedures.

3.1. Sequence of partial proof trees

Notation 3. Let T, T' be partial proof trees and N be a non-final leaf node in T labeled by a non-assumption σ.[5]

- T' is an *immediate expansion of T at N* if there is a rule r of the form $\sigma \leftarrow b_1, \ldots, b_m$ such that T' is obtained from T by adding m children $N.(r, b_1), \ldots, N.(r, b_m)$ to N (for $m = 0$, a child node $N.(r, \text{true})$ is added to N), i.e. $T' = T \cup \{N.(r, b_1), \ldots, N.(r, b_m)\}$.
- We write $T' = \mathbf{exp}(\mathbf{T}, \mathbf{N}, \mathbf{r})$.
- We say T' is an *immediate expansion* of T if T' is an immediate expansion of T at some leaf node N of T.
- We define

$$\mathrm{CE}(T, N) = \{\exp(T, N, r') \mid r' \text{ is a rule s.t. } \mathrm{hd}(r') = \sigma\} \quad \text{and}$$
$$\mathrm{CE}(T, N, S) = \{\exp(T, N, r') \mid r' \text{ is a rule s.t. } \mathrm{hd}(r') = \sigma, \mathrm{bd}(r) \cap S = \emptyset\}$$

  where S is a set of assumptions.

It is easy to see that $\mathrm{CE}(T, N) = \mathrm{CE}(T, N, \emptyset)$.

Example 7. Consider the partial proof tree $C_{20} = \{\pi_0, \pi_1, \pi_1'\}$ in Fig. 3.
Let $N = \pi_1$. Then

$$\exp(C_{20}, N, r_0) = C_{20} \cup \{\pi_1.(r_0, f(1))\} = \{\pi_0, \pi_1', \pi_1, \pi_2\} = C_{21}.$$

Notation 4. Let T_0, T_1 be partial proof trees for σ_0.
We say T_0 is a *prefix* of T_1 iff $T_0 \subseteq T_1$.
We say T_0 is a *proper prefix* of T_1 if T_0 is a prefix of T_1 and $T_0 \neq T_1$.

Lemma 2. *Let T_0, T_1 be partial proof trees. The following statements hold:*

(1) *If T_1 is an immediate expansion of T_0 then T_0 is a prefix of T_1.*
(2) *Suppose T_0 is a prefix of T_1. It holds that*
   (a) *the roots of T_0, T_1 coincide; and*
   (b) *if N is a node in T_0 then the parent and children of N in T_0 (if exist) are respectively also the parent and children of N in T_1.*

Proof. Obvious. □

Lemma 3. *Let T be an argument, T_0 be a partial proof tree such that T_0 is a proper prefix of T. Furthermore, let N be a leaf node in T_0 and S be a set of assumptions. The following statements hold:*

(1) *Suppose* $\mathrm{CE}(T_0, N) \neq \emptyset$. *Then there is $T_1 \in \mathrm{CE}(T_0, N)$ such that T_1 is a prefix of T.*
(2) *Suppose* $\mathrm{Ass}(T) \cap S = \emptyset$ *and* $\mathrm{CE}(T_0, N, S) \neq \emptyset$. *Then there is $T_1 \in \mathrm{CE}(T_0, N, S)$ such that T_1 is a prefix of T*

[5] See Notations 2 and 1.

Proof. The first statement follows directly from the second one. We prove the second one.

Since $\text{CE}(T_0, N, S) \neq \emptyset$, N is labeled by a non-assumption sentence δ different to true. Since T is an argument, N is not a leaf node in T. Thus N has a child of the form $N.(r, \gamma)$ in T where $\text{hd}(r) = \delta$ and $\gamma \in \text{bd}(r)$. From the definition of proof trees (Definition 2), it follows $T_1 = \exp(T_0, N, r) \subseteq T$. Since $\text{Ass}(T) \cap S = \emptyset$, it follows that $\text{bd}(r) \cap S = \emptyset$. Hence $T_1 \in \text{CE}(T_0, N, S)$. The second statement holds. □

An increasing sequence of partial proof trees $T_0 \subseteq T_1 \subseteq \dots T_i \subseteq \dots$ is said to be *fair* if for each T_i, for each non-final leaf node $N \in T_i$ there is a node $M \in T_j$, $j > i$, such that N is a proper prefix of M.

Lemma 4. *Let* $\text{sq} \equiv T_0 \subseteq T_1 \subseteq \dots T_i \subseteq \dots$ *be an increasing sequence of partial proof trees. The following statements hold:*

(1) $T_0 \cup T_1 \cup \dots T_i \cup \dots$ *is a partial proof tree.*
(2) *If the sequence* sq *is fair then* $T_0 \cup T_1 \cup \dots T_i \cup \dots$ *is an argument.*

Proof. The first statement is obvious. We prove the second. Suppose $T \equiv T_0 \cup T_1 \cup \dots T_i \cup \dots$ is not an argument. Hence T has a non-final leaf node N labeled by δ. Thus $N \in T_i$ for some i. Since sq is fair, N is a proper prefix of some node $M \in T_j$, $j > i$. Hence N is not a leaf of T. Contradiction. □

Notation 5. Let T be a partial proof tree and $N \equiv (\text{root}, \sigma_0).(r_1, \sigma_1).\dots.(r_i, \sigma_i)$ be a node in T.

The *height* of N in T, denoted by $h(N, T)$, is defined by $h(N, T) = i$.[6]

The *maximum of the heights* of the non-final leaf nodes in T is denoted by $\text{ha}(T)$, i.e.

$$\text{ha}(T) = \max\{h(N, T) \mid N \text{ is a non-final leaf node in } T\}$$

Notation 6. Two partial proof trees T, T' are *compatible* iff $T \cup T'$ is also a partial proof tree.

Lemma 5. *Let* Π *be a infinite set of partial proof trees wrt a finitary ABA* $\mathcal{F}$ *such that*

- *for all* $n \geqslant 0$, *for each* $T \in \Pi$ *such that* $\text{ha}(T) > n$, *there is* $T' \in \Pi$ *such that* $T' \subseteq T$ *and* $\text{ha}(T') = n$; *and*
- *for each* $n \geqslant 0$, *the set* $\{T \in \Pi \mid \text{ha}(T) \leqslant n\}$ *is finite; and*
- *for all* $T, T' \in \Pi$ *such that* T, T' *are compatible, it holds that* $T \subseteq T'$ *or* $T' \subseteq T$.

Then there is an infinite strictly increasing sequence of proof trees

$$T_0 \subset T_1 \subset \dots \subset T_n \subset \dots$$

such that for each $i \geqslant 0$, $T_i \in \Pi$.

Proof. See Appendix A.2. □

[6]Hence the height of the root is 0.

3.2. Dialectical proof procedure

We give below a dialectical proof procedure for constructing an admissible set of arguments supporting some sentence σ. The procedure could be viewed as a stage-wise construction of the dispute derivation of the form $\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PT}_i, \mathrm{PA}_i, \mathrm{OT}_i, \mathrm{OA}_i\rangle$ where at each stage i,

- PT_i is the set of partial proof trees the proponent has constructed until stage i;
- OT_i is the set of partial proof trees the opponent has constructed and not yet attacked by the proponent until stage i;
- PA_i (resp OA_i) is the set of assumptions that 1) appear in the partial proof trees constructed by the proponent (resp opponent) until stage i and 2) have been attacked by the other party until stage i.

Definition 7 (Proof tree-based dispute derivation)**.** A proof-tree-based dispute derivation for a sentence σ is a (possibly infinite) sequence of the form

$$\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PT}_i, \mathrm{PA}_i, \mathrm{OT}_i, \mathrm{OA}_i\rangle$$

where

- for each i, PA_i, OA_i are sets of assumptions and PT_i, OT_i are sets of partial proof trees, and
- PT_0 contains exactly one partial proof tree consisting of only the root labeled by σ (i.e. $\mathrm{PT}_0 = \{\{(\mathrm{root}, \sigma)\}\}$), and
- $\mathrm{PA}_0 = \mathrm{OT}_0 = \mathrm{OA}_0 = \emptyset$, and
- at stage i, one of the dispute parties makes a move satisfying the following properties:
  (1) Suppose the proponent makes a move at stage i. The proponent has two options:
    (a) The proponent expands one of her partial arguments by selecting a partial proof tree $T \in \mathrm{PT}_i$, a non-final leaf node N in T labeled by δ, a rule r with head δ such that $\mathrm{bd}(r) \cap \mathrm{OA}_i = \emptyset$ and expanding T resulting in:

$$\mathrm{PT}_{i+1} = \big(\mathrm{PT}_i \setminus \{T\}\big) \cup \big\{\exp(T, N, r)\big\}.$$
$$\mathrm{PA}_{i+1} = \mathrm{PA}_i$$
$$\mathrm{OT}_{i+1} = \mathrm{OT}_i$$
$$\mathrm{OA}_{i+1} = \mathrm{OA}_i$$

    (b) The proponent attacks partial proof trees in OT_i at an assumption $\alpha \in \mathrm{Ass}(\mathrm{OT}_i) \setminus \mathrm{Ass}(\mathrm{PT}_i)$.[7] Then

$$\mathrm{PT}_{i+1} = \mathrm{PT}_i \cup \big\{\big\{(\mathrm{root}, \overline{\alpha})\big\}\big\}$$
$$\mathrm{PA}_{i+1} = \mathrm{PA}_i$$
$$\mathrm{OT}_{i+1} = \mathrm{OT}_i \setminus \big\{T' \in \mathrm{OT}_i \mid \alpha \in \mathrm{Ass}\big(T'\big)\big\}$$
$$\mathrm{OA}_{i+1} = \mathrm{OA}_i \cup \{\alpha\}.$$

[7] See Notation 2.

Table 1

A successful tree-based dispute derivation

Stage	PT	PA	OT	OA	Step
0	A_0	∅	∅	∅	1a
1	A	∅	∅	∅	2a
2	A	not_δ	C_0	∅	2b
3	A	not_δ	C_{20}	∅	2b
4	A	not_δ	C_{21}	∅	1b
5	A, B_0	not_δ	∅	not_β	1a
6	A, B	not_δ	∅	not_β	success

(2) Suppose the opponent makes a move at stage i. The opponent has two options:

(a) The opponent attacks a proponent partial proof tree $T \in \mathrm{PT}_i$ at a leaf node labeled by an assumption $\alpha \in \mathrm{Ass}(T) \setminus \mathrm{PA}_i$:

$$\mathrm{PT}_{i+1} = \mathrm{PT}_i$$
$$\mathrm{PA}_{i+1} = \mathrm{PA}_i \cup \{\alpha\}$$
$$\mathrm{OT}_{i+1} = \mathrm{OT}_i \cup \bigl\{\bigl\{(\mathrm{root}, \overline{\alpha})\bigr\}\bigr\}$$
$$\mathrm{OA}_{i+1} = \mathrm{OA}_i.$$

(b) The opponent expands an opponent partial proof tree $T \in \mathrm{OT}_i$ at a non-final leaf node N filtered by the assumptions in OA_i, i.e.,

$$\mathrm{PT}_{i+1} = \mathrm{PT}_i$$
$$\mathrm{PA}_{i+1} = \mathrm{PA}_i$$
$$\mathrm{OT}_{i+1} = \bigl(\mathrm{OT}_i \setminus \{T\}\bigr) \cup \mathrm{CE}(T, N, \mathrm{OA}_i)$$
$$\mathrm{OA}_{i+1} = \mathrm{OA}_i.$$

Definition 8. A proof-tree-based dispute derivation $\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$ is successful if $\mathrm{OT}_n = \emptyset$, and PT_n consists only of full proof trees and $\mathrm{PA}_n = \mathrm{Ass}(\mathrm{PT}_n)$.

Remark 3. For simplicity, in the next two Sections 4, 5, we refer to proof tree-based dispute derivation just as dispute derivation if there is no possibility for misunderstanding.

Example 8. Consider again the argumentation framework $\mathcal{F}_2$ in Example 1 with the partial arguments in Fig. 3.

A proof-tree based-dispute derivation for p is given in Table 1.

4. Soundness of dispute derivation

Before giving the soundness theorem, we present a number of lemmas to illuminate the structure of dispute derivations.

Lemma 6. *Let* dd $= \langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \dots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$ *be a dispute derivation. The following statements hold:*

(1) $\mathrm{Ass}(\mathrm{OT}_n) \cap \mathrm{OA}_n = \emptyset$.
(2) *For each* $\alpha \in \mathrm{OA}_n$*, there is a partial proof tree* $T \in \mathrm{PT}_n$ *such that* $\mathrm{Cl}(T) = \overline{\alpha}$.
(3) *For each* $\beta \in \mathrm{PA}_n$ *there is a unique stage* i*,* $i < n$*, such that* $\{\beta\} = \mathrm{PA}_{i+1} \setminus \mathrm{PA}_i$ *and* $\{(\mathrm{root}, \overline{\beta})\} \in \mathrm{OT}_{i+1} \setminus \mathrm{OT}_i$.
For each a partial proof tree $T \in \mathrm{OT}_i$*,* $i \leqslant n$*, there is a unique* $\beta \in \mathrm{PA}_n$ *such that* $\mathrm{Cl}(T) = \overline{\beta}$.
(4) *Let* $T, T' \in \mathrm{OT}_i$*,* $i \geqslant 0$*. If* T, T' *are compatible then* T, T' *are identical.*

Proof. See Appendix A.3. □

We next introduce a new relevant notion of scope of a proof tree in dispute derivation to describe the expansion process of opponent proof trees in dispute derivations.

Definition 9 (Scope)**.** Let T be a proof tree and dd $= \langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \dots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$ be a dispute derivation for σ.

A *scope of length* k of T in dd is a pair (sq, i) where $\mathrm{sq} \equiv T_0, T_1, \dots, T_{k-1}$, $i \geqslant 0$ and $k \geqslant 1$, such that the following conditions hold:

(1) $T_0 = \{(\mathrm{root}, \mathrm{Cl}(T))\}$ and $T_0 \in \mathrm{OT}_i \setminus \mathrm{OT}_{i-1}$.
(2) For each $0 \leqslant j \leqslant k-1$: $T_j \in \mathrm{OT}_{i+j}$ and $T_j \subseteq T$;
(3) For each $0 \leqslant j < k-1$: either $T_{j+1} = T_j$ or T_{j+1} is an immediate expansion of T_j.

A scope (seq, i) of T in dd is a *full scope* if there is no prefix of T in OT_{i+k}.

Lemma 7. *Let* T *be a proof tree and* dd $= \langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \dots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$ *be a dispute derivation. The following statements hold:*

(1) *Let* (sq, i)*,* (sq', j) *be two scopes of* T *in* dd *of equal length. Then* $i = j$ *and* sq, sq′ *are identical;*
(2) *If* $T \in \mathrm{OT}_n$ *then there is a scope* (sq, i)*,* $\mathrm{sq} = T_0, \dots, T_{k-1}$*, of* T *in* dd *such that* $i + k - 1 = n$ *and* $T = T_{k-1}$*;*
(3) *If* dd *is a successful dispute derivation that terminates at* $\langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$ *and* T *is an argument attacking some argument in* PT_n *then there is a unique full scope of* T *in* dd.

Proof. See Appendix A.4. □

Lemma 8. *Let* dd $= \langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \dots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$ *be a successful dispute derivation and let* T *be an argument attacking some argument in* PT_n*. Then* $\mathrm{Ass}(T) \cap \mathrm{OA}_n \neq \emptyset$.

Proof. Let (sq, i) with $\mathrm{sq} = T_0, \dots, T_{k-1}$, $1 \leqslant k$, be the unique full scope of T in dd (the existence of (sq, i) follows from Lemma 7). As (sq, i) is a full scope of T, there is no prefix of T in OT_{i+k}. Hence there are two cases:

- T_{k-1} is attacked by the proponent at stage $i + k - 1$ in dd using (1.b). It follows that T_{k-1} (and hence T) contains an assumptions in $\mathrm{OA}_{i+k-1} \subseteq \mathrm{OA}_n$.

- T_{k-1} is filtered out by the opponent at stage $i+k-1$ in dd using (2.b). It follows that a non-final leaf node N of T_{k-1} is selected such that

$$\mathrm{OT}_{i+k} = \big(\mathrm{OT}_{i+k-1} \setminus \{T_{k-1}\}\big) \cup \mathrm{CE}(T_{k-1}, N, \mathrm{OA}_{i+k-1})$$

  Suppose $\mathrm{Ass}(T) \cap \mathrm{OA}_n = \emptyset$. Hence $\mathrm{Ass}(T) \cap \mathrm{OA}_{i+k-1} = \emptyset$. From Lemma 3, it follows that there is $T' \in \mathrm{CE}(T_{k-1}, N, \mathrm{OA}_{i+k-1})$ such that T' is a prefix of T. Contradiction since there is no proof tree in OT_{i+k} that is a prefix of T.

Therefore $\mathrm{Ass}(T) \cap \mathrm{OA}_n \neq \emptyset$. □

Lemma 9. *Let* $\mathrm{dd} = \langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$ *be a successful dispute derivation for* σ. *Then* $\mathrm{PA}_n \cap \mathrm{OA}_n = \emptyset$.

Proof. Suppose there exists $\alpha \in \mathrm{PA}_n \cap \mathrm{OA}_n$. Let i be the stage in dd where α is attacked by the proponent (step (1.b) in Definition 7) and inserted into OA_i. Therefore α does not appear in PT_i. The only ways for α to show up in PT_n later is by applying step (1.a).

It is also clear that for each $i < j \leqslant n$, $\alpha \in \mathrm{OA}_j$. At any stage $j > i$ in dd, if the proponent expands a partial proof tree $T \in \mathrm{PT}_j$ then only rules r with $\mathrm{bd}(r) \cap \mathrm{OA}_j = \emptyset$ are selected (the filtering condition in step (1.a) of Definition 7). Therefore α can never show up in PT_j. Hence $\alpha \notin \mathrm{PA}_n$. Contradiction. □

Theorem 2 (Soundness theorem)**.** *Let* $\mathrm{dd} = \langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$ *be a successful dispute derivation for* σ. *The following statements hold:*

(1) PT_n *is strictly admissible and* $\sigma \in \mathrm{Cl}(\mathrm{PT}_n)$.
(2) $\mathrm{PA}_n \mid\!\sim \sigma$.

Proof. The second statement follows directly from the first. We proceed to prove the first statement below.

It is obvious that σ labels the root of some full proof tree in PT_n.

Suppose an argument T attacks PT_n. From the Lemmas 8, 6, it follows immediately that T is attacked by PT_n.

We show now that PT_n is conflict-free. Suppose the contrary that PT_n is not conflict-free. Then there are arguments $T, T' \in \mathrm{PT}_n$ such that T attacks T'. From Lemma 8, T contains an assumption from OA_n. From $\mathrm{Ass}(T) \subseteq \mathrm{Ass}(\mathrm{PT}_n) = \mathrm{PA}_n$, it follows $\mathrm{PA}_n \cap \mathrm{OA}_n \neq \emptyset$. Contradiction to Lemma 9. □

5. Completeness of proof-tree-based dispute derivation

Theorem 3 (Completeness theorem)**.** *Let* $\mathcal{F}$ *be a finitary ABA framework,* H *be a strictly admissible finite set of assumptions and* σ *be a sentence such that* $H \mid\!\sim \sigma$. *Then there is a successful proof-tree-based dispute derivation* $\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$ *for* σ *such that* $\mathrm{PA}_n \subseteq H$.

To prove the theorem, we construct a successful dispute derivation for σ by imposing certain extra conditions on the steps of proponent and opponent in the dispute derivation procedure (Definition 7). The formal proof is given in Section 5.2 below.

5.1. H-Constrained dispute derivation

Given $H|\!\sim \sigma$, the task is to construct a successful dispute derivation

$$\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$$

for σ such that $\mathrm{PA}_n \subseteq H$ and $\mathrm{PT}_n \subseteq \mathrm{AR}_H$.[8]

The key idea is rather simple: Associate each partial proof tree $T \in \mathrm{PT}_i$ to some argument $A \in \mathrm{AR}_H$ such that 1) T is a prefix of A and 2) at step (1.a) when T is selected, expand T towards A so that at the end of the derivation, A is fully constructed.

We illustrate this idea in Example 9 below.

Example 9.

$$r_1 : \sigma \leftarrow \text{not\_}\delta \qquad r_2 : \delta \leftarrow \text{not\_}\beta$$

$$r_3 : \beta \leftarrow \text{not\_}\gamma \qquad r_4 : \beta \leftarrow d$$

Let $H = \{\text{not\_}\delta, \text{not\_}\gamma\}$. Hence $\mathrm{AR}_H \supseteq \{T_1, T_3\}$ (see Fig. 5).

Consider the dispute derivations for σ represented in Tables 2, 3. At stage 4, either r_3 or r_4 could be selected for an expansion of T_2. Selecting r_4 leads to a failed derivation (Table 2). If we associate T_2 with T_3 at stage 4 when it is created, we would select r_3 to expand T_2 towards T_3 leading to a successful dispute derivation (Table 3).

More formally, at each stage $\langle \mathrm{PT}_i, \mathrm{PA}_i, \mathrm{OT}_i, \mathrm{OA}_i\rangle$ in a dispute derivation, we add a new component λ_i representing a function that associates each proof tree $T \in \mathrm{PT}_i$ to an argument $\lambda_i(T) \in \mathrm{AR}_H$ such

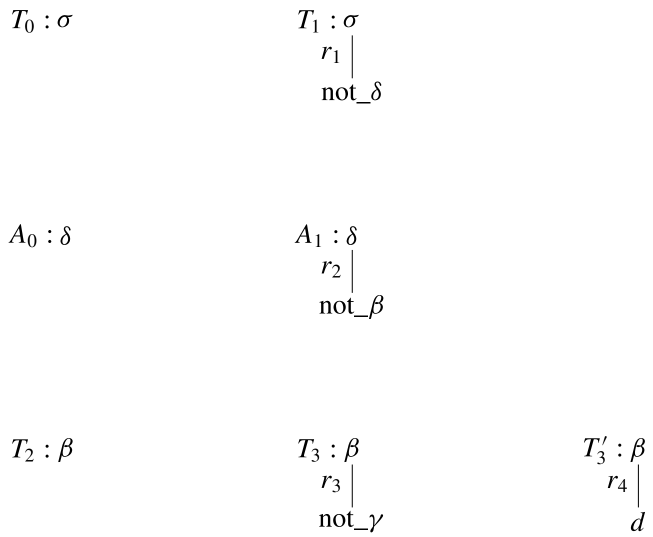

Fig. 5. Partial proof trees of Example 9.

[8] AR_H is the set of all finite arguments whose assumptions belong to H.

Table 2

A failed tree-based dispute derivation

Stage	PT	PA	OT	OA	Step
0	T_0	∅	∅	∅	1a
1	T_1	∅	∅	∅	2a
2	T_1	not_δ	A_0	∅	2b
3	T_1	not_δ	A_1	∅	1b
4	T_1, T_2	not_δ	∅	not_β	1a
5	T_1, T_3'	not_δ	∅	not_β	fail

Table 3

A successful tree-based dispute derivation

Stage	PT	PA	OT	OA	Step
0	T_0	∅	∅	∅	1a
1	T_1	∅	∅	∅	2a
2	T_1	not_δ	A_0	∅	2b
3	T_1	not_δ	A_1	∅	1b
4	T_1, T_2	not_δ	∅	not_β	1a
5	T_1, T_3	not_δ	∅	not_β	success

that T is a prefix of $\lambda_i(T)$. $\lambda(T)$ will guide the expansions of T so that at the end of the derivation, $\lambda(T)$ is fully constructed. Hence if procedure step (1.a) is applied at stage i, and T is selected and expanded into T' then T' is also a prefix of $\lambda_i(T)$ and $\lambda_{i+1}(T') = \lambda_i(T)$.

Convention 2. For ease of representation, we often represent a function f from X to Y as a binary relation $f = \{(x, y) \mid x \in X, y \in Y : y = f(x)\}$.

We introduce two new notions.

Notation 7. The *minimum of the heights* of the non-final leaf nodes in a proof tree T is denoted by

$$\mathrm{hi}(T) = \min\{h(N, T) \mid N \text{ is a non-final leaf node in } T\}$$ [9]

For a set S of proof trees, define

$$\mathrm{hi}(S) = \min\{\mathrm{hi}(T) \mid T \in S\}$$

A partial proof tree T is said to be *almost balanced* iff $\mathrm{ha}(T) \leqslant \mathrm{hi}(T) + 1$.

Definition 10. Let H be a set of assumptions. A *H-constrained dispute derivation* for σ is a (possibly infinite) sequence

$$\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0, \lambda_0\rangle, \ldots, \langle \mathrm{PT}_i, \mathrm{PA}_i, \mathrm{OT}_i, \mathrm{OA}_i, \lambda_i\rangle$$

where

[9] See also Notation 5.

- for each i, PA_i, OA_i are sets of assumptions, PT_i, OT_i are sets of partial proof trees, and
- $\lambda_i : \mathrm{PT}_i \to \mathrm{AR}_H$; and
- PT_0, PA_0, OT_0, OA_0 are defined as in Definition 7; and
- $\{(\mathrm{root}, \sigma)\} \subseteq \lambda_0(\{(\mathrm{root}, \sigma)\})$;
- at stage i, one of the dispute parties makes a move satisfying the following properties:
  (1) Suppose the proponent makes a move at stage i. The proponent has two options:
    (a) The proponent proceeds as in step (1.a) in Definition 7 with an extra condition that $\exp(T, N, r) \subseteq \lambda_i(T)$, and computes λ_{i+1} as follows:
    $$\lambda_{i+1} = \big(\lambda_i \setminus \big\{\big(T, \lambda_i(T)\big)\big\}\big) \cup \big\{\big(\exp(T, N, r), \lambda_i(T)\big)\big\}$$
    (b) The proponent selects an assumption $\alpha \in \mathrm{Ass}(\mathrm{OT}_i) \setminus \mathrm{Ass}(\mathrm{PT}_i)$ and an argument $A \in \mathrm{AR}_H$ such that $\mathrm{Cl}(A) = \overline{\alpha}$ and then proceeds to compute PT_{i+1}, PA_{i+1}, OT_{i+1}, OA_{i+1} as in step (1.b) in Definition 7, and computes λ_{i+1} as follows:
    $$\lambda_{i+1} = \lambda_i \cup \big\{(T_0, A)\big\} \quad \text{with } T_0 = \big\{(\mathrm{root}, \overline{\alpha})\big\}.$$
  (2) Suppose the opponent makes a move at stage i. The opponent has two options:
    (a) The opponent proceeds as in step (2.a) in Definition 7 where λ_{i+1} is computed by $\lambda_{i+1} = \lambda_i$.
    (b) The opponent proceeds as in step (2.b) in Definition 7 with two extra conditions:
      * $h(N, T) = \mathrm{hi}(\mathrm{OT}_i)$; and
      * it is not possible for the proponent to execute step 1.b.[10]

    PT_{i+1}, PA_{i+1}, OT_{i+1}, OA_{i+1} are computed as in step (2.b) in Definition 7, and $\lambda_{i+1} = \lambda_i$

Definition 11. A H-constrained dispute derivation

$\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0, \lambda_0\rangle, \dots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n, \lambda_n\rangle$ is *successful* iff the dispute derivation

$\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \dots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$ is successful.

Example 10. Consider again Example 9 (the partial proof trees of this example can be seen in Fig. 5) where $H = \{\mathrm{not}\_\delta, \mathrm{not}\_\gamma\}$ and $\mathrm{AR}_H \supseteq \{T_1, T_3\}$.

A H-constrained dispute derivation for σ is given in Table 4. Note that at stage 4, guided by $\lambda_4(T_2) = T_3$, rule r_3 must be selected to expand T_2 into T_3. In other words, rule r_4 can not be selected and hence no failed derivation can be constructed.

Lemma 10. *Let* $\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0, \lambda_0\rangle, \dots, \langle \mathrm{PT}_i, \mathrm{PA}_i, \mathrm{OT}_i, \mathrm{OA}_i, \lambda_i\rangle$ *be a* H*-constrained dispute derivation for* σ*. The following conditions hold:*

(1) $\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \dots, \langle \mathrm{PT}_i, \mathrm{PA}_i, \mathrm{OT}_i, \mathrm{OA}_i\rangle$ *is a dispute derivation for* σ *(as defined in Definition 7).*
(2) $\lambda_i : \mathrm{PT}_i \to \mathrm{AR}_H$, $i \geqslant 0$, *such that for each* $T \in \mathrm{PT}_i$*:* $T \subseteq \lambda_i(T)$*;*

[10] I.e. step (2.b) is selected by the opponent only if it is not possible for the proponent to execute step (1.b) meaning that an opponent proof tree is expanded only if no assumptions appearing in it could be attacked by the proponent.

Table 4

A successful tree-based dispute derivation

Stage	PT	PA	OT	OA	λ	Step
0	T_0	∅	∅	∅	(T_0, T_1)	$1a$
1	T_1	∅	∅	∅	(T_1, T_1)	$2a$
2	T_1	not_δ	A_0	∅	(T_1, T_1)	$2b$
3	T_1	not_δ	A_1	∅	(T_1, T_1)	$1b$
4	T_1, T_2	not_δ	∅	not_β	$(T_1, T_1), (T_2, T_3)$	$1a$
5	T_1, T_3	not_δ	∅	not_β	$(T_1, T_1), (T_3, T_3)$	success

Proof. Statement 1 follows directly from Definition 7 and 10.

We prove the second statement by induction on n.

Basic step: $n = 0$. The statement follows directly from the definition of $\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0, \lambda_0\rangle$ in Definition 10.

Inductive step. We show the statement holds for $n + 1$ assuming it holds for n.

If the step applied at stage n is (2.a) or (2.b), then the statement holds obviously since $\mathrm{PT}_n = \mathrm{PT}_{n+1}$, $\lambda_n = \lambda_{n+1}$.

Suppose step (1.a) is applied at stage n. Let $T \in \mathrm{PT}_n$ be the selected proof tree. Therefore for any $T' \in \mathrm{PT}_n$ and $T' \neq T$, $\lambda_{n+1}(T') = \lambda_n(T') \supseteq T'$.

Let $\exp(T, N, r)$ be the expansion of T at stage n. Hence

$$\lambda_{n+1}\big(\exp(T, N, r)\big) = \lambda_n(T) \supseteq \exp(T, N, r).$$

The statement holds.

Suppose step (1.b) is applied at stage n. The statement follows immediately from the fact that

$$\mathrm{PT}_{n+1} = \mathrm{PT}_n \cup \big\{\big\{(\mathrm{root}, \overline{\alpha})\big\}\big\} \quad \text{and}$$

$$\lambda_{i+1} = \lambda_i \cup \big\{(T_0, A)\big\} \quad \text{with } T_0 = \big\{(\mathrm{root}, \overline{\alpha})\big\} \text{ and } A \in \mathrm{AR}_H \text{ such that } \mathrm{Cl}(A) = \overline{\alpha}. \qquad \square$$

Lemma 11. *Let H be a strictly admissible set of assumptions and*

$$\mathrm{cdd} \equiv \langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0, \lambda_0\rangle, \ldots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n, \lambda_n\rangle$$

be a terminated H-constrained dispute derivation for σ (i.e. neither of the players could make a move at stage n).

Then cdd *is successful.*

Proof. Since step (1.a) can not be executed at stage n, each proof tree in PT_n is full. Therefore $\mathrm{PT}_n \subseteq \mathrm{AR}_H$.

Since the opponent can not make a move with step (2.a) at stage n, it follows that

$$\mathrm{PA}_n = \mathrm{Ass}(\mathrm{PT}_n) \subseteq H.$$

Since the proponent can not make a move with step (1.b) at n, it follows that

$$\mathrm{Ass}(\mathrm{OT}_n) \subseteq \mathrm{Ass}(\mathrm{PT}_n) = \mathrm{PA}_n \subseteq H.$$

Since the opponent can not make a move with step(2.b) at stage n, it follows that all partial proof trees in OT_n are full. Hence each argument in OT_n attacks H.

From $\mathrm{Ass}(\mathrm{OT}_n) \subseteq \mathrm{Ass}(\mathrm{PT}_n) = \mathrm{PA}_n \subseteq H$, it follows $\mathrm{OT}_n \subseteq \mathrm{AR}_H$. Since H is strictly admissible, OT_n is empty.

From Definition 8, 11, cdd is successful. □

Lemma 12. *Let* cdd $= \langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PT}_i, \mathrm{PA}_i, \mathrm{OT}_i, \mathrm{OA}_i\rangle$ *be a* H*-constrained dispute derivation for* σ. *For all* $i \geqslant 0$, OT_i *is finite and every proof tree in* OT_i *is almost balanced.*

Proof. See Appendix A.5. □

Lemma 13. *Let* H *be a strictly admissible finite set of assumptions. Then there is no infinite* H*-constrained dispute derivation for* σ *wrt* $\mathcal{F}$.[11]

Proof. See Appendix A.6. □

5.2. *Proof of completeness theorem (Theorem 3)*

Construct a H-constrained dispute derivation for σ. The construction will terminate since no infinite H-constrained dispute derivation exists (Lemma 13). From Lemma 11, it follows that any terminated constrained dispute derivation is successful.

6. Flatten dispute derivation

In many proof procedures for assumption-based argumentation [15–17,22], only the supports of proof trees are of interest, not the arguments themselves. In such cases, there is often no need to carry along entire proponent or opponent trees.

A closer observation of the definition of proof tree-based dispute derivation reveals that to compute $\langle \mathrm{PT}_{i+1}, \mathrm{PA}_{i+1}, \mathrm{OT}_{i+1}, \mathrm{OA}_{i+1}\rangle$ from $\langle \mathrm{PT}_i, \mathrm{PA}_i, \mathrm{OT}_i, \mathrm{OA}_i\rangle$, we need only the sentences in the supports of arguments in $\mathrm{PT}_i \cup \mathrm{OT}_i$. Hence a "simplification" of the proof tree based dispute derivation could be obtained by replacing each tree in $\mathrm{PT}_i \cup \mathrm{OT}_i$ by its support.

The definition of flatten dispute derivations uses the notion of multisets. A very short introduction to multisets is given in Appendix A.1.

Definition 12 (Flatten dispute derivation)**.** A flatten dispute derivation for a sentence σ is a sequence of the form

$$\langle \mathrm{PS}_0, \mathrm{PA}_0, \mathrm{OS}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PS}_n, \mathrm{PA}_n, \mathrm{OS}_n, \mathrm{OA}_n\rangle$$

where

- for each i, PA_i, OA_i are sets of assumptions and PS_i is a multiset of sentences and OS_i is a multiset of multisets of sentences, and
- $\mathrm{PS}_0 = \{\sigma\}$, and $\mathrm{PA}_0 = \mathrm{OS}_0 = \mathrm{OA}_0 = \emptyset$, and

[11] Note that $\mathcal{F}$ is finitary.

- at step i, one of the dispute parties makes a move satisfying the following properties:
  (1) Suppose the proponent makes a move at step i. The proponent has two options:
    (a) The proponent selects a non-assumption $\delta \in \mathrm{PS}_i$, a rule r with head δ such that $\mathrm{bd}(r) \cap \mathrm{OA}_i = \emptyset$ and proceeds as follows:

$$\mathrm{PS}_{i+1} = \big(\mathrm{PS}_i \setminus \{\delta\}\big) \oplus \big(\mathrm{bd}(r) \setminus \mathrm{PA}_i\big).$$
$$\mathrm{PA}_{i+1} = \mathrm{PA}_i$$
$$\mathrm{OS}_{i+1} = \mathrm{OS}_i$$
$$\mathrm{OA}_{i+1} = \mathrm{OA}_i$$

    (b) The proponent selects an assumption α appearing in OS_i but not in $\mathrm{PS}_i \cup \mathrm{PA}_i$ and proceeds as follows:

$$\mathrm{PS}_{i+1} = \mathrm{PS}_i \oplus \{\overline{\alpha}\}$$
$$\mathrm{PA}_{i+1} = \mathrm{PA}_i$$
$$\mathrm{OS}_{i+1} = \mathrm{OS}_i \setminus \{S \in \mathrm{OS}_i \mid \alpha \in S\}$$ [12]
$$\mathrm{OA}_{i+1} = \mathrm{OA}_i \cup \{\alpha\}.$$

  (2) Suppose the opponent makes a move at step i. The opponent has two options:
    (a) The opponent selects an assumption $\alpha \in \mathrm{PS}_i$ and proceeds as follows:

$$\mathrm{PS}_{i+1} = \mathrm{PS}_i - \{\alpha\}$$
$$\mathrm{PA}_{i+1} = \mathrm{PA}_i \cup \{\alpha\}$$
$$\mathrm{OS}_{i+1} = \mathrm{OS}_i \oplus \big\{\{\overline{\alpha}\}\big\}$$
$$\mathrm{OA}_{i+1} = \mathrm{OA}_i.$$

    (b) The opponent selects $S \in \mathrm{OS}_i$ and a non-assumption $\delta \in S$ and proceeds as follows:

$$\mathrm{PS}_{i+1} = \mathrm{PS}_i$$
$$\mathrm{PA}_{i+1} = \mathrm{PA}_i$$
$$\mathrm{OS}_{i+1} = \big(\mathrm{OS}_i \setminus \{S\}\big) \oplus \big\{\big(S \setminus \{\delta\}\big) \oplus \mathrm{bd}(r) \mid \mathrm{hd}(r) = \delta, \mathrm{bd}(r) \cap \mathrm{OA}_i = \emptyset\big\}$$
$$\mathrm{OA}_{i+1} = \mathrm{OA}_i.$$

Definition 13. A dispute derivation $\langle \mathrm{PS}_0, \mathrm{PA}_0, \mathrm{OS}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PS}_n, \mathrm{PA}_n, \mathrm{OS}_n, \mathrm{OA}_n\rangle$ is successful if $\mathrm{PS}_n = \mathrm{OS}_n = \emptyset$.

Example 11. Consider again the argumentation framework $\mathcal{F}_2$ in Example 1.
An flatten dispute derivation for p is given in Table 5.

[12]For clarity, consider a multiset $X = \{2, 2, 5\}$. Then $Y = \{x \in X \mid x \text{ is even}\} = \{2, 2\}$.

Table 5

A successful flatten dispute derivation for p wrt $\mathcal{F}_2$ in Example 1

Stage	PS	PA	OS	OA	Step
0	p	$\emptyset$	$\emptyset$	$\emptyset$	1a
1	not_δ	$\emptyset$	$\emptyset$	$\emptyset$	2a
2	$\emptyset$	not_δ	$\{\delta\}$	$\emptyset$	2b
3	$\emptyset$	not_δ	$\{\text{not\_}\beta, f(0)\}$	$\emptyset$	2b
4	$\emptyset$	not_δ	$\{\text{not\_}\beta, f(1)\}$	$\emptyset$	1b
5	β	not_δ	$\emptyset$	not_β	1a
6	$\emptyset$	not_δ	$\emptyset$	not_β	success

Lemma 14. *Let* $\langle \text{PS}_0, \text{PA}_0, \text{OS}_0, \text{OA}_0\rangle, \ldots, \langle \text{PS}_n, \text{PA}_n, \text{OS}_n, \text{OA}_n\rangle$ *be a (possibly infinite) flatten dispute derivation. Then for any* $n \geqslant 0$*, the following properties hold:*

(1) $\text{PS}_n \cap \text{PA}_n = \emptyset$;
(2) $\text{OA}_n \cap \bigoplus \text{OS}_n = \emptyset$.

Proof. See Appendix A.7. $\square$

Notation 8. For any proof tree T, $\text{Spm}(T)$ denotes the multiset of the sentences labeling the leaf nodes of T and different to true.[13] $\text{Spm}(T)$ is referred to as the *multiset support* of T.

For a set of proof trees S, $\text{Spm}(S)$ is the union of the multiset supports of trees belonging to S.[14]

The following lemmas show that each proof-tree-based dispute derivation could be translated into an equivalent flatten one and for each flatten dispute derivation there is an equivalent proof-tree-based one.

Lemma 15. *Let* $\text{dd} = \langle \text{PT}_0, \text{PA}_0, \text{OT}_0, \text{OA}_0\rangle, \ldots, \langle \text{PT}_n, \text{PA}_n, \text{OT}_n, \text{OA}_n\rangle$ *be a proof-tree-based dispute derivation for* σ.

Then the sequence $\langle \text{PS}_0, \text{PA}_0, \text{OS}_0, \text{OA}_0\rangle, \ldots, \langle \text{PS}_n, \text{PA}_n, \text{OS}_n, \text{OA}_n\rangle$ *where for each* $0 \leqslant i \leqslant n$, $\text{PS}_i = \text{Spm}(\text{PT}_i) - \text{PA}_i$ *and* $\text{OS}_i = \{\text{Spm}(T) \mid T \in \text{OT}_i\}$, *is a flatten dispute derivation for* σ.

Proof. See Appendix A.8. $\square$

Lemma 16. *Let* $\langle \text{PS}_0, \text{PA}_0, \text{OS}_0, \text{OA}_0\rangle, \ldots, \langle \text{PS}_n, \text{PA}_n, \text{OS}_n, \text{OA}_n\rangle$ *be a flatten dispute derivation for* σ.

There is a proof-tree-based dispute derivation

$$\langle \text{PT}_0, \text{PA}_0, \text{OT}_0, \text{OA}_0\rangle, \ldots, \langle \text{PT}_n, \text{PA}_n, \text{OT}_n, \text{OA}_n\rangle$$

for σ *such that for all* $0 \leqslant i \leqslant n$*, the following properties hold:*

(1) $\text{PA}_i \subseteq \text{Ass}(\text{PT}_i)$ *and* $\text{PS}_i = \text{Spm}(\text{PT}_i) - \text{PA}_i$;
(2) $\text{OS}_i = \{\text{Spm}(T) \mid T \in \text{OT}_i\}$.

Proof. See Appendix A.9. $\square$

[13] That means that if a sentence δ labels three leaf nodes of T, then $\mu_{\text{Spm}(T)} = 3$.

[14] I.e. $\text{Spm}(S) = \bigoplus\{\text{Spm}(T) \mid T \in S\}$.

Theorem 4 (Soundness theorem for flatten dispute derivation)**.** *Let*

$$\mathrm{dd} = \langle \mathrm{PS}_0, \mathrm{PA}_0, \mathrm{OS}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PS}_n, \mathrm{PA}_n, \mathrm{OS}_n, \mathrm{OA}_n\rangle$$

be a successful flatten dispute derivation for σ*. Then* $\mathrm{PA}_n \mid\!\sim \sigma$.

Proof. From Lemma 16, there exists a proof-tree-based dispute derivation

$$\mathrm{dd}' = \langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$$

for σ such that $\mathrm{PS}_i = \mathrm{Spm}(\mathrm{PT}_i) - \mathrm{PA}_i$ and $\mathrm{OS}_i = \{\mathrm{Spm}(T) \mid T \in \mathrm{OT}_i\}$.

Since $\mathrm{PS}_n = \emptyset$, the set $\mathrm{Spm}(\mathrm{PT}_n)$ contains only assumptions belonging to PA_n. Therefore all proof trees in PT_n are full. From $\mathrm{PA}_n \subseteq \mathrm{Ass}(\mathrm{PT}_n)$, it follows that $\mathrm{Ass}(\mathrm{PT}_n) = \mathrm{PA}_n$.

Since $\mathrm{OS}_n = \{\mathrm{Spm}(T) \mid T \in \mathrm{OT}_n\}$ and $\mathrm{OS}_n = \emptyset$, it follows $\mathrm{OT}_n = \emptyset$.

Therefore dd′ is a successful proof-tree-based dispute derivation. From Theorem 2, it follows $\mathrm{PA}_n \mid\!\sim \sigma$. □

Theorem 5 (Completeness theorem for flatten dispute derivation)**.** *Let* $\mathcal{F}$ *be a finitary ABA framework,* H *be a strictly admissible finite set of assumptions and* σ *be a sentence such that* $H \mid\!\sim \sigma$*. Then there is a successful flatten dispute derivation*

$$\langle \mathrm{PS}_0, \mathrm{PA}_0, \mathrm{OS}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PS}_n, \mathrm{PA}_n, \mathrm{OS}_n, \mathrm{OA}_n\rangle$$

for σ *such that* $\mathrm{PA}_n \subseteq H$.

Proof. From the completeness Theorem 3, there is a successful tree-based dispute derivation

$$\mathrm{dd} = \langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle$$

such that $\mathrm{PA}_n \mid\!\sim \sigma$.

It holds that $\mathrm{OT}_n = \emptyset$ and all proof trees PT_n are full and $\mathrm{Ass}(\mathrm{PT}_n) = \mathrm{PA}_n$.

Let $\mathrm{dd}' = \langle \mathrm{PS}_0, \mathrm{PA}_0, \mathrm{OS}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PS}_n, \mathrm{PA}_n, \mathrm{OS}_n, \mathrm{OA}_n\rangle$ be the flatten dispute derivation for σ as defined in Lemma 15.

Since all proof trees PT_n are full and $\mathrm{Ass}(\mathrm{PT}_n) = \mathrm{PA}_n$, it follows $\mathrm{PS}_n = \mathrm{Spm}(\mathrm{PT}_n) - \mathrm{PA}_n = \emptyset$.

Since $\mathrm{OT}_n = \emptyset$, $\mathrm{OS}_n = \emptyset$. Hence dd′ is successful and $\mathrm{PA}_n \mid\!\sim \sigma$. □

7. Discussion

Dialectical proof procedures for assumption-based argumentation are in general sound but not complete wrt the admissibility semantics. The reason is that these procedures may get stuck in infinite loops. To give a precise characterization of their semantics, we represent these loops as infinite arguments. Infinite arguments are characterized by two distinctive features: i) they attack other arguments in the same way as finite arguments do (and hence showing up in the execution of the dialectical procedures as the finite arguments do), and ii) they attack themselves (and hence can not be accepted as support of their conclusions).

The inclusion of infinite arguments in assumption-based argumentation leads to a new semantics referred to as strict admissibility that is stricter than admissibility, i.e. if a set of arguments is strictly admissible, it is also admissible but not vice versa.

To show that dialectical proof procedures are sound and complete wrt strict admissibility, we proceed in two stages: We first develop a new dialectical proof procedure based explicitly on the notion of arguments and partial arguments (also referred to as full proof trees and partial proof trees) and show the soundness and completeness of this procedure wrt strict admissibility semantics. We then flatten this procedure to get more "traditional" ones that are based on sets and multisets similar to procedures in [15,16].

Our version of flatten dispute derivation differs from the ones in [15,17] in two interesting aspects: 1) the first and more relevant point is that a stronger filtering mechanism for filtering opponent supports in step (1.b) is employed. In [15,17], only one element of OS_i is removed while in our case, all elements of OS_i containing the selected assumption are removed, 2) the second and more or less technical point is that while in our case PA_i is the set of all proponent assumptions that have been attacked by the opponent, in [15,17] PA_i is the set of all assumptions appearing in the supports of proponent proof trees.

As strict admissibility implies admissibility, Theorem 4 could be viewed as generalizing the soundness theorem wrt admissibility of the dialectical proof procedures in [15,17].

[15,17] show the completeness of dialectical proof procedure wrt admissibility semantics for ABA frameworks without infinite arguments. As strict admissibility and admissibility coincide for this restricted class of ABA frameworks, the completeness result in [15,17] could be viewed as a special case of Theorem 5.

[22] proposes structured dispute derivation where instead of proof trees only their supports and conclusions are represented. Gaertner and Toni's procedure could also be viewed as a flattened version of our proof-tree based procedure. [8] has introduced argument graph to deal with a loop in frameworks with a finite set of rules. For frameworks with infinite set of rules as in our Example 1, argument graphs can not capture strict admissiblity.

A novel contribution of this paper is to view partial proof trees as sets of partial proofs allowing us to define infinite arguments simply as the union of an increasing sequence of finite partial arguments. This simplifies the technical machinery needed for understanding properties of the dialectical procedures in no small amount.

Infinite arguments have also been introduced in [25] to capture ambiguity blocking and ambiguity propagating proof theories in defeasible logic as discussed in [30]. As the frameworks of assumption-based argumentation and defeasible logics are based on distinct concepts, it would be interesting to look at the relationship between our approach and the approach in [25].

In an abstract argumentation framework by assuming the existence of preferred extensions we actually implicitly assume the maximality axiom or the axiom of choice. More on this point can be seen in [9,33].

Appendix

A.1. Multisets

We introduce the basic operations of multisets. More about multisets can be found in [4,11,27].
Intuitively, a multiset is like a set but allowing each element to have many instances.

Formally, a multiset is a pair $A = (B, \mu)$ where B is a set referred to as the base set of A, and μ is a function from B into the set of positive integers. The function μ is referred to as the *multiplicity function* of the multiset A and often denoted by μ_A.

For simplicity, we often represent a multiset like a set where some element may occur multiple times like the multiset representing the prime factorization of 40 is represented by $\{2, 2, 2, 5\}$ instead of $(\{2, 5\}, \mu)$ where $\mu = \{(2, 3), (5, 1)\}$.[15]

Definition 14. The union and intersection of two multisets $A = (B, \mu)$, $A' = (B', \mu')$ are defined by:

(1) $A \oplus A' = (B \cup B', \mu + \mu')$ where for $x \in B \cup B'$, $(\mu + \mu')(x) = \mu(x) + \mu'(x)$.
(2) $A \cap A' = (B \cap B', \mu'')$ where $\mu''(x) = \min\{\mu(x), \mu'(x)\}$.

We also introduce two notions of difference and strong difference between multisets and sets in the definition below.

Definition 15. Let $A = (B, \mu)$, $A' = (B', \mu')$ be multisets and S be a set.

(1) The difference between A, A' is defined by:

$$A \setminus A' = \left(B'', \mu''\right)$$

where the following conditions are satisfied:

(a) $B'' = (B \setminus B') \cup \{x \in B \cap B' \mid \mu(x) > \mu'(x)\}$.
(b)

$$\mu''(x) = \begin{cases} \mu(x) & \text{if } x \in B \setminus B' \\ \mu(x) - \mu'(x) & \text{if } x \in B \cap B' \end{cases}$$

(c) The strong difference between A and S is defined by

$$A - S = \left(B \setminus S, \mu'\right)$$

where for each $x \in B \setminus S$, $\mu'(x) = \mu(x)$.

A.2. Proof of Lemma 5

For $T \in \Pi$, define $\Pi_T = \{T' \in \Pi \mid T \subseteq T', \mathrm{ha}(T') \geqslant \mathrm{ha}(T) + 1\}$.
We construct inductively an increasing sequence of proof trees

$$T_0 \subset T_1 \subset \cdots \subset T_n \subset \cdots$$

such that for each $n \geqslant 0$, $T_n \in \Pi$ and $\mathrm{ha}(T_n) = n$ and Π_{T_n} is infinite.

(1) Base Step: We construct T_0 and Π_{T_0}.
Let $\Pi_0 = \{T \in \Pi \mid \mathrm{ha}(T) = 0\}$. It follows $\bigcup\{\Pi_T \mid T \in \Pi_0\} = \Pi \setminus \Pi_0$.
Since the set Π_0 is finite and non-empty by the assumptions, it follows that there is $T_0 \in \Pi_0$ such that Π_{T_0} is infinite.

[15] $\{(2, 3), (5, 1)\}$ represents the function assigning 3 to 2 and 1 to 5.

(2) Suppose we have constructed an increasing sequence

$$T_0 \subset T_1 \subset \cdots \subset T_n$$

such that for each $0 \leqslant i \leqslant n$, $T_i \in \Pi$ and $\text{ha}(T_i) = i$ and Π_{T_i} is infinite.
Define $X = \{T \in \Pi_{T_n} \mid \text{ha}(T) = n + 1\}$. From the lemma assumptions, X is finite.

- We show that X is non-empty and $\bigcup\{\Pi_T \mid T \in X\} \supseteq \Pi_{T_n} \setminus X$.
Since Π_{T_n} is infinite and X is finite, $\Pi_{T_n} \setminus X \neq \emptyset$.
Let $T' \in \Pi_{T_n} \setminus X$. Hence $\text{ha}(T') \geqslant n+2$. From the lemma assumptions, there is $T \in \Pi$ such that $\text{ha}(T) = n + 1$ and $T \subset T'$. Since $T_n \subset T'$, it follows from $T \subset T'$ that T, T_n are compatible. From the lemma assumptions, either $T \subseteq T_n$ or $T_n \subseteq T$. Since $\text{ha}(T) = n+1 > n = \text{ha}(T_n)$, it follows $T_n \subset T$. Thus $T \in \Pi_{T_n}$. From $\text{ha}(T) = n + 1$, it follows $T \in X$. Hence X is not empty. From $T \subset T'$, it follows that $T' \in \Pi_T \subseteq \bigcup\{\Pi_T \mid T \in X\}$. Thus $\Pi_{T_n} \setminus X \subseteq \bigcup\{\Pi_T \mid T \in X\}$.
- It holds obviously that $\bigcup\{\Pi_T \mid T \in X\} \subseteq \Pi_{T_n} \setminus X$.

We have proved that $\bigcup\{\Pi_T \mid T \in X\} = \Pi_{T_n} \setminus X$. Thus $\bigcup\{\Pi_T \mid T \in X\}$ is infinite. Since X is finite, there is $T_{n+1} \in X$ such that $\Pi_{T_{n+1}}$ is infinite and $T_n \subset T_{n+1}$.

A.3. *Proof of Lemma 6*

(1) We prove by induction on n. The statement holds obviously for $n = 0$.
We prove $\text{Ass}(\text{OT}_{n+1}) \cap \text{OA}_{n+1} = \emptyset$ assuming $\text{Ass}(\text{OT}_n) \cap \text{OA}_n = \emptyset$.

- If the procedure step applied at stage n is (1.a) or (2.a), then $\text{Ass}(\text{OT}_{n+1}) = \text{Ass}(\text{OT}_n)$ and $\text{OA}_{n+1} = \text{OA}_n$. The statement holds.
- Suppose the procedure step applied at stage n is (1.b). Then $\text{Ass}(\text{OT}_{n+1}) \subseteq \text{Ass}(\text{OT}_n) \setminus \{\alpha\}$ and $\text{OA}_{n+1} = \text{OA}_n \cup \{\alpha\}$. It holds obviously $\text{Ass}(\text{OT}_{n+1}) \cap \text{OA}_{n+1} \subseteq \text{Ass}(\text{OT}_n) \cap \text{OA}_n = \emptyset$.
- Suppose the procedure step applied at stage n is (2.b). It holds $\text{OA}_{n+1} = \text{OA}_n$. Let $S = \{\exp(T, N, r) \mid \text{bd}(r) \cap \text{OA}_i = \emptyset\}$.
There are two cases:
  - ∗ $S = \emptyset$. Hence $\text{Ass}(\text{OT}_{n+1}) \subseteq \text{Ass}(\text{OT}_n)$. It holds obviously

    $$\text{Ass}(\text{OT}_{n+1}) \cap \text{OA}_{n+1} = \emptyset.$$

  - ∗ $S \neq \emptyset$. Let $T' = \exp(T, N, r') \in S$. It follows that $\text{Ass}(T') = \text{Ass}(T) \cup \text{Ass}(\text{bd}(r'))$. Therefore $\text{Ass}(\text{OT}_{n+1}) = \text{Ass}(\text{OT}_n) \cup X$ where

    $$X = \bigcup\bigl\{\text{Ass}\bigl(\text{bd}(r)\bigr) \mid \text{hd}(r) = \delta, \text{bd}(r) \cap \text{OA}_n = \emptyset\bigr\}.$$

  Thus $X \cap \text{OA}_n = \emptyset$.
  Hence

  $$\text{Ass}(\text{OT}_{n+1}) \cap \text{OA}_{n+1} = \bigl(\text{Ass}(\text{OT}_n) \cup X\bigr) \cap \text{OA}_n = \bigl(\text{Ass}(\text{OT}_n) \cap \text{OA}_n\bigr) \cup (X \cap \text{OA}_n) = \emptyset.$$

(2) Let $\alpha \in \text{OA}_n$ and i be the stage where α is inserted into OA_i, i.e. $\{\alpha\} = \text{OA}_{i+1} \setminus \text{OA}_i$. Therefore $T_0 = \{(\text{root}, \overline{\alpha})\} \in \text{PT}_{i+1}$. Therefore there is a partial proof tree $T \in \text{PT}_n$ such that $T_0 \subseteq T$. Hence $\text{Cl}(T) = \overline{\alpha}$.

(3) $\beta \in \mathrm{PA}_n$ and i be the stage where β is inserted into PA_i, i.e. $\{\alpha\} = \mathrm{PA}_{i+1} \setminus \mathrm{PA}_i$. Therefore $\{(\mathrm{root}, \overline{\beta}\}) \in \mathrm{OT}_{i+1} \setminus \mathrm{OT}_i$. As β can not be inserted into PA_n twice, i is unique.
Let $T \in \bigcup\{\mathrm{OT}_i \mid i \leqslant n\}$ and $T_0 = \{(\mathrm{root}, \mathrm{Cl}(T)\}$. Hence $T_0 \subseteq T$ and there is an i such that $\{T_0\} = \mathrm{OT}_{i+1} \setminus \mathrm{OT}_i$. Therefore there is an assumption $\alpha \in \mathrm{PA}_{i+1} \setminus \mathrm{PA}_i$ such that $\overline{\alpha} = \mathrm{Cl}(T)$. Since $\mathrm{PA}_{i+1} \subseteq \mathrm{PA}_n$, it holds that $\alpha \in \mathrm{PA}_n$. The uniqueness of α comes from the one-one property of the contrary mapping.

(4) We prove by induction on i.
It is clear that the statement holds for $i = 0$. Suppose the statement holds for i. We show that it also holds for $i + 1$.
Assume the contrary that there are $T, T' \in \mathrm{OT}_{i+1}$ such that T, T' are compatible and $T \neq T'$. From the induction hypothesis, it follows that the procedure step applied at stage i in dd can not be (1.a) or (1.b).

- Suppose the procedure step applied at stage i in dd is (2.a). From the induction hypothesis, one of T, T', say T, is the newly introduced proof tree of the form $\{(\mathrm{root}, \overline{\alpha})\}$ with $\{\alpha\} = \mathrm{PA}_{i+1} \setminus \mathrm{PA}_i$. Since $T' \neq T$, $T' \in \mathrm{OT}_i$. Since T, T' are compatible, it follows $T \subseteq T'$. Hence from two previous statements in this lemma, there is a unique $j < i$ such that $\{\alpha\} = \mathrm{PA}_{j+1} \setminus \mathrm{PA}_j$. Since $j < i$, $j + 1 \leqslant i$. Thus $\{\alpha\} \in \mathrm{PA}_i$. Contradiction. Hence this case does not happen.
- Suppose the procedure step applied at stage i in dd is (2.b). From the induction hypothesis, one of T, T', say T, is of the form $\exp(T_1, N, r)$, $T_1 \in \mathrm{OT}_i$. Therefore $T' \in \mathrm{OT}_i$ and $T' \neq T_1$. Therefore T', T_1 is not compatible. Since $T_1 \subseteq T$, it follows $T' \cup T$ is also not compatible. Contradiction. Hence this case can not happen.

Therefore the assumption that T, T' are compatible and $T \neq T'$ leads to a contradiction, and hence impossible.

A.4. *Proof of Lemma 7*

(1) We first prove that $i = j$. Let $\sigma = \mathrm{Cl}(T)$. It follows immediately from Definition 9 that $\{(\mathrm{root}, \sigma)\} \in \mathrm{OT}_i \setminus \mathrm{OT}_{i-1}$ and $\{(\mathrm{root}, \sigma)\} \in \mathrm{OT}_j \setminus \mathrm{OT}_{j-1}$.
From Lemma 6, it follows that there is an assumption $\alpha \in \mathrm{PA}_n$ such that $\overline{\alpha} = \sigma$ and $\alpha \in \mathrm{PA}_{i+1} \setminus \mathrm{PA}_i$ (i.e. α is attacked by the opponent at stage i); and $\alpha \in \mathrm{PA}_{j+1} \setminus \mathrm{PA}_j$ (i.e. α is attacked by the opponent at stage j).
Since $\mathrm{PA}_0 \subseteq \mathrm{PA}_1 \subseteq \cdots \subseteq \mathrm{PA}_n$, it follows from the uniqueness of i, j (Lemma 6), that $i = j$.
We prove by induction that $\mathrm{sq} = \mathrm{sq}'$.
Basic Step: The length of sq, sq′ is 1. It is obvious that $\mathrm{sq} = \mathrm{sq}' = \{(\mathrm{root}, \sigma)\}$.
Inductive Step: Let $\mathrm{sq} = T_0, \ldots, T_k, T_{k+1}$ and $\mathrm{sq}' = T_0, \ldots, T_k, T'_{k+1}$. Since both T_{k+1}, T'_{k+1} are prefixes of T, they are compatible. Because both T_{k+1}, T'_{k+1} belong to OT_{i+k}, it follows from Lemma 6 (last statement) that they are identical.

(2) We prove by induction on n.
Base Step: $n = 0$. The statement holds obviously.
Inductive Step: Suppose the statement holds for n. We prove that it also holds for $n + 1$. There are two cases:

- $T \in \mathrm{OT}_n$. From the induction hypothesis, there is a scope (sq', i), $\mathrm{sq}' = T_0, \ldots, T_j$ of T in dd such that with $T_j = T$ and $i + j = n$. It follows that $(\mathrm{sq}'.T, i)$ is also a scope of T in dd
- $T \notin \mathrm{OT}_n$. Therefore the procedure step applied at stage n is either (2.a) or (2.b).

- ∗ The procedure step applied at stage n is (2.a).
  Therefore $T = \{(\text{root}, \overline{\alpha})\}$ for $\alpha \in \text{PA}_{n+1} \setminus \text{PA}_n$. Then $(T, n+1)$ is a desired scope (of length 1).
- ∗ The procedure step applied at stage n is (2.b). T is hence an immediate expansion of $T' \in \text{OT}_n$. From the induction hypothesis, there is a scope (sq', i), $\text{sq}' = T_0, \ldots, T_j$ of T' in dd such that $i + j = n$ and $T' = T_j$.
  It is obvious that $(\text{sq}'.T, i)$ is a scope of T in dd of length $j+1$ such that $i + j + 1 = n + 1$ and T is the last element in $\text{sq}'.T$.

(3) Since T attacks PT_n, there is an assumption $\alpha \in \text{Ass}(\text{PT}_n)$ such that $\text{Cl}(T) = \overline{\alpha}$. Since dd is successful, $\text{Ass}(\text{PT}_n) = \text{PA}_n$. Lemma 6 implies that there is a unique stage i such that $\{(\text{root}, \text{Cl}(T))\} \in \text{OT}_i \setminus \text{OT}_{i-1}$. Statement 1 of this lemma implies that there exists a unique maximal scope (sq, i) of T of length k in dd.
We show that (sq, i) is a full scope of T. Let $\text{sq} = T_0, \ldots, T_{k-1}$. Suppose the contrary that (sq, i) is not a full scope of T. Hence there is $T' \in \text{OT}_{i+k}$ such that T' is a prefix of T. From the previous statement of this lemma, there is a scope (sq', i) of length $k+1$ of T' in dd such that T' is its last element. From the statement 4 in Lemma 6, sq is a prefix of sq′. Hence $\text{sq}' = \text{sq}.T'$. Contradiction the assumption that (sq, i) is maximal scope of T in dd.

A.5. *Proof of Lemma 12*

By induction on i. It is obvious that the lemma holds for $i = 0$. Suppose the lemma holds for i. We prove that it also holds for $i + 1$.

If the stage i in cdd is not step (2.b), the lemma holds obviously.

Suppose stage i in cdd is step (2.b). Let T be the partial proof tree selected at stage i and N be the selected non-final leaf node.

Let $S = \{\exp(T, N, r) \mid \text{bd}(r) \cap \text{OA}_i = \emptyset\}$

Since $\mathcal{F}$ is finitary, S is finite. Since OT_i is finite (inductive hypothesis), $\text{OT}_{i+1} = (\text{OT}_i \setminus \{T\}) \cup S$ is finite.

Let $T' \in \text{OT}_{i+1}$. If $T' \in \text{OT}_i$ then T' is almost balanced (inductive hypothesis). Let $T' \in \text{OT}_{i+1} \setminus \text{OT}_i$. Therefore $T' \in S$, i.e. $T' = \exp(T, N, r)$ for some rule r. Let N' be a child of N in T'. Therefore $h(N', T') = h(N, T) + 1$. It holds: $\text{ha}(T') = \max\{\text{ha}(T), h(N', T')\} = \max\{\text{ha}(T), h(N, T) + 1\}$. From $\text{ha}(T) \leqslant \text{hi}(T) + 1 \leqslant h(N, T) + 1$, it follows $\text{ha}(T') = h(N, T) + 1$.

There are two cases:

- N is the only non-final leaf node of height $\text{hi}(T)$ in T. Therefore all non-final leaf nodes of T' are of the same height. Thus T' is almost balanced.
- N is not the only non-final leaf node of height $\text{hi}(T)$ in T. Hence $\text{hi}(T) = \text{hi}(T')$. From $\text{hi}(T) = h(N, T)$ and $\text{ha}(T') = h(N, T) + 1$, it follows that $\text{ha}(T') = \text{hi}(T') + 1$. T' is thus most balanced.

A.6. *Proof of Lemma 13*

Suppose the contrary that there exists an infinite H-constrained dispute derivation

$$\text{cdd} = \langle \text{PT}_0, \text{PA}_0, \text{OT}_0, \text{OA}_0, \lambda_0 \rangle, \ldots, \langle \text{PT}_n, \text{PA}_n, \text{OT}_n, \text{OA}_n, \lambda_n \rangle \quad \text{for } \sigma.$$

(1) We show that the set $\Pi = \bigcup\{\mathrm{OT}_i \mid i \geqslant 0\}$ is infinite (i.e. step (2.b) is applied infinitely many times).
Suppose the contrary that the set $\Pi = \bigcup\{\mathrm{OT}_i \mid i \geqslant 0\}$ is finite. Hence there are only finitely many applications of steps (1.b), (2.a) and (2.b) in cdd. Let $\mathrm{PIT} = \{\{(\mathrm{root}, \overline{\alpha})\} \mid \exists i : \alpha \in \mathrm{OA}_i\} \subseteq \bigcup\{\mathrm{PT}_i \mid i \geqslant 0\}$. Thus PIT is finite.
Hence the set $\{A \in \mathrm{AR}_H \mid \exists i, \exists T \in \mathrm{PIT} : \lambda_i(T) = A\}$ is finite. It follows that the number of applications of steps (1.a) is also finite. Thus cdd is finite. Contradiction.
Therefore Π is infinite.

(2) We show that the following conditions are satisfied.

(a) For all $n \geqslant 0$, for each $T \in \Pi$ such that $\mathrm{ha}(T) > n$, there is $T' \in \Pi$ such that $T' \subseteq T$ and $\mathrm{ha}(T') = n$.
Let $T \in \mathrm{OT}_m$. Lemma 7 (second statement) implies that there a scope (sq, i), $\mathrm{sq} \equiv T_0, T_1, \ldots, T_{k-1}$, of T in cdd such that $i + k - 1 = m$ and $T_{k-1} = T$.[16]
From the definition of scope (Definition 9), it follows immediately that $\mathrm{ha}(T_0) = 0$ and for each $0 \leqslant i < k - 1$, $\mathrm{ha}(T_{i+1}) \leqslant \mathrm{ha}(T_i) + 1$. Therefore there is T_j such that $\mathrm{ha}(T_j) = n$.

(b) For all $T, T' \in \Pi$ such that T, T' are compatible, it holds that $T \subseteq T'$ or $T' \subseteq T$.
Let $n, m \geqslant 0$ such that $T \in \mathrm{OT}_n$ and $T' \in \mathrm{OT}_m$.
If $m = n$, it follows immediately from Lemma 6 (fourth statement) that $T = T'$.
Without loss of generality, let $m > n$. From Lemma 7 (second statement), there are scopes (sq, i), (sq', j), $\mathrm{sq} = T_0, \ldots, T_{k-1}$, $\mathrm{sq}' = T'_0, \ldots, T'_{h-1}$, of T, T' respectively such that $i + k - 1 = n$ and $j + h - 1 = m$ and $T_{k-1} = T$ and $T'_{h-1} = T'$.
Since $\mathrm{Cl}(T) = \mathrm{Cl}(T')$, if follows $T_0 = T'_0$ and $i = j$. Further since T, T' are compatible, T_{k-1}, T'_{k-1} are also compatible. Since $T_{k-1}, T'_{k-1} \in \mathrm{OT}_n$, Lemma 6 (fourth statement) implies that $T = T_{k-1} = T'_{k-1} \subseteq T'$.

(c) For each $n \geqslant 0$, the set $\{T \in \Pi \mid \mathrm{ha}(T) \leqslant n\}$ is finite.
Let $\Gamma_n = \{T \in \Pi \mid \mathrm{ha}(T) \leqslant n\}$ and $\Gamma'_n = \{T \in \Pi \mid \mathrm{ha}(T) = n\}$
We prove by induction on n that Γ_n is finite.
Base Step: $n = 0$.
From $\Gamma_0 \subseteq \{T_\alpha \mid \alpha \in H\}$ where $T_\alpha = \{(\mathrm{root}, \overline{\alpha})\}$, it is clear that Γ_0 is finite.
Inductive Step: We show that Γ_{n+1} is finite assuming that Γ_n is finite.
Let Δ be the set of sentences labeling the non-final leaf nodes in proof trees in Γ'_n and

$$k = \max\{n_\delta \mid \delta \in \Delta,\ \text{and } n_\delta \text{ is the number of rules with head } \delta\}$$

$$m = \max\bigl\{m_T \mid T \in \Gamma'_n \text{ and } m_T \text{ is the number of non-final leaf nodes in } T \text{ labeled by some sentence from } \Delta\bigr\}.$$

Since the ABA framework $\mathcal{F}$ is finitary, k is finite.
Since for each $T \in \Gamma'_{n+1}$ there is $T' \in \Gamma'_n$ such that $T' \subset T$, it holds that $|\Gamma'_{n+1}| \leqslant (k+1)^m.|\Gamma'_n|$.
From $\Gamma_{n+1} = \Gamma_n \cup \Gamma'_{n+1}$, the finiteness of Γ_{n+1} follows from the finiteness of Γ_n and Γ'_{n+1}.

[16] As H-constrained dispute derivations could be viewed as special form of dispute derivations, the notion of scope in Definition 9 could be applied directly on H-constrained derivations.

From Lemma 5, there exists increasing sequence $T_0 \subset T_1 \subset \cdots \subset T_k \subset \ldots$ of almost balanced partial proof trees in Π.

(3) Let $T = \bigcup\{T_i \mid i \geqslant 0\}$. From Lemma 4, T is a partial proof tree.
We show that the sequence $\mathrm{sq} \equiv T_0 \subset T_1 \subset \cdots \subset T_k \subset \ldots$ is fair.
Suppose the contrary that sq is not fair. Hence there is T_i and a non-final leaf node M in T_i such that M is also a non-final leaf node in all T_j, $j \geqslant i$.
Let $h(M, T_i) = n$.
Since sq is strictly increasing, there is T_j, $j > i$ such that $\mathrm{ha}(T_j) = n + 3$. Since all trees in sq are almost balanced (Lemma 12), it follows that $\mathrm{ha}(T_j) < \mathrm{hi}(T_j) + 1$. Since M is a non-final leaf node in T_j, it holds that $h(M, T_j) \geqslant \mathrm{hi}(T_j)$.
Therefore $n + 3 = \mathrm{ha}(T_j) < \mathrm{hi}(T_j) + 1 < h(M, T_j) + 1 = n + 1$. Contradiction.
We have proved that sq is fair. Therefore T is an argument.
The sequence sq can only be constructed by infinitely many applications of procedure step (2.b). Since step (2.b) is selected only if step (1.b) is not possible, it follows that no assumption appearing in T could be attacked by any argument in AR_H. Hence T is an infinite argument attacking H but no argument in AR_H attacks T. Contradiction since H is strictly admissible.
Therefore cdd does not exist. The lemma holds.

A.7. *Proof of Lemma 14*

The proof is by induction on n.
Base Step: $n = 0$. Obvious
Inductive Step. We show that the lemma holds for $n + 1$ assuming that it holds for n.
There are four cases:

(1) The proponent makes a move at stage n according to step (1.a) in Definition 12.

$$\mathrm{PS}_{n+1} \cap \mathrm{PA}_{n+1} = \big((\mathrm{PS}_n \setminus \{\delta\}) \oplus (\mathrm{bd}(r) \setminus \mathrm{PA}_n)\big) \cap \mathrm{PA}_n = \big((\mathrm{PS}_n \setminus \{\delta\}) \cap \mathrm{PA}_n\big) \cup \big((\mathrm{bd}(r) \setminus \mathrm{PA}_n) \cap \mathrm{PA}_n\big).$$

From $((\mathrm{bd}(r) \setminus \mathrm{PA}_n)) \cap \mathrm{PA}_n = \emptyset$, and $(\mathrm{PS}_n \setminus \{\delta\}) \cap \mathrm{PA}_n \subseteq \mathrm{PS}_n \cap \mathrm{PA}_n = \emptyset$, it follows

$$\mathrm{PS}_{n+1} \cap \mathrm{PA}_{n+1} = \emptyset.$$

Since $\mathrm{OS}_{n+1} = \mathrm{OS}_n$ and $\mathrm{OA}_{n+1} = \mathrm{OA}_n$, and $\mathrm{OA}_n \cap \bigoplus \mathrm{OS}_n = \emptyset$, it follows

$$\mathrm{OA}_{n+1} \cap \bigoplus \mathrm{OS}_{n+1} = \emptyset.$$

(2) The proponent makes a move at stage n according to step (1.b) in Definition 12.

$$\mathrm{PS}_{n+1} \cap \mathrm{PA}_{n+1} = \big(\mathrm{PS}_n \oplus \{\overline{\alpha}\}\big) \cap \mathrm{PA}_n = (\mathrm{PS}_n \cap \mathrm{PA}_n) \cup \big(\{\overline{\alpha}\} \cap \mathrm{PA}_n\big) = \emptyset.$$

$$\begin{aligned}\Big(\bigoplus \mathrm{OS}_{n+1}\Big) \cap \mathrm{OA}_{n+1} &= \bigoplus\big(\mathrm{OS}_i \setminus \{S \in \mathrm{OS}_i \mid \alpha \in S\}\big) \cap \big(\mathrm{OA}_i \cup \{\alpha\}\big) \\ &= \bigcup\big\{S \cap \big(\mathrm{OA}_i \cup \{\alpha\}\big) \mid S \in \mathrm{OS}_i, \alpha \notin S\big\} \\ &= \bigcup\{S \cap \mathrm{OA}_i \mid S \in \mathrm{OS}_i, \alpha \notin S\} \subseteq \bigcup\{S \cap \mathrm{OA}_i \mid S \in \mathrm{OS}_i\} = \emptyset.\end{aligned}$$

(3) The opponent makes a move at stage n according to step (2.a) in Definition 12.
From $\alpha \in \mathrm{PS}_n$ and the induction hypothesis that $\mathrm{PS}_n \cap \mathrm{PA}_n = \emptyset$, it follows $\alpha \notin \mathrm{PA}_n$.
As PS_{n+1} is obtained from PS_n by removing all instances of α, and $\alpha \notin \mathrm{PA}_n$, it follows

$$\mathrm{PS}_{n+1} \cap \mathrm{PA}_{n+1} = \mathrm{PS}_{n+1} \cap \big(\mathrm{PA}_n \cup \{\alpha\}\big) = \mathrm{PS}_{n+1} \cap \mathrm{PA}_n \subseteq \mathrm{PS}_n \cap \mathrm{PA}_n = \emptyset.$$

$$\begin{aligned} \mathrm{OA}_{n+1} \cap \bigoplus \mathrm{OS}_{n+1} &= \mathrm{OA}_n \cap (\bigoplus(\mathrm{OS}_n \oplus \big\{\{\overline{\alpha}\}\big\}) \\ &= \Big(\mathrm{OA}_n \cap \bigoplus \mathrm{OS}_n\Big) \cup \big(\{\overline{\alpha}\} \cap \mathrm{OA}_n\big) = \emptyset. \end{aligned}$$

(4) The opponent makes a move at stage n according to step (2.b) in Definition 12.
It is obvious that

$$\mathrm{PS}_{n+1} \cap \mathrm{PA}_{n+1} = \mathrm{PS}_n \cap \mathrm{PA}_n = \emptyset.$$

$$\begin{aligned} &\mathrm{OA}_{n+1} \cap \bigoplus \mathrm{OS}_{n+1} \\ &\quad = \mathrm{OA}_n \cap \bigoplus\Big(\big(\mathrm{OS}_n \setminus \{S\}\big) \oplus \big\{\big(S \setminus \{\delta\}\big) \oplus \mathrm{bd}(r) \mid \mathrm{hd}(r) = \delta, \mathrm{bd}(r) \cap \mathrm{OA}_n = \emptyset\big\}\Big) \\ &\quad \subseteq \Big(\mathrm{OA}_n \cap \bigoplus \mathrm{OS}_n\Big) \cup \bigcup\big\{\mathrm{OA}_n \cap \big(\big(S \setminus \{\delta\}\big) \oplus \mathrm{bd}(r)\big) \mid \mathrm{hd}(r) = \delta, \mathrm{bd}(r) \cap \mathrm{OA}_n = \emptyset\big\} \\ &\quad = \bigcup\big\{\mathrm{OA}_n \cap \big(S \oplus \mathrm{bd}(r)\big) \mid \mathrm{hd}(r) = \delta, \mathrm{bd}(r) \cap \mathrm{OA}_n = \emptyset\big\} \\ &\quad = \mathrm{OA}_n \cap S \subseteq \mathrm{OA}_n \cap \bigoplus \mathrm{OS}_n = \emptyset. \end{aligned}$$

A.8. *Proof of Lemma 15*

We prove the lemma by induction on n.
Base Step: $n = 0$. Obvious.
Inductive Step: We prove that the lemma holds for $n + 1$ assuming it holds for n.
There are four cases to consider:

(1) At stage n in dd, step (1.a) is applied. Hence

$$\mathrm{PT}_{n+1} = \big(\mathrm{PT}_n \setminus \{T\}\big) \cup \big\{\exp(T, N, r)\big\}$$

for some proof tree $T \in \mathrm{PT}_n$ and a non-final leaf node N in T labeled by the head δ of r such that $\mathrm{bd}(r) \cap \mathrm{OA}_i = \emptyset$.
Hence

$$\begin{aligned} \mathrm{Spm}(\mathrm{PT}_{n+1}) &= (\mathrm{Spm}\big(\mathrm{PT}_n \setminus \{T\}\big) \oplus \mathrm{Spm}\big(\exp(T, N, r)\big) \\ &= \big(\mathrm{Spm}(\mathrm{PT}_n) \setminus \mathrm{Spm}(T)\big) \oplus \mathrm{Spm}\big(\exp(T, N, r)\big). \end{aligned}$$

From $\mathrm{Spm}(\exp(T, N, r)) = (\mathrm{Spm}(T) \setminus \{\delta\}) \oplus \mathrm{bd}(r)$, it follows

$$\mathrm{Spm}(\mathrm{PT}_{n+1}) = \big(\mathrm{Spm}(\mathrm{PT}_n) \setminus \{\delta\}\big) \oplus \mathrm{bd}(r).$$

Hence

$$\mathrm{Spm}(\mathrm{PT}_{n+1}) - \mathrm{PA}_{n+1} = \big(\mathrm{Spm}(\mathrm{PT}_n) \setminus \{\delta\}\big) \oplus \mathrm{bd}(r)) - \mathrm{PA}_{n+1}.$$

Since $\mathrm{PA}_{n+1} = \mathrm{PA}_n$, we have

$$\begin{aligned}\mathrm{PS}_{n+1} &= \mathrm{Spm}(\mathrm{PT}_{n+1}) - \mathrm{PA}_{n+1} = \big(\mathrm{Spm}(\mathrm{PT}_n) \setminus \{\delta\}\big) \oplus \mathrm{bd}(r)) - \mathrm{PA}_n \\ &= \big(\big(\mathrm{Spm}(\mathrm{PT}_n) \setminus \{\delta\}\big) - \mathrm{PA}_n\big) \oplus \big(\mathrm{bd}(r) - \mathrm{PA}_n\big)\end{aligned}$$

Because $\delta \notin \mathrm{PA}_n$, it holds $(\mathrm{Spm}(\mathrm{PT}_n) \setminus \{\delta\}) - \mathrm{PA}_n = (\mathrm{Spm}(\mathrm{PT}_n) - \mathrm{PA}_n) \setminus \{\delta\} = \mathrm{PS}_n \setminus \{\delta\}$.
Since both $\mathrm{bd}(r)$ and PA_n are sets, it holds $\mathrm{bd}(r) - \mathrm{PA}_n = \mathrm{bd}(r) \setminus \mathrm{PA}_n$
Thus

$$\mathrm{PS}_{n+1} = \big(\mathrm{PS}_n \setminus \{\delta\}\big) \oplus \big(\mathrm{bd}(r) \setminus \mathrm{PA}_n\big).$$

Because $\mathrm{OT}_{n+1} = \mathrm{OT}_n$ and $\mathrm{PA}_{n+1} = \mathrm{PA}_n$, it follows obviously from inductive hypothesis that $\mathrm{OS}_{n+1} = \mathrm{Spm}(\mathrm{OT}_{n+1})$.
We have proved that $\langle \mathrm{PS}_{n+1}, \mathrm{PA}_{n+1}, \mathrm{OS}_{n+1}, \mathrm{OA}_{n+1}\rangle$ can be obtained from $\langle \mathrm{PS}_n, \mathrm{PA}_n, \mathrm{OS}_n, \mathrm{OA}_n\rangle$ by application of step (1.a) of the flatten dispute derivation procedure.

(2) At stage n in dd, step (1.b) is applied. Hence

$$\mathrm{PT}_{n+1} = \mathrm{PT}_n \cup \big\{\big\{(\mathrm{root}, \overline{\alpha})\big\}\big\} \quad \text{and} \quad \mathrm{OT}_{n+1} = \mathrm{OT}_n \setminus \big\{T' \in \mathrm{OT}_n \mid \alpha \in \mathrm{Ass}\big(T'\big)\big\}.$$

It follows that

$$\begin{aligned}\mathrm{PS}_{n+1} &= \mathrm{Spm}(\mathrm{PT}_{n+1}) - \mathrm{PA}_{n+1} = \big(\mathrm{Spm}(\mathrm{PT}_n) \oplus \mathrm{Spm}\big(\big\{(\mathrm{root}, \overline{\alpha})\big\}\big)\big) - \mathrm{PA}_n = \mathrm{PS}_n \cup \{\overline{\alpha}\}. \\ \mathrm{OS}_{n+1} &= \big\{\mathrm{Spm}(T) \mid T \in \mathrm{OT}_{n+1}\big\} = \big\{\mathrm{Spm}(T) \mid T \in \mathrm{OT}_n, \alpha \notin \mathrm{Ass}(T)\big\} \\ &= \big\{\mathrm{Spm}(T) \mid T \in \mathrm{OT}_n\big\} \setminus \big\{\mathrm{Spm}\big(T'\big) \mid T' \in \mathrm{OT}_n, \alpha \in \mathrm{Ass}\big(T'\big)\big\} \\ &= \mathrm{OS}_n \setminus \{S \mid S \in \mathrm{OS}_n, \alpha \in S\}.\end{aligned}$$

We have proved that $\langle \mathrm{PS}_{n+1}, \mathrm{PA}_{n+1}, \mathrm{OS}_{n+1}, \mathrm{OA}_{n+1}\rangle$ can be obtained from $\langle \mathrm{PS}_n, \mathrm{PA}_n, \mathrm{OS}_n, \mathrm{OA}_n\rangle$ by application of step (1.b) of the flatten dispute derivation procedure.

(3) At stage n in dd, step (2.a) is applied. Hence

$$\begin{aligned}&\mathrm{PT}_{n+1} = \mathrm{PT}_n \quad \text{and} \\ &\mathrm{OT}_{n+1} = \mathrm{OT}_n \cup \big\{\big\{(\mathrm{root}, \overline{\alpha})\big\}\big\}.\end{aligned}$$

From $\mathrm{PA}_{n+1} = \mathrm{PA}_n \cup \{\alpha\}$, it follows that

$$\mathrm{PS}_{n+1} = \mathrm{Spm}(\mathrm{PT}_{n+1}) - \mathrm{PA}_{n+1} = \big(\mathrm{Spm}(\mathrm{PT}_n) - \mathrm{PA}_n\big) - \{\alpha\} = \mathrm{PS}_n - \{\alpha\}.$$

It follows that

$$\mathrm{OS}_{n+1} = \big\{\mathrm{Spm}(T) \mid T \in \mathrm{OT}_{n+1}\big\} = \big\{\mathrm{Spm}(T) \mid T \in \mathrm{OT}_n\big\} \oplus \big\{\mathrm{Spm}\big(\big\{(\mathrm{root}, \overline{\alpha})\big\}\big)\big\} = \mathrm{OS}_n \oplus \big\{\{\overline{\alpha}\}\big\}.$$

We have proved that $\langle \mathrm{PS}_{n+1}, \mathrm{PA}_{n+1}, \mathrm{OS}_{n+1}, \mathrm{OA}_{n+1}\rangle$ can be obtained from $\langle \mathrm{PS}_n, \mathrm{PA}_n, \mathrm{OS}_n, \mathrm{OA}_n\rangle$ by application of step (2.a) of the flatten dispute derivation procedure.

(4) At stage n in dd, step (2.b) is applied. Hence

$$\mathrm{PT}_{n+1} = \mathrm{PT}_n, \qquad \mathrm{PA}_{n+1} = \mathrm{PA}_n \quad \text{and}$$
$$\mathrm{OT}_{n+1} = \big(\mathrm{OT}_n \setminus \{T\}\big) \cup \big\{\exp(T, N, r) \mid \mathrm{bd}(r) \cap \mathrm{OA}_n = \emptyset\big\}$$

Therefore it is obvious that $\mathrm{PS}_{n+1} = \mathrm{PS}_n$.

$$\begin{aligned}&\big\{\mathrm{Spm}\big(T'\big) \mid T' \in \mathrm{OT}_{i+1}\big\}\\ &\quad = \big\{\mathrm{Spm}\big(T'\big) \mid T' \in \mathrm{OT}_i, T' \neq T\big\}) \oplus \big\{\mathrm{Spm}\big(\exp(T, N, r)\big) \mid \mathrm{bd}(r) \cap \mathrm{OA}_n = \emptyset\big\}.\end{aligned}$$

Let $S = \mathrm{Spm}(T)$ and δ be the sentence labeling N in T. Hence

$$\begin{aligned}\mathrm{Spm}\big(\exp(T, N, r)\big) &= \big(S \setminus \{\delta\}\big) \oplus \mathrm{bd}(r)\\ \mathrm{OS}_{n+1} &= \big\{\mathrm{Spm}\big(T'\big) \mid T' \in \mathrm{OT}_{n+1}\big\}\\ &= \big\{\mathrm{Spm}\big(T'\big) \mid T' \in \mathrm{OT}_n, T' \neq T\big\}) \oplus \big\{\mathrm{Spm}\big(\exp(T, N, r)\big) \mid \mathrm{bd}(r) \cap \mathrm{OA}_n = \emptyset\big\}\\ &= \big(\mathrm{OS}_n \setminus \{S\}\big) \oplus \big\{\big(S \setminus \{\delta\}\big) \oplus \mathrm{bd}(r) \mid \mathrm{bd}(r) \cap \mathrm{OA}_n = \emptyset\big\}\end{aligned}$$

We have proved that $\langle \mathrm{PS}_{n+1}, \mathrm{PA}_{n+1}, \mathrm{OS}_{n+1}, \mathrm{OA}_{n+1}\rangle$ can be obtained from $\langle \mathrm{PS}_n, \mathrm{PA}_n, \mathrm{OS}_n, \mathrm{OA}_n\rangle$ by application of step (2.b) of the flatten dispute derivation procedure.

A.9. *Proof of Lemma 16*

We prove the lemma by induction on n.
Base Step: $n = 0$. Obvious.
Inductive Step: We prove that the lemma holds for $n + 1$ assuming it holds for n.
There are four cases to consider:

(1) At stage n in dd, step (1.a) is applied.
Let $\mathrm{PT}_{n+1} = (\mathrm{PT}_n \setminus \{T\}) \cup \{\exp(T, N, r)\}$ and $\mathrm{OT}_{n+1} = \mathrm{OT}_n$.
From the induction hypothesis, it is obvious that

$$\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \ldots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle, \langle \mathrm{PT}_{n+1}, \mathrm{PA}_{n+1}, \mathrm{OT}_{n+1}, \mathrm{OA}_{n+1}\rangle$$

is a tree-based dispute derivation.

(a) Let $\delta \in \mathrm{PS}_n$ be the selected non-assumption sentence. From the inductive hypothesis, there is a tree $T \in \mathrm{PT}_n$ such that $\delta \in \mathrm{Spm}(T)$.
It holds:

$$\mathrm{Spm}(\mathrm{PT}_{n+1}) = \big(\mathrm{Spm}(\mathrm{PT}_n) \setminus \mathrm{Spm}(T)\big) \oplus \mathrm{Spm}\big(\exp(T, N, r)\big)$$

From $\text{Spm}(\exp(T, N, r)) = \text{Spm}(T) \setminus \{\delta\} \oplus \text{bd}(r)$, it follows

$$\begin{aligned}\text{Spm}(\text{PT}_{n+1}) &= \big(\text{Spm}(\text{PT}_n) \setminus \text{Spm}(T)\big) \oplus \text{Spm}\big(\exp(T, N, r)\big)\\ &= \big(\big(\text{Spm}(\text{PT}_n) \setminus \text{Spm}(T)\big) \oplus \big(\text{Spm}(T) \setminus \{\delta\}\big) \oplus \text{bd}(r)\big)\\ &= \big(\text{Spm}(\text{PT}_n) \setminus \{\delta\}\big) \oplus \text{bd}(r).\end{aligned}$$

Since $\text{PA}_{n+1} = \text{PA}_n$, we have

$$\begin{aligned}&\text{Spm}(\text{PT}_{n+1}) - \text{PA}_{n+1}\\ &\quad = \big(\big(\text{Spm}(\text{PT}_n) \setminus \{\delta\}\big) \oplus \text{bd}(r)\big) - \text{PA}_n\\ &\quad = \big(\text{Spm}(\text{PT}_n) \setminus \{\delta\}\big) - \text{PA}_n) \oplus \big(\text{bd}(r) - \text{PA}_n\big)\\ &\quad = \big(\text{PS}_n \setminus \{\delta\}\big) \oplus \big(\text{bd}(r) - \text{PA}_n\big) = \big(\text{PS}_n \setminus \{\delta\}\big) \oplus \big(\text{bd}(r) \setminus \text{PA}_n\big) = \text{PS}_{n+1}\end{aligned}$$

We have proved $\text{PS}_{n+1} = \text{Spm}(\text{PT}_{n+1}) - \text{PA}_{n+1}$.
Since δ is not an assumption, from $\text{PT}_{n+1} = (\text{PT}_n \setminus \{T\}) \cup \{\exp(T, N, r)\}$, it follows $\text{Ass}(\text{PT}_{n+1}) = \text{Ass}(\text{PT}_n) \cup \text{Ass}(\text{bd}(r))$. From $\text{PA}_n \subseteq \text{Ass}(\text{PT}_n)$ (induction hypothesis) and $\text{PA}_{n+1} = \text{PA}_n$, it holds: $\text{PA}_{n+1} \subseteq \text{Ass}(\text{PT}_{n+1})$.

(b) Because $\text{OS}_{n+1} = \text{OS}_n$, $\text{OT}_{n+1} = \text{OT}_n$ and $\text{OS}_n = \{\text{Spm}(T) \mid T \in \text{OT}_n\}$ (induction hypothesis), it follows $\text{OS}_{n+1} = \{\text{Spm}(T) \mid T \in \text{OT}_{n+1}\}$.

(2) At stage n in dd, step (1.b) is applied. Let $\alpha \in S$, $S \in \text{OS}_n$, be the selected assumption to be attacked.
From the inductive hypothesis, $\text{PS}_n = \text{Spm}(\text{PT}_n) - \text{PA}_n$.
Let $\text{PT}_{n+1} = \text{PT}_n \cup \{\{(\text{root}, \overline{\alpha})\}\}$, and $\text{OT}_{n+1} = \text{OT}_n \setminus \{T' \in \text{OT}_n \mid \alpha \in \text{Ass}(T')\}$. Therefore

$$\langle \text{PT}_0, \text{PA}_0, \text{OT}_0, \text{OA}_0\rangle, \dots, \langle \text{PT}_n, \text{PA}_n, \text{OT}_n, \text{OA}_n\rangle, \langle \text{PT}_{n+1}, \text{PA}_{n+1}, \text{OT}_{n+1}, \text{OA}_{n+1}\rangle$$

is a tree-based dispute derivation.

(a) From $\text{PA}_{n+1} = \text{PA}_n$, it holds:

$$\begin{aligned}\text{Spm}(\text{PT}_{n+1}) - \text{PA}_{n+1} &= (\text{Spm}(\text{PT}_n) \oplus \text{Spm}\big(\big\{(\text{root}, \overline{\alpha})\big\}\big) - \text{PA}_n\\ &= \big(\text{Spm}(\text{PT}_n) \oplus \{\overline{\alpha}\}\big) - \text{PA}_n\\ &= \big(\text{Spm}(\text{PT}_n) - \text{PA}_n\big) \oplus \{\overline{\alpha}\} = \text{PS}_n \oplus \{\overline{\alpha}\} = \text{PS}_{n+1}.\end{aligned}$$

We have proved that $\text{PS}_{n+1} = \text{Spm}(\text{PT}_{n+1}) - \text{PA}_{n+1}$.
Since $\text{PA}_{n+1} = \text{PA}_n$ and $\text{PT}_n \subset \text{PT}_{n+1}$, it holds obviously $\text{PA}_{n+1} \subseteq \text{Spm}(\text{PT}_{n+1})$.

(b) Since $\text{OS}_n = \{\text{Spm}(T) \mid T \in \text{OT}_n\}$ (inductive hypothesis), there is a tree $T' \in \text{OT}_n$ such that $S = \text{Spm}(T')$.
It holds:

$$\begin{aligned}\big\{\text{Spm}(T) \mid T \in \text{OT}_{n+1}\big\} &= \big\{\text{Spm}(T) \mid T \in \text{OT}_n\big\} \setminus \big\{\text{Spm}\big(T'\big) \mid T' \in \text{OT}_n, \alpha \in \text{Ass}\big(T'\big)\big\}\\ &= \text{OS}_n \setminus \{S \mid S \in \text{OS}_n, \alpha \in S\} = \text{OS}_{n+1}.\end{aligned}$$

We have proved that $\mathrm{OS}_{n+1} = \{\mathrm{Spm}(T) \mid T \in \mathrm{OT}_{n+1}\}$.

(3) At stage n in dd, step (2.a) is applied.
Let $\mathrm{PT}_{n+1} = \mathrm{PT}_n$, and $\mathrm{OT}_{n+1} = \mathrm{OT}_n \cup \{\{(\mathrm{root}, \overline{\alpha})\}\}$. Therefore from the induction hypothesis,

$$\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \dots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle, \langle \mathrm{PT}_{n+1}, \mathrm{PA}_{n+1}, \mathrm{OT}_{n+1}, \mathrm{OA}_{n+1}\rangle$$

is a tree-based dispute derivation.

(a) It holds:

$$\mathrm{Spm}(\mathrm{PT}_{n+1}) - \mathrm{PA}_{n+1} = \mathrm{Spm}(\mathrm{PT}_n) - (\mathrm{PA}_n \cup \{\alpha\}) = (\mathrm{Spm}(\mathrm{PT}_n) - \mathrm{PA}_n) - \{\alpha\} = \mathrm{PS}_n - \{\alpha\} = \mathrm{PS}_{n+1}.$$

We have proved that $\mathrm{PS}_{n+1} = \mathrm{Spm}(\mathrm{PT}_{n+1}) - \mathrm{PA}_{n+1}$.
From $\alpha \in \mathrm{PS}_n \subseteq \mathrm{Spm}(\mathrm{PT}_n)$, it follows $\alpha \in \mathrm{Ass}(\mathrm{PT}_n)$.
From $\mathrm{PA}_n \subseteq \mathrm{Ass}(\mathrm{PT}_n) = \mathrm{Ass}(\mathrm{PT}_{n+1})$ and $\mathrm{PA}_{n+1} = \mathrm{PA}_n \cup \{\alpha\}$ and $\alpha \in \mathrm{Ass}(\mathrm{PT}_n)$, it follows $\mathrm{PA}_{n+1} = \mathrm{PA}_n \cup \{\alpha\} \subseteq \mathrm{Ass}(\mathrm{PT}_{n+1})$.

(b) It holds that

$$\{\mathrm{Spm}(T) \mid T \in \mathrm{OT}_{n+1}\} = \{\mathrm{Spm}(T) \mid T \in \mathrm{OT}_n\} \oplus \{\mathrm{Spm}(\{(\mathrm{root}, \overline{\alpha})\})\} = \mathrm{OS}_n \oplus \{\{\overline{\alpha}\}\} = \mathrm{OS}_{n+1}.$$

We have proved that $\mathrm{OS}_{n+1} = \{\mathrm{Spm}(T) \mid T \in \mathrm{OT}_{n+1}\}$.

(4) At stage n in dd, step (2.b) is applied.
Let $\delta \in S$, $S \in \mathrm{OS}_n$, be the selected non-assumption sentence. From the inductive hypothesis, there is a tree $T \in \mathrm{OT}_n$ such that $S = \mathrm{Spm}(T)$.
Let $\mathrm{PT}_{n+1} = \mathrm{PT}_n$ and $\mathrm{OT}_{n+1} = (\mathrm{OT}_n \setminus \{T\}) \cup \{\exp(T, N, r) \mid \mathrm{bd}(r) \cap \mathrm{OA}_n = \emptyset\}$.
Therefore $\langle \mathrm{PT}_0, \mathrm{PA}_0, \mathrm{OT}_0, \mathrm{OA}_0\rangle, \dots, \langle \mathrm{PT}_n, \mathrm{PA}_n, \mathrm{OT}_n, \mathrm{OA}_n\rangle, \langle \mathrm{PT}_{n+1}, \mathrm{PA}_{n+1}, \mathrm{OT}_{n+1}, \mathrm{OA}_{n+1}\rangle$ is a tree-based dispute derivation.

(a) Because $\mathrm{PA}_{n+1} = \mathrm{PA}_n$, $\mathrm{PS}_{n+1} = \mathrm{PS}_n$ and $\mathrm{PS}_n = \mathrm{Spm}(\mathrm{PT}_n) - \mathrm{PA}_n$ (induction hypothesis), it follows immediately that $\mathrm{PS}_{n+1} = \mathrm{Spm}(\mathrm{PT}_{n+1}) - \mathrm{PA}_{n+1}$.
Since $\mathrm{PT}_{n+1} = \mathrm{PT}_n$ and $\mathrm{PA}_{n+1} = \mathrm{PA}_n$ and $\mathrm{PA}_n \subseteq \mathrm{Ass}(\mathrm{PT}_n)$, it holds $\mathrm{PA}_{n+1} \subseteq \mathrm{Ass}(\mathrm{PT}_{n+1})$.

(b) Because $\mathrm{Spm}(T) = S$, it follows that $\mathrm{Spm}(\exp(T, N, r)) = (S \setminus \{\delta\}) \oplus \mathrm{bd}(r)$ where δ labels N and $\mathrm{hd}(r) = \delta$
It holds

$$\begin{aligned} &\{\mathrm{Spm}(T') \mid T' \in \mathrm{OT}_{n+1}\} \\ &\quad = (\{\mathrm{Spm}(T') \mid T' \in \mathrm{OT}_n\} \setminus \{\mathrm{Spm}(T)\}) \oplus \{\mathrm{Spm}(\exp(T, N, r)) \mid \mathrm{bd}(r) \cap \mathrm{OA}_n = \emptyset\} \\ &\quad = (\mathrm{OS}_n \setminus \{S\}) \oplus \{S \setminus \{\delta\} \oplus \mathrm{bd}(r) \mid \mathrm{bd}(r) \cap \mathrm{OA}_n = \emptyset\} \\ &\quad = \mathrm{OS}_{n+1} \end{aligned}$$

We have proved that $\mathrm{OS}_{n+1} = \{\mathrm{Spm}(T') \mid T' \in \mathrm{OT}_{n+1}\}$.

References

[1] O. Arieli and C. Straßer, Logical argumentation by dynamic proof systems, *Theoretical Computer Science* **781** (2019), 63–91, https://www.sciencedirect.com/science/article/pii/S0304397519301252. doi:10.1016/j.tcs.2019.02.019.

[2] P. Baroni, F. Cerutti, M. Giacomin and G. Simari (eds), *Proceedings of Conference on Computational Models of Arguments*, IOS Press, 2010.

[3] P. Besnard, S. Doutre and A. Hunter (eds), *Proceedings of Conference on Computational Models of Arguments Computational Models of Arguments*, IOS Press, 2008.

[4] W.D. Blizard, Multiset theory, *Notre Dame Journal of Formal Logic* **30**(1) (1988), 36–66. doi:10.1305/ndjfl/1093634995.

[5] A. Bondarenko, P.M. Dung, R.A. Kowalski and F. Toni, An abstract, argumentation-theoretic approach to default reasoning, *Artif. Intell.* **93** (1997), 63–101. doi:10.1016/S0004-3702(97)00015-5.

[6] M. Caminada and L. Amgoud, On the evaluation of argumentation formalisms, *Artificial Intelligence* **171** (2007), 286–310. doi:10.1016/j.artint.2007.02.003.

[7] C. Cayrol, S. Doutre and J. Mengin, On decision problems related to the preferred semantics for argumentation frameworks, *J. Log. Comput.* **13**(3) (2003), 377–403. doi:10.1093/logcom/13.3.377.

[8] R. Craven and F. Toni, Argument graphs and assumption-based argumentation, *Artificial Intelligence* **233** (2016), 1–59. doi:10.1016/j.artint.2015.12.004.

[9] B.A. Davey and H.A. Priestley, *Introduction to Lattices and Order*, Cambridge University Press, 2002.

[10] M. Denecker and E. Ternovska, A logic of nonmonotone inductive definitions, *ACM Trans. Comput. Logic* **9**(2) (2008). doi:10.1145/1342991.1342998.

[11] N. Dershowitz and Z. Manna, Proving termination with multiset orderings, in: *Automata, Languages and Programming*, Springer, Berlin Heidelberg, 1979, pp. 188–202. doi:10.1007/3-540-09510-1_15.

[12] P.M. Dung, Logic programming as dialogue games, Technical Report, Division of Computer Science, Asian Institute of Technology, Thailand (submitted to LPNMR 1993), 1993.

[13] P.M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artif. Intell.* **77**(2) (1995), 321–358. doi:10.1016/0004-3702(94)00041-X.

[14] P.M. Dung, An argumentation theoretic foundation for logic programming, *Journal of Logic Programming* **22** (1995), 151–177. doi:10.1016/0743-1066(95)94697-X.

[15] P.M. Dung, R.A. Kowalski and F. Toni, Dialectic proof procedures for assumption-based, admissible argumentation, *Artif. Intell.* **170**(2) (2006), 114–159. doi:10.1016/j.artint.2005.07.002.

[16] P.M. Dung, R.A. Kowalski and F. Toni, Assumption-based argumentation, in: *Argumentation in AI*, Springer-Verlag, 2009.

[17] P.M. Dung, P. Mancarella and F. Toni, Computing ideal skeptical argumentation, *Artificial Intelligence* **171** (2007).

[18] P.M. Dung and P.M. Thang, A modular framework for dialectical dispute in argumentation, in: *Proc. of IJCAI*, 2009.

[19] P.E. Dunne and T. Bench-Capon (eds), *Proceedings of Conference on Computational Models of Arguments Computational Models of Arguments*, IOS Press, 2006.

[20] P.E. Dunne and T.J.M. Bench-Capon, Two party immediate response disputes: Properties and efficiency, *Artif. Intell.* **149**(2) (2003), 221–250. doi:10.1016/S0004-3702(03)00076-6.

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

[22] D. Gaertner and F. Toni, Computing arguments and attacks in assumption-based argumentation, *IEEE Intelligent Systems* **22**(6) (2007), 24–33. doi:10.1109/MIS.2007.105.

[23] A.J. Garcia and G.R. Simari, Defeasible logic programming: An argumentative approach, *TPLP* **4**(1–2) (2004), 95–138.

[24] A.J. Garcia and G.R. Simari, Defeasible logic programming: DeLP servers, contextual queries and explanation for answers, *J. Arguments and Computation* (2014).

[25] G. Governatori, M.J. Maher, G. Antoniou and D. Billington, Argumentation semantics for defeasible logic, *J. Log. Comput.* **14**(5) (2004), 675–702. doi:10.1093/logcom/14.5.675.

[26] J.W. Lloyd, *Foundations of Logic Programming*, Springer Verlag, 1987.

[27] Z. Manna and R. Waldinger, *The Logical Basis for Computer Programming*, Addison-Wesley Professional, 1985.

[28] S. Modgil and M. Caminada, Proof theories and algorithms for abstract argumentation frameworks, in: *Argumentation in AI*, Springer Verlag, 2009.

[29] S. Modgil and H. Prakken, A general account of argumentation with preferences, *Artificial Intelligence* **195** (2013), 361–397, http://www.sciencedirect.com/science/article/pii/S0004370212001361. doi:10.1016/j.artint.2012.10.008.

[30] D. Nute, Defeasible logic programming: DeLP servers, contextual queries and explanation for answers, in: *Proc. 20th Hawaii International Conference on System Science*, IEEE Press, 1987.

[31] J.L. Pollock, Defeasible reasoning, *Cognitive Science* **11**(4) (1987), 481–518. doi:10.1207/s15516709cog1104_4.

[32] I. Rahwan and G. Simari (eds), *Handbook of Argumentation in AI*, Springer Verlag, 2009.

[33] C. Spanring, Set and graph theoretic investigations in abstract argumentation, PhD thesis, University of Liverpool, 2017.

[34] P.M. Thang and P.M. Dung, Tribute to Guillermo Simari, in: *Infinite Arguments and Semantics of Assumption-Based Argumentation*, College Publication, 2019.
[35] F. Toni, A generalised framework for dispute derivations in assumption-based argumentation, *Artif. Intell.* **195** (2013), 1–43. doi:10.1016/j.artint.2012.09.010.
[36] F. Toni, A tutorial on assumption-based argumentation, *Journal of Arguments and Computation* (2013).
[37] B. Verheij, A labeling approach to the computation of credulous acceptance in argumentation, in: *JCAI 2007*, M. Veloso, ed., Morgan Kaufmann, 2007, pp. 623–628.
[38] B. Verheij, S. Szeider and S. Woltran (eds), *Proceedings of Conference on Computational Models of Arguments*, IOS Press, 2012.
[39] G. Vreeswijk and H. Prakken, Credulous and sceptical argument games for preferred semantics, in: *JELIA 2000*, M. Ojeda-Aciego, I.P. de Guzmán, G. Brewka and L.M. Pereira, eds, Lecture Notes in Computer Science, Vol. 1919, Springer, 2000, pp. 239–253.