THE STATUS OF ARGUMENTS IN ABSTRACT ARGUMENTATION FRAMEWORKS. A TABLEAUX METHOD*

Gustavo A. Bodanza[1]
https://orcid.org/0000-0001-6254-7683
[1] *Universidad Nacional del Sur and CONICET*
Departamento de Humanidades, and
Instituto de Investigaciones Económicas y Sociales del Sur
8000 Bahía Blanca, Argentina
gbodanza@iiess-conicet.gob.ar

Enrique Hernández-Manfredini[2]
[2] *Universidade de Aveiro*
Departamento de Matemática
Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
hernandezmanfredini@ua.pt

Article Info:
CDD: 100

*This research was partially supported by Agencia Nacional de Promoción de la Investigación, el Desarrollo Tecnológico y la Innovación (PICT 2017-1702 and PICT 2022-75) and Universidad Nacional del Sur (PGI 24/I265), Argentina. Five anonymous referees are acknowledged for valuable comments and criticisms on this and previous versions that notably improved the work. The first author appreciates the invitation of the Department of Mathematics of the Universidad de Aveiro, Portugal, for a visit in September 2019, during which this paper was developed.

Received: 19.03.2022; Revised: 09.02.2023 ; Accepted: 10.02.2023
https://doi.org/10.1590/0100-6045.2023.V46N2.GE

Keywords:
Argumentation frameworks
Extension semantics
Tableaux

Abstract: Dung's argumentation frameworks are formalisms widely used to model interaction among arguments. Although their study has been profusely developed in the field of Artificial Intelligence, it is not common to see its treatment among those less connected to computer science within the logical-philosophical community. In this paper we propose to bring to that audience a proof-theory for argument justification based on tableaux, very similar to those the Logic students are familiar with. The tableaux enable to calculate whether an argument or subset of arguments are accepted or rejected in accordance to Dung's preferred and grounded extension-based semantics. Soundness and completeness results regarding those semantics are provided.

1 Introduction: Abstract argumentation

Argumentation is a process in which arguments are advanced by contending parties, each one trying to defeat the arguments of the other. The abstract model by Dung [23] is a widely used tool for the representation of attacks among arguments and the study of argument justification. The model is abstract in the sense that there are no considerations about the nature of arguments or the attack relation: all that matters is what arguments have been presented and the attacks between them. Several semantics have been proposed in terms of this model. Extension semantics, as proposed by Dung and other authors, define subsets of the arguments in the framework that can be defended together, according to some specific conditions for defense. The subject is widely studied in the field of Artificial Intelligence and several algorithms have been proposed to compute the status of arguments in correspondence with different extension semantics [28, 29, 24, 15]. Those developments presuppose the knowledge of computational techniques that can result in discouraging Logic students in the field of Philosophy. For this reason, the aim of this work is to give that audience a formalism of familiar traits to access the study of argumentation frameworks.

We propose a logical representation of proofs for the acceptance/rejection of arguments. Our idea is inspired partly by dialogue games [28] and partly by the analytic tableaux method for logics [6, 34]. The latter is a familiar decision method for logicians, and it is widely used for teaching because of its simplicity and intuitiveness. It consists of a procedure for deciding the satisfiability of a formula or a set of formulae by developing a tree (the tableau) through the application of rules that "expand" each formula according to the truth table of its main connective and its valuation (true or false). In a sequence of steps the procedure ends and an answer is found: satisfiable or not satisfiable. Similarly, we want a method to decide if sentences like 'argument x is *in*' (*in* meaning acceptable) or 'argument x is *out*' (*out* meaning rejectable) are satisfiable. From dialogue games [37, 13, 37] we take the idea of developing a tree with root in a focus argument and following the paths traced by the arguments on the attack line. But unlike dialogue games, the root is not an argument to be defended by a player *Pro* but a sentence claiming that the argument is either *in* or *out*. This last option introduces a novelty, since dialogues always begin by advancing an acceptance thesis, not a rejection thesis. Then, we can calculate the status of the attackers. To represent that, arguments are attached with a *label* I or O (for *in* and *out*, respectively), and two scheme-rules are used to attach labels to the attackers, inspired by *labeling* semantics [27, 7]. If an argument x is assumed to be *out*, then -according to the argumentation framework- *some* of its attackers $y_1, \ldots, y_k$ must be *in*. Then, a rule taking as premise the sentence $x^{\mathtt{O}}$ yields k branches, each headed by a sentence $y_i^{\mathtt{I}}$ $(1 \leq i \leq k)$. This represents the fact that *at least one* accepted attacker y_i of x is enough to reject x. On the other hand, if x is assumed to be *in*, then *all* its attackers must be *out*. So, the rule that takes as premise the sentence $x^{\mathtt{I}}$ does not open new branches, but puts in a sequence all the sentences $y^{\mathtt{O}}$ such that y is an attacker of x, meaning that all of them must be true at the same time. The method proceeds by applying these rules, expanding each new sentence. If at the end, that is, when no further rules can be applied, some of the branches remain "open" (i.e. without contradiction), then we say that the root sentence is satisfiable. We define argumentation versions of the notions of satisfiability and validity with respect to an argumentation framework, and show how the method can be applied to the problem of determining credulous/skeptical acceptance/rejection for Dung's preferred and grounded semantics. On the other hand, the method is not intended to compute whole extensions. Nevertheless, particular sets of arguments can be checked for weak and strong admissibility, i.e. their inclusion in preferred and grounded extensions, respectively [3].

Related works comprehend *dialectical trees* [17, 16] and *dialogue games* [37, 13, 37] for argument justification. Our work is also related to logical proof-theoretic approaches to argument justification, like those of [1], who proposed a sequent-based formalism, and that of [4, 5], who also take into account logically structured arguments. These and other works will be discussed in Section 6.

The text is organized as follows. In Section 2 we recall the basics of Dung's argumentation frameworks and extension semantics. In Section 3 the tableaux method is introduced, defining the rules and the key concepts of argument satisfiability and validity. Sections 4 and 5 show soundness and completeness results in terms of the correspondences of specific outcomes of the tableaux, as proof-theoretic formalisms, with argument justification according to extension semantics: in Section 4 we deal with credulous and skeptical justification with respect to preferred semantics, while in Section 5 we deal with (skeptical) justification with respect to grounded semantics. In Section 6 we discuss related work and in Section 7 we give our conclusions.

2 Background: Dung's argumentation frameworks

Dung's argumentation frameworks are very simple structures for modeling attack among arguments:

Definition **2.1** An *abstract argumentation framework* is a pair $AF = \langle A,\ R\rangle$, where A is a set of abstract entities called 'arguments' and $R \subseteq A \times A$ represents an attack relation among arguments. We will say that x *attacks* y iff $(x, y) \in R$. Moreover, we will also say that, for any subsets $S, S' \subseteq A$, and for any argument $x \in A$, S *attacks* (*is attacked by*) x iff there exists $y \in S$ such that $(y, x) \in R$ (resp., $(y, x) \in R$), and S *attacks* (*is attacked by*) S' iff there exists $x \in S'$ such that S attacks (resp., is attacked by) x.

The following simple examples show how argument situations can be modeled through argumentation frameworks, abstracting every element which is not recognized as either an argument or an attack, including internal structure, evidence, strength of the attack, etc.

EXAMPLE **2.1** Consider the two arguments involved in Tweety's famous example of non-monotonic reasoning:
a: *Tweety is a bird. Birds usually fly. Then, Tweety flies.*
b: *Tweety is a penguin. Penguins do not fly. Then, Tweety does not fly.*
Assuming that b attacks a[1], we represent the situation through the argumentation framework $\langle A, R\rangle$, where $A = \{a, b\}$ and $R = \{(b, a)\}$. The model can also be diagrammed by a digraph: Figure 1.

$$a \longleftarrow b$$

Figure 1: Example 2.1.

EXAMPLE **2.2** Consider the arguments:
a: *The government of X cannot negotiate with the government of Y because the government of Y does not even recognize the government of X.*
b: *The government of X does not recognize the government of Y either.*
Assuming that a and b attack each other, we represent the situation through the argumentation framework $\langle A, R\rangle$, where $A = \{a, b\}$ and $R = \{(a, b), (b, a)\}$ (Figure 2).

Intuitively, the acceptance of an argument depends on the way it can be defended from attacks. Argument

[1] Here we are assuming that b is an *undercutting defeater* of a, i.e., a is a reason not to believe b's conclusion (Tweety flies) on the basis of b's premise (Tweety is a bird), but not vice versa [30]. Indeed, from a common sense point of view, since a uses more specific information than b, it is impractical to use b as an attacker of a. Note that Dung's model is purportedly nonspecific about the nature of 'attack', hence we have to resort to external criteria when representing argumentative situations.

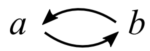


Figure 2: Example 2.2.

defense is understood in this framework as counterargument: an argument a attacked by an argument b can be defended by an argument c that attacks b. But different requirements on this intuition lead to different notions of justification or warrant. A substantial difference between the different types of semantics that we will discuss is whether or not self-defense is enabled. A justification criterion can be defined as an *extension semantics* $\mathcal{S}$ that yields, for every argumentation framework AF, a family $\mathcal{E}_{\mathcal{S}}(AF) \subseteq 2^A$ of *extensions* of AF (under $\mathcal{S}$) (notation taken from [3]). An argument is said *credulously* justified under a semantics $\mathcal{S}$ if and only if it belongs to *some* extension $E \in \mathcal{E}_{\mathcal{S}}(AF)$, and is said to be *skeptically* justified if and only if it belongs to *every* extension $E \in \mathcal{E}_{\mathcal{S}}(AF)$. Dung defines "grounded" and "preferred" semantics as ways of capturing skeptical and credulous behaviors, respectively.

Definition **2.2** [23] Given an argumentation framework $AF = \langle A, R \rangle$, an argument $x \in A$ and a subset $S \subseteq A$, we say that

- x is *acceptable* w.r.t. S iff for every argument $y \in A$ such that $(y, x) \in R$, there exists some argument $z \in S$ such that $(z, y) \in R$,

- S is *conflict-free* iff, for every pair of arguments $x, y \in S$, $(x, y) \notin R$, i.e., the attack relation does not hold for any pair of arguments belonging to S,

- S is *admissible* iff each $x \in S$ is acceptable w.r.t. S and S is *conflict-free*,

- S is a *preferred extension* iff S is maximally (w.r.t. $\subseteq$) admissible,

- S is a *complete extension* iff S is conflict-free and is a fixed point of $F(\cdot)$, where $F(S) = \{x : x$ is acceptable w.r.t. $S\}$ (every preferred extension is a complete extension),

- S is the *grounded extension* iff S is the least (w.r.t. $\subseteq$) complete extension (the grounded extension is well-defined and unique, for every argumentation framework).

The grounded extension is contained in the intersection of all the preferred extensions, but they do not always coincide. Hence, grounded semantics defines a more skeptical criterion than preferred semantics. Credulous acceptance in preferred semantics lies in the arbitrary choice of one of the preferred extensions, in case there are more than one. In grounded semantics, credulous acceptance reduces to skeptical acceptance since there can only be one grounded extension. In Example 2.1, $\{b\}$ is the only extension for all the above defined semantics (hence, b is justified both skeptically and credulously under those semantics). In Example 2.2, $\emptyset$ is the grounded extension and $\{a\}$ and $\{b\}$ are preferred extensions, while all $\emptyset$, $\{a\}$ and $\{b\}$ are complete extensions, i.e. no argument is skeptically justified under any semantics, but both $\{a\}$ and $\{b\}$ are credulously justified under preferred and complete semantics. For a case where the grounded extension does not coincide with the intersection of all the preferred extensions, see AF in Figure 13, Section 4, where the grounded extension is $\emptyset$ but the intersection of all the preferred extensions is $\{a\}$.

3 Tableaux for argumentation frameworks

We will define the tableaux scheme rules of a generic argumentation framework $AF = \langle A, R \rangle$. Consider a language L whose formulae are of the form $x^{\mathtt{V}}$, where $x \in A$ and $\mathtt{V} \in \{\mathtt{I}, \mathtt{O}\}$.[2] The formula '$x^{\mathtt{I}}$' is understood as '$x$ is in' and '$x^{\mathtt{O}}$' is understood as 'x is out'. Then the rules will be such that, given $x^{\mathtt{I}}$, we can deduce $y^{\mathtt{O}}$ for *every* argument y which is an attacker of x and, given $x^{\mathtt{O}}$, we can deduce $y^{\mathtt{I}}$ for *some* argument y which is an attacker of x. To formally introduce the rules we will refer to the set of attackers of an argument x as: $Attackers(x) = \{y : (y, x) \in R\}$.[3]

DEFINITION **3.1** (Scheme rules) Given an argumentation framework $AF = \langle A,\ R \rangle$, for every argument $x \in A$ let $\{y_1, \ldots, y_k\}$ be an arbitrary (but fixed) enumeration of $Attackers(x)$. We define the *scheme rules* as follows:

If $Attackers(x) \neq \emptyset$ then

(In x)

$$\frac{x^{\mathtt{I}}}{\begin{array}{c} y_1^{\mathtt{O}} \\ \vdots \\ y_k^{\mathtt{O}} \end{array}}$$

(Out x)

$$\frac{x^{\mathtt{O}}}{y_1^{\mathtt{I}} \mid \ldots \mid y_k^{\mathtt{I}}}$$

otherwise

[2]Strictly, 'x' in a formula '$x^{\mathtt{V}}$' stands for the name of argument x. For the sake of simplicity, we use the same symbol for an argument and its name.

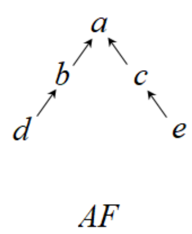
[3]We will also refer to the attackers of a set of arguments $S \subseteq A$ as $Attackers(S) = \bigcup_{x \in S} Attackers(x)$.

(In x) (Out x)

$$\frac{x^{\mathtt{I}}}{-} \qquad\qquad \frac{x^{\mathtt{O}}}{\times}$$

The rule (In x) yields a list of formulae, each one labeling with $\mathtt{O}$ an attacker y_i of x. This list must be interpreted as a conjunction. If x has no attackers then the rule yields – (dash), meaning that $x^{\mathtt{I}}$ is true (i.e., it is correct that x is *in*, since it has no attackers). The rule (Out x) splits the proof into as many branches as attackers have x, each one labeled with $\mathtt{I}$. The meaning of this rule is that the assumption that x is *out* implies that at least one of its attackers must be *in*, i.e., branches denote inclusively disjunct possibilities. If x has no attackers then the rule yields $\times$, meaning that $x^{\mathtt{O}}$ is false (i.e., x cannot be *out* since it has no attackers).

Before formally introducing the notion of tableau, let us informally describe the intended procedure. The method will proceed by applying the (In ...) and (Out ...) rules on unexpanded sentences until no further rule can be applied. Take, for instance, the argumentation framework $AF = \langle\{a, b, c, d, e\},\ \{(b, a),\ (c, a),\ (d, b),\ (e, c)\}\rangle$ (Figure 3, left), and let us develop a tableau for $a^{\mathtt{I}}$ (Figure 3, right), as a way of finding if a can be successfully defended. Then we first obtain $b^{\mathtt{O}}$ and $c^{\mathtt{O}}$ after applying the rule (In a) over $a^{\mathtt{I}}$. Next we expand the formulae $b^{\mathtt{O}}$ and $c^{\mathtt{O}}$, one at a time. This can be done non-deterministically but, as we will see, it can be done strategically in order to shorten the proof. If we choose $b^{\mathtt{O}}$, then we apply the rule (Out b). This rule yields $d^{\mathtt{I}}$ in the fourth line. In the fifth line we obtain – (dash), as a consequence of applying (In d) in the previous line. This is so because d has no attackers. Then we go back to line 3 to expand $c^{\mathtt{O}}$, obtaining $e^{\mathtt{I}}$ in the sixth line. The tableau is finished after applying (In e), which yields a dash, since no formula remains to be expanded. The meaning of this tableau is that a can be successfully defended, since the only branch shows -without contradiction- that every attack (i.e., those of b and c) can be responded by unchallenged arguments (i.e., d and e, respectively). The tableaux are formally defined as follows:



1. $a^{\mathtt{I}}$	
2. $b^{\mathtt{O}}$	(In a, 1)
3. $c^{\mathtt{O}}$	(In a, 1)
4. $d^{\mathtt{I}}$	Out b, 2)
5. –	(In d, 4)
6. $e^{\mathtt{I}}$	(Out c, 3)
7. –	(In e, 6)

Figure 3: Tableau for $a^{\mathtt{I}}$ in AF.

Definition **3.2** Given an argumentation framework $AF = \langle A, R \rangle$ and an argument $x \in A$, a *tableau* (*on* AF) for a formula $x^{\mathtt{V}}$, where $\mathtt{V} \in \{\mathtt{I}, \mathtt{O}\}$, is a labeled tree $T(x^{\mathtt{V}})$ where:

- the labels are formulae of the language,
- the root of $T(x^{\mathtt{V}})$ is labeled with $x^{\mathtt{V}}$,
- for any non-root node n, the label of n is the result of the application of (In ...) or (Out ...) over the label of some ancestor node of n that has not been expanded before.

When the symbol $\times$ is introduced in some branch of a tableau, that means that the branch is closed (no further developments are needed) as a result of an inconsistency. The derivation of a sentence $x^{\mathtt{O}}$ yields contradiction whenever x has no attackers, since it should be deemed *in*, hence, in those cases the rule (Out...) yields $\times$. But another reason for closure is the occurrence of two formulae of the form $x^{\mathtt{I}}$ and $x^{\mathtt{O}}$, respectively, in the same branch. This can happen because of the presence of odd-length cycles in R. The following meta-rule closes the branch in those cases, as it involves contradiction:

(Closure)

$$\frac{\begin{array}{c} x^{\mathtt{V}} \\ \vdots \\ x^{-\mathtt{V}} \end{array}}{\times}$$

where $\mathtt{V} \in \{\mathtt{I}, \mathtt{O}\}$ *and* $-\mathtt{V} \in \{\mathtt{I}, \mathtt{O}\} \setminus \{\mathtt{V}\}$ *(the branch should not be developed further).*

As a result, we can characterize the tableaux according to their branches as follows:

Definition **3.3** A branch of a tableau is:

- *closed* iff $\times$ occurs in it,
- *open* iff it is not closed, and
- *grounded* iff it is open and ω does not occur in it.

A tableau is:

- *closed* iff every branch is closed,
- *open* iff it is not closed, and
- *grounded* iff it has a branch that is grounded.

The first two properties in the above definition are obviously exhaustive and exclude each other. Also, the third one implies the second one, but the reciprocal does not hold.

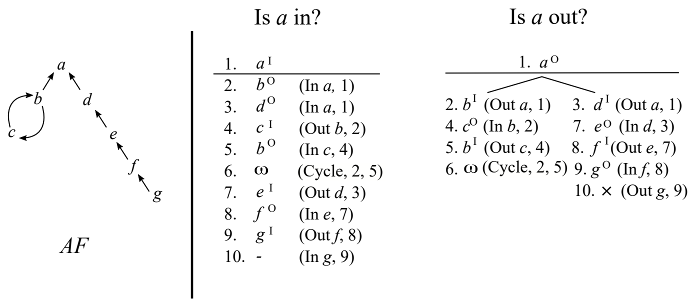


Figure 4: Example 3.1.

EXAMPLE **3.1** (Figure 4) Let $AF = \langle\{a, b, c, d, e, f, g\}, \{(b, a), (b, c), (c, b), (d, a), (e, d), (f, e), (g, f)\}\rangle$. Then both the tableaux for $a^{\mathtt{I}}$ and $a^{\mathtt{O}}$ are open but not grounded. This can be interpreted as the answer for both the questions *Is a in?* and *Is a out?* being 'credulously, yes'. As we will see later, this means that a belongs to some preferred extension while it does not belong to some other preferred extension.

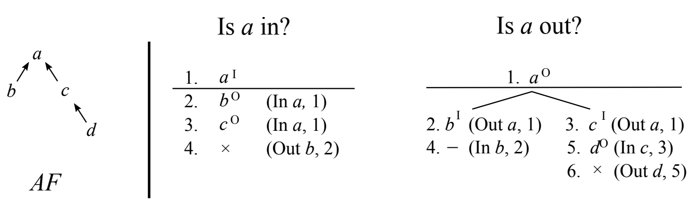


Figure 5: Example 3.2.

EXAMPLE **3.2** (Figure 5) Let $AF = \langle\{a, b, c, d\}, \{(b, a), (c, a), (d, c)\}\rangle$. The tableau for $a^{\mathtt{I}}$ has only one branch which is closed, hence the tableau is closed. The tableau for $a^{\mathtt{O}}$ is open, but we have two branches, one headed by $b^{\mathtt{I}}$, which is grounded, and another one headed by $c^{\mathtt{I}}$, which is closed. So, the answer for *Is a in?* can be interpreted as 'skeptically (i.e. certainly), no'; while the answer for *Is a out?* can be interpreted as 'skeptically, yes'. As we will see later, this means that a does not belong to any extension.

DEFINITION **3.4** Given $AF = \langle A, R \rangle$ and $x \in A$, a formula $x^{\mathtt{V}}$, where $\mathtt{V} \in \{\mathtt{I}, \mathtt{O}\}$, is:

- *satisfiable (in AF)* iff the tableau $T(x^{\mathtt{V}})$ for $x^{\mathtt{V}}$ is open, and
- *valid (in AF)* iff the tableau $T(x^{\mathtt{V}})$ for $x^{\mathtt{V}}$ is grounded.

Obviously, all valid formulae are satisfiable, but not vice versa. The argumentative meaning of satisfiability and validity is related to credulity and skepticism, respectively, as it will be formally established in the next sections. In the meantime, we can informally note the following. An open tableau for $x^{\mathtt{I}}$ (i.e., $x^{\mathtt{I}}$ being satisfiable) means that x can be accepted by adopting an admissible set of arguments to which x belongs (and that admissible set exists). A grounded tableau for $x^{\mathtt{I}}$ (i.e., $x^{\mathtt{I}}$ being valid), moreover, means that there is no admissible set of arguments attacking x. An open tableau for $x^{\mathtt{O}}$ means that x can be rejected by adopting an admissible set of arguments that attacks x (and that admissible set exists). A grounded tableau for $x^{\mathtt{O}}$, moreover, means that there is no admissible set of arguments to which x belongs. Finally, closed tableaux are indicative of arguments that *cannot* be accepted/rejected on *any* rational basis.

EXAMPLE **3.3** (Example 3.2 revisited). $a^{\mathtt{I}}$ is not satisfiable, while $a^{\mathtt{O}}$ is satisfiable and valid. This means that argument a will not be accepted (i.e., it will be rejected) even by credulous agents.

EXAMPLE **3.4** (Example 3.1 revisited). Both $a^{\mathtt{I}}$ and $a^{\mathtt{O}}$ are satisfiable but not valid. This means that credulous agents may accept or reject a (since both $a^{\mathtt{I}}$ and $a^{\mathtt{O}}$ are satisfiable), but skeptics will reject it (since $a^{\mathtt{I}}$ is not valid).

The following result is immediate:

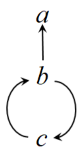
PROPOSITION **3.1** For every tableau T, $x^{\mathtt{V}}$ occurs in an open (grounded) branch of T iff $x^{\mathtt{V}}$ is satisfiable (valid).

Note that different tableaux can be obtained for the same argument, since the order in which the rules expand a tree is not always determined. However, it is easy to see that, for two different tableaux $T_1(x^{\mathtt{V}})$ and $T_2(x^{\mathtt{V}})$ of a given sentence $x^{\mathtt{V}}$ ($\mathtt{V} \in \{\mathtt{I}, \mathtt{O}\}$), there exists an open/grounded/closed branch in $T_1(x^{\mathtt{V}})$ iff there exists an open/grounded/closed (resp.) branch in $T_2(x^{\mathtt{V}})$. This clearly leads to an equivalence relation among tableaux. For this reason, from now on we will take the license to speak of "the tableau" of a given sentence, understanding that we speak of any tableau of its equivalence class.

3.1 Shortening the tableaux

Attack cycles of even length do not yield contradiction, but lead to infinite computations. To avoid that, repetitions in the same "thread" of reasoning can be aborted. With "thread" we refer the sequence of inferences within the same path of attacks, which can be traced through the line numbers of the ancestors of the line at stake. To make this explicit, we can put the ancestors' line numbers in a set on the right of each relevant line as, for instance, in the tableau for

$a^{\mathtt{I}}$, where $AF = \langle \{a, b, c\}, \{(b, a), (c, b), (b, c)\}\rangle$ (Figure 6). The following "cycle-breaking" meta-rule will stop the cycling inferences:



AF

1.	$a^{\mathtt{I}}$	$\emptyset$	
2.	$b^{\mathtt{O}}$	(In a, 1)	$\{1\}$
3.	$c^{\mathtt{I}}$	(Out b, 2)	$\{1,2\}$
4.	$b^{\mathtt{O}}$	(In c, 3)	$\{1,2,3\}$
5.	$c^{\mathtt{I}}$	(Out b, 4)	$\{1,2,3,4\}$
	$\vdots$		

Figure 6: Cyclical attacks yield infinite tablaux.

$$\begin{array}{ll} & \text{(Cycle } x^{\mathtt{V}}\text{)} \\ x^{\mathtt{V}} & < ancestors_1 > \\ \vdots & \\ x^{\mathtt{V}} & < ancestors_2 > \\ \hline \omega & \end{array}$$

where $\mathtt{V} \in \{\mathtt{I}, \mathtt{O}\}$, $< ancestors_1 >$ *and* $< ancestors_2 >$ *are the sets of line numbers of the respective ancestors and* $< ancestors_1 > \subseteq < ancestors_2 >$ *(*$x^{\mathtt{V}}$ *should not be developed further).*

The symbol 'ω' is introduced after the repeated formula as a signal of the aborted cycle. In the above example, 'ω' would be placed in line 5 after the repetition of $b^{\mathtt{O}}$ in line 4, given that the ancestors of line 2 are a subset of those of line 4:

1.	$a^{\mathtt{I}}$	$\emptyset$	
2.	$b^{\mathtt{O}}$	(In a, 1)	$\{1\}$
3.	$c^{\mathtt{I}}$	(Out b, 2)	$\{1,2\}$
4.	$b^{\mathtt{O}}$	(In c, 3)	$\{1,2,3\}$
5.	ω	(Cycle b, 2, 4)	$\{1,2,3,4\}$

Placing this symbol does not prevent for eventual further developments of the branch, contrarily to the case of '×'. More specifically, rules can continue to be applied on previous lines that remain to be expanded (see Figure 4, left tableau, lines 6 et seq.).

Moreover, since one argument can attack several arguments, we can have repetitions even without cycles, as in the following case.

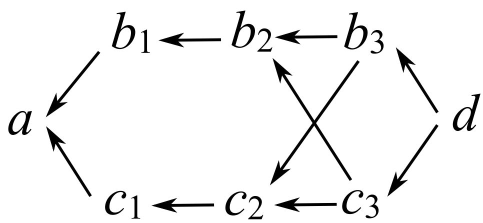


Figure 7: Example 3.5.

Example **3.5** [8] Let $AF = \langle\{a, b_1, b_2, b_3, c_1, c_2, c_3, d\}$, $\{(b_1, a), (c_1, a), (b_2, b_1), (c_2, c_1), (b_3, b_2), (c_3, c_2), (d, b_3),$ $(d, c_3), (c_3, b_2), (b_3, c_2)\}\rangle$ (Figure 7). We get the following tableau for $a^{\mathtt{I}}$:

1.	$a^{\mathtt{I}}$	$\emptyset$	
2.	$b_1^{\mathtt{O}}$	(In a, 1)	$\{1\}$
3.	$c_1^{\mathtt{O}}$	(In a, 1)	$\{1\}$
4.	$b_2^{\mathtt{I}}$	(Out b_1, 2)	$\{1,2\}$
5.	$b_3^{\mathtt{O}}$	(In b_2, 4)	$\{1,2,4\}$
6.	$c_3^{\mathtt{O}}$	(In b_2, 4)	$\{1,2,4\}$
7.	$d^{\mathtt{I}}$	(Out b_3, 5)	$\{1,2,4,5\}$

8.	–	(In d, 7)	$\{1, 2, 4, 5, 7\}$
9.	$d^{\mathtt{I}}$	(Out c_3, 6)	$\{1, 2, 4, 6\}$
10.	–	(In d, 9)	$\{1, 2, 4, 5, 9\}$
11.	$c_2^{\mathtt{I}}$	(Out c_1, 3)	$\{1, 3\}$
12.	$b_3^{\mathtt{O}}$	(In c_2, 11)	$\{1, 3, 11\}$
13.	$c_3^{\mathtt{O}}$	(In c_2, 11)	$\{1, 3, 11\}$
14.	$d^{\mathtt{I}}$	(Out b_3, 12)	$\{1, 3, 11, 12\}$
15.	$d^{\mathtt{I}}$	(Out c_3, 13)	$\{1, 3, 11, 13\}$
16.	–	(In d, 14)	$\{1, 3, 11, 12, 14\}$
17.	–	(In d, 15)	$\{1, 3, 11, 12, 15\}$

Lines 12 to 17 repeat lines 5 to 10. Several repetitions of sentences ($b_3^{\mathtt{O}}$, $c_3^{\mathtt{O}}$, $d^{\mathtt{I}}$) are not due to cycles but correspond to new "threads", as can be seen from the sets of ancestors. So, for example, although it is necessary to expand $b_3^{\mathtt{O}}$ from its appearance in line 5, we must avoid expanding it again from the appearance in line 12. To stop such redundant replay of sequences we introduce the following meta-rule:

$$\text{(Replay } x^{\mathtt{V}}\text{)}$$

$$\frac{\begin{array}{ll} x^{\mathtt{V}} & < ancestors_1 > \\ \vdots & \\ x^{\mathtt{V}} & < ancestors_2 > \end{array}}{-}$$

where $\mathtt{V} = \{\mathtt{I}, \mathtt{O}\}$, $< ancestors_1 >$ and $< ancestors_2 >$ are the sets of the line numbers of the respective ancestors and $< ancestors_1 > \not\subseteq < ancestors_2 >$ ($x^{\mathtt{V}}$ must not be expanded from its last appearance).

Example **3.6** (Example 3.5 revisited) Applying the Replay rule we can get:

1.	$a^{\mathtt{I}}$	$\emptyset$	
2.	$b_1^{\mathtt{O}}$	(In a, 1)	$\{1\}$
3.	$c_1^{\mathtt{O}}$	(In a, 1)	$\{1\}$
4.	$b_2^{\mathtt{I}}$	(Out b_1, 2)	$\{1,2\}$
5.	$b_3^{\mathtt{O}}$	(In b_2, 4)	$\{1,2,4\}$
6.	$c_3^{\mathtt{O}}$	(In b_2, 4)	$\{1,2,4\}$
7.	$d^{\mathtt{I}}$	(Out b_3, 5)	$\{1,2,4,5\}$
8.	–	(In d, 7)	$\{1,2,4,5,7\}$
9.	$d^{\mathtt{I}}$	(Out c_3, 6)	$\{1,2,4,6\}$
10.	–	(Replay d, 7, 9)	$\{1,2,4,5,6,7,9\}$
11.	$c_2^{\mathtt{I}}$	(Out c_1, 3)	$\{1,3\}$
12.	$b_3^{\mathtt{O}}$	(In c_2, 11)	$\{1,3,11\}$
13.	–	(Replay b_3, 5, 11)	$\{1,2,3,4,11\}$

The application of the Replay rule in line 13 avoids further expansions from $b_3^{\mathtt{O}}$ from line 12. In line 10, the application is correct but useless, since d has no attackers (it does not introduce any difficulties anyway).

Checking the ancestors through the (Cycle) and (Replay) rules prevents superfluous computations, as is common in some tree-based dialogue procedures [8]. From here on, we will omit the numbers of the ancestors in the tableaux for clarity and simplicity.

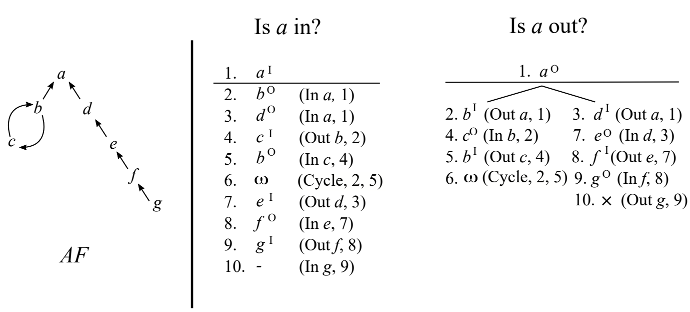


Figure 8: Two tableaux for $a^{\mathtt{I}}$.

In order to avoid other futile computations, all the meta-rules (Closure), (Cycle), and (Replay) will always have priority over the rules (In ...) and (Out ...). Moreover, (Closure) will always have priority over (Cycle) and (Replay). Other strategies to gain efficiency include applying (In ...) rules with priority over (Out ...) rules. For example, Figure 8 shows two tableaux for the same sentence, the one on the right being shorter due to the application of (In f) on line 4, anticipating the bifurcation that the rule (Out b) would yield. Repeated subtrees or branches can also be conveniently pruned, though we will not specify further rules here.

3.2 Extending tableaux for sets of arguments

The notion of tableau can be easily extended to sets of sentences with the purpose of checking if all the sentences of the set are jointly satisfiable.

Definition **3.5** Given an argumentation framework $AF = \langle A, R \rangle$, a *tableau* (*on* AF) for a set of formulae $S = \{\phi_1, \dots, \phi_n\}$[4] such that for each i, ϕ_i has the form $x^{\mathtt{V}}$, where $x \in A$ and $\mathtt{V} \in \{\mathtt{I}, \mathtt{O}\}$, is a labeled tree $T(S)$ where:

- the labels on the nodes are formulae of the language,
- the root is labeled with ϕ_1,
- a node labeled with ϕ_i has as only child a node labeled with ϕ_{i+1} ($1 \leq i < n$),
- for any other node, the label is the result of the application of (In ...) or (Out ...) over the la-

[4]Note that the order is arbitrary and does not affect the outcome.

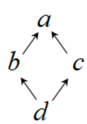
bel of some ancestor node that has not been expanded before.

The use of the meta-rules (Closure), (Cycle...) and (Replay...) can extend this notion of tableaux in the expected way. Analogously to the case of sentences, we can define:

DEFINITION **3.6** A set of formulae S is

- *satisfiable (in AF)* iff a tableau $T(S)$ for S is open, and
- *valid (in AF)* iff a tableau $T(S)$ for S is grounded.

EXAMPLE **3.7** Let $AF = \langle\{a, b, c, d\}, \{(b, a), (c, a), (d, b), (d, c)\}\rangle$. Assume we want to test if $\{a^{\mathtt{I}}, d^{\mathtt{I}}\}$ is grounded. Figure 9 shows the tableau, where we can see that the only branch is grounded, hence the set is grounded (and satisfiable).



AF

1. a^{I}	
2. d^{I}	
3. b^{O}	(In a, 1)
4. c^{O}	(In a, 1)
5. –	(In d, 2)
6. d^{I}	(Out b, 3)
7. –	(Replay d^{I}, 2, 6)
8. d^{I}	(Out c, 4)
9. –	(Replay d^{I}, 2, 8)

Figure 9: $\{a^{\mathtt{I}}, d^{\mathtt{I}}\}$ is grounded.

In the next section we deal with tableaux for individual arguments in order to check acceptance in preferred semantics. The notion of tableaux for sets of arguments will be useful for proving skeptical acceptance in preferred semantics.

4 Tableaux criteria applied to preferred semantics

An argument is *credulously* justified with respect to a given semantics when it belongs to some of the extensions sanctioned by that semantics. The *skeptical* justification, instead, is established when the argument belongs to each one of the sanctioned extensions. We prove next that $x^{\mathtt{I}}$ is satisfiable if, and only if, x is credulously justified with respect to preferred semantics.

We begin by establishing some facts. First, we prove that the set of all arguments marked with I in an open branch form an admissible set.

DEFINITION **4.1** Let B be a branch of a given tableau. We define $B^{\mathtt{I}} =_{def} \{x : x^{\mathtt{I}} \text{ occurs in } B\}$ and $B^{\mathtt{O}} =_{def} \{x : x^{\mathtt{O}} \text{ occurs in } B\}$.

LEMMA **4.1** Let $T(x^{\mathtt{V}})$ be the tableau of a formula $x^{\mathtt{V}}$. For every open branch B of $T(x^{\mathtt{V}})$, $B^{\mathtt{I}}$ is admissible.

Proof. Let B be an open branch of $T(x^{\mathtt{V}})$. Observe that B is open iff $B^{\mathtt{I}} \cap B^{\mathtt{O}} = \emptyset$.
(i) $B^{\mathtt{I}}$ is conflict free. For, let $u, v \in B^{\mathtt{I}}$. That is, $u^{\mathtt{I}}$ and $v^{\mathtt{I}}$ occur in B. Suppose that $(u, v) \in R$. Then $u \in B^{\mathtt{O}}$ and $u \in B^{\mathtt{I}} \cap B^{\mathtt{O}}$, contradicting the fact that $B^{\mathtt{I}} \cap B^{\mathtt{O}} = \emptyset$.
(ii) $B^{\mathtt{I}}$ accepts all its elements. For, let $v \in B^{\mathtt{I}}$ and u be such that $(u, v) \in R$. Then, $u \in B^{\mathtt{O}}$. As B is open, there is w in $B^{\mathtt{I}}$ such that $(w, u) \in R$ (otherwise, B would be closed). Thus, every element of $B^{\mathtt{I}}$ is acceptable w.r.t. $B^{\mathtt{I}}$.
From (i) and (ii), $B^{\mathtt{I}}$ is admissible.□

An important notion will be that of a "minimal defense" of an argument.

DEFINITION **4.2** Let $\langle A, R\rangle$ be an argumentation framework. For every argument $x \in A$ and for every subset

$S \subseteq A$, we say that S is a *minimal defense of* x iff S is a minimal (w.r.t. $\subseteq$) admissible set such that x is acceptable w.r.t. S.

LEMMA **4.2** Let $\langle A, R \rangle$ be an argumentation framework, and let $S \subseteq A$ an admissible subset of arguments. If $x \in S$, then $T(x^{\mathtt{I}})$ has an open branch.

Proof. To see that $T(x^{\mathtt{I}})$ has an open branch, let us consider any subset $S' \subseteq S$ which is a minimal defense of x (since S is admissible, such a minimal defense exists). Hence, in $T(x^{\mathtt{I}})$ we will have that, for every $y \in Attackers(S')$, $y^{\mathtt{O}}$ can be expanded using (Out $y^{\mathtt{O}}$) to yield a sentence $w^{\mathtt{I}}$ such that $w \in S'$. Moreover, since S' is admissible, it is conflict-free, implying that the branch produced by that procedure will not yield a closure. Hence, in that way we can only obtain an open branch.□

THEOREM **4.1** *An argument x is credulously justified w.r.t. the preferred semantics iff $x^{\mathtt{I}}$ is satisfiable.*

Proof. We show that $x^{\mathtt{I}}$ is satisfiable iff x belongs to some preferred extension.
(If) Suppose x belongs to a preferred extension E. Since E is admissible, by Lemma 4.2 we have that some branch of $T(x^{\mathtt{I}})$ is open, since E contains some minimal defense of x. Thus, $x^{\mathtt{I}}$ is satisfiable.
(Only if) If $x^{\mathtt{I}}$ is satisfiable then its tableau has an open branch B. By Lemma 4.1, $B^{\mathtt{I}}$ is admissible. This implies that x belongs to an admissible set. Every admissible set is a subset of a preferred extension. Thus, x belongs to a preferred extension.□

Now we turn to the problem of how to use the tableaux method for proving preferred skeptical justifications. First, note the following

OBSERVATION **4.1** *If x is skeptically justified w.r.t. the preferred semantics, then $x^{\mathtt{O}}$ is not satisfiable.*

Proof. By contraposition: we show that if $x^{\mathtt{O}}$ is satisfiable then there exists some preferred extension to which x does not belong. If $x^{\mathtt{O}}$ is satisfiable then there exists some open branch B in its tableau. Since the branch is open, by Lemma 4.1, $B^{\mathtt{I}}$ is an admissible set. Then, there must be a preferred extension E such that $B^{\mathtt{I}} \subseteq E$. But $B^{\mathtt{I}}$ includes an argument y such that y attacks x. Clearly, $y^{\mathtt{I}}$ is introduced in the tableau by the application of (Out x). Now, since $B^{\mathtt{I}} \cup \{x\}$ is not conflict-free we have that $B^{\mathtt{I}} \subseteq E$ and $B^{\mathtt{I}} \cup \{x\} \not\subseteq E$. Therefore x does not belong to E.□

Hence, if $T(x^{\mathtt{O}})$ is open, then we have a prove that x is not skeptically justified w.r.t. preferred semantics. But if it is not open we cannot assert the contrary. The reason is that x can still be outside of some preferred extension (see the case of a in Example 4.1). On the other hand, when every preferred extension is stable the condition is also sufficient (i.e. when the argumentation framework is *coherent* [23]).

More importantly, we can still establish a necessary and sufficient condition for preferred skeptical justification, even for non coherent frameworks. In order to find if a given argument x belongs to every preferred extension we first want to know if $x^{\mathtt{I}}$ is valid. If it is, then the answer is $y$es (since the grounded extension is always included in every preferred extension); otherwise, we want to know if it is satisfiable. If $x^{\mathtt{I}}$ is not satisfiable, then the answer is *no*; otherwise, we need to keep searching. We first try to prove that x does not belong to some preferred extension. The following result will help us to develop a strategy.

LEMMA **4.3** Let E be a preferred extension of $\langle A, R\rangle$. For every argument $x \in A$ and every minimal defense D of x, $E \cup D$ is not conflict-free iff $x \notin E$.

Proof. (If) By contraposition, assume that there exists a minimal defense D of x such that $E \cup D$ is conflict-free. Then, $E \cup D$ is clearly admissible, because both sets are admissible. Now, since by definition of preferred extension, E is maximally (w.r.t. $\subseteq$) admissible, it follows that $D \subseteq E$. And since x is acceptable w.r.t. D, then x is acceptable w.r.t. E. Therefore, $x \in E$.

(Only if) Assume $E \cup D$ is not conflict-free, for every minimal defense D of x. Since E is admissible, E attacks every such D. By the absurd, assume that $x \in E$. This implies that there exists some minimal defense D of x such that $D \subseteq E$. In turn, since E attacks D, we have that E is not conflict-free and, then, E is not admissible. This contradicts that E is a preferred extension. Therefore, $x \notin E$.$\square$

This result implies that if x is not skeptically justified, then there exists some preferred extension which is in conflict with *every* minimal defense of x. So our strategy consists in finding some admissible set of arguments in conflict with all the minimal defenses of x. If such a set is found then, since there exists some preferred extension containing it, there also exists a preferred extension not inlcuding x.[5] To illustrate the strategy, consider the argumentation framework of Figure 10[6], left. a is not skeptically justified since it does not belong to the preferred extension $E = \{e, f\}$. There exist two minimal defenses of a, $\{c\}$ and $\{d\}$, and E is in conflict with both of them. How can we relate that fact to what we observe in the tableau (Figure 10, right)? First, both $e^{\mathtt{O}}$ and $f^{\mathtt{O}}$ occur each in an open branch. Each open branch makes reference to a minimal defense of x, so e and f attack minimal defenses of a. Second, $e^{\mathtt{I}}$ and $f^{\mathtt{I}}$ are clearly satisfiable (their open tableaux can be trivially constructed given the cycles $(e, c), (c, e) \in R$ and $(f, d), (d, f) \in R$,

[5]This strategy is similar to that followed in the algorithm recently proposed in [35].

[6]This example was provided by a reviewer.

respectively). Third, $\{e^{\mathtt{I}}, f^{\mathtt{I}}\}$ can be proved to be satisfiable, implying that $\{e, f\}$ is included in an admissible set S. Clearly, x cannot belong to S, nor to any admissible set containing S (particularly, a preferred extension). Hence, we can conclude that there exists some preferred extension to which a does not belong.

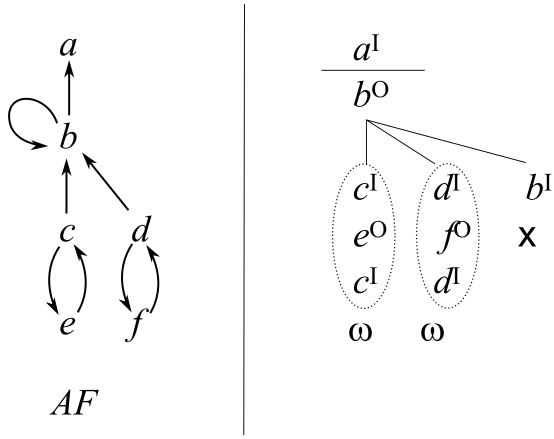


Figure 10: a does not belong to the preferred extension $\{e, f\}$.

To state the general result, we first introduce some more auxiliary notions and claims.

LEMMA **4.4** Let $T(x^{\mathtt{I}})$ be a tableau. Then: (i) for every open branch B of $T(x^{\mathtt{I}})$ there exists a minimal defense D of x such that $D \subseteq B^{\mathtt{I}}$, and (ii) for each minimal defense D of x, there exists some open branch B of $T(x^{\mathtt{I}})$ such that $D \subseteq B^{\mathtt{I}}$.

Proof. (i) Assume B is an open branch. By Lemma 4.1, $B^{\mathtt{I}}$ is admissible. Moreover, x is acceptable w.r.t. $B^{\mathtt{I}}$. Then, it is obvious that $D \subseteq B^{\mathtt{I}}$ for some minimal defense D of x. (ii) Assume that D is a minimal defense of x. It is clear that we can choose elements only from D to obtain $B^{\mathtt{I}}$, for an open branch B. But then we need to choose *every* element in D, since D is minimal. Hence, $D \subseteq B^{\mathtt{I}}$. □

DEFINITION **4.3** Let $AF = \langle A, R \rangle$ and $x \in A$. For every set of arguments $S \subseteq A$, we say that S *blocks* (or is a *blocker of*)[7] x iff S is admissible and, for every minimal defense D of x, S attacks D.

LEMMA **4.5** Let $AF = \langle A, R \rangle$ and $x \in A$. Then, $S \subseteq A$ is a blocker of x iff for every open branch B of the tableau $T(x^{\mathtt{I}})$, $B^{\mathtt{O}} \cap S \neq \emptyset$.

Proof. Immediate from Definition 4.3 and Lemma 4.4.□

Given this fact, and by a little abuse of language, we will say that $S^{\mathtt{O}} = \{y^{\mathtt{O}} : y \in S\}$ is a blocker of $x^{\mathtt{I}}$ iff S is a blocker of x.

THEOREM **4.2** *x is skeptically justified w.r.t. preferred semantics iff*

1. *$x^{\mathtt{I}}$ is satisfiable, and*
2. *$x^{\mathtt{I}}$ has no blockers.*

Proof. (If) Assume that $x^{\mathtt{I}}$ is satisfiable and x does not belong to some preferred extension E. We prove that $x^{\mathtt{I}}$ has a blocker. By Lemma 4.3, E attacks every minimal defense D of x. Let $S \subseteq E$ be an admissible set that contains all the arguments y in E such that y attacks D, for some minimal defense D of x (note that, since E is maximally admissible, such a set exists). By Lemma 4.4, it follows that in every open branch B in the tableau of $x^{\mathtt{I}}$, occurs some sentence $y^{\mathtt{O}}$ such that $y \in S$. Then, clearly, S is a blocker of x, i.e., $S^{\mathtt{O}}$ is a blocker of $x^{\mathtt{I}}$.

(Only if) By contraposition. Let $x^{\mathtt{I}}$ be satisfiable and suppose that $S^{\mathtt{O}}$ is a blocker of $x^{\mathtt{I}}$. Then, by Lemma 4.5, in each open branch B in the tableau of $x^{\mathtt{I}}$ occurs some sentence $y^{\mathtt{O}} \in S^{\mathtt{O}}$. Hence, $S = \{y : y^{\mathtt{O}} \in S^{\mathtt{O}}\}$ attacks every minimal defense of x. And since

[7] We borrow the terms 'blocks' and 'blocker' from [28].

S is admissible and every admissible set is contained in some preferred extension, there exists a preferred extension E such that $S \subseteq E$. Then, E attacks every minimal defense of x. By Lemma 4.3, it clearly follows that $x \notin E$.□

EXAMPLE **4.1** [37] See AF in Figure 11: a is not skeptically justified. There are two preferred extensions: $\{c, a\}$ and $\{d\}$ (no argument is skeptically justified w.r.t. preferred semantics). $d^{\mathtt{O}}$ occurs in the only open branch and $S^{\mathtt{I}} = \{d^{\mathtt{I}}\}$ is satisfiable, which implies that $S = \{d\}$ is included in some admissible set. Hence, $S^{\mathtt{O}}$ is a blocker of $x^{\mathtt{I}}$, which proves that d belongs to a preferred extension to which a does not belong.

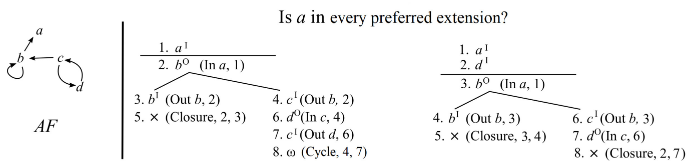


Figure 11: a is not skeptically justified. $\{d\}$ blocks a.

EXAMPLE **4.2** (Figure 12). AF has four preferred extensions, $\{d, e, a\}$, $\{d, f, a\}$, $\{c, e, a\}$, and $\{c, f, a\}$, and we want to prove that a belongs to all of them. Given the tableau for $a^{\mathtt{I}}$, we only have one "out" argument in each open branch: d, e, and f. So, we only have to prove that $\{d^{\mathtt{I}}, e^{\mathtt{I}}, f^{\mathtt{I}}\}$ is not satisfiable. That is easy to see, given that e "refutes" f in B_2 and f "refutes" e in B_3, meaning that e and f cannot stand together.

5 Tableaux criteria applied to grounded semantics

Dung's grounded semantics models a skeptical behavior which does not coincide in general with skeptical

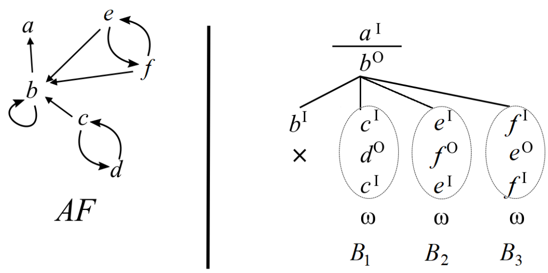


Figure 12: a is skeptically justified w.r.t. preferred semantics.

preferred semantics: while the grounded extension is always included in the intersection of all the preferred extensions, the converse is not always the case.

To prove the correspondence between grounded semantics and validity, let us first consider the following result.

Lemma **5.1** Let $S \subseteq A$ be a set of arguments such that for every $x \in S$, $x^{\mathtt{I}}$ is valid, and let $z \in A$ be acceptable w.r.t. S. Then $z^{\mathtt{I}}$ is valid.

Proof. Assume that, for every $x \in S$, $x^{\mathtt{I}}$ is valid, and z is acceptable w.r.t. S. Let $S' = \{y : (y, z) \in R\}$. By definition, if z is acceptable w.r.t. S then, for every argument $y \in S'$, S attacks y. Now, since $x^{\mathtt{I}}$ is valid for every $x \in S$ and S attacks every $y \in S'$, it follows that, for every $y \in S'$, $y^{\mathtt{O}}$ is valid too. Therefore, $z^{\mathtt{I}}$ is valid.□

Lemma **5.2** Given $\langle A, R \rangle$ and $x \in A$, let $T(x)$ be the tableau of x. For every branch B of $T(x)$, let $B^{args} = B^{\mathtt{I}} \cup B^{\mathtt{O}}$. If B is grounded, then the attack relation among the arguments in B^{args}, i.e., $R \restriction_{B^{args}}$, has no cycles.

Proof. By contraposition, assume there exists a cycle in $R \restriction_{B^{args}}$. Then, ω should occur in B by application of the Cycle rule. But then, by definition, B is not grounded. □

THEOREM **5.1** *An argument x is justified w.r.t. the grounded semantics iff $x^{\mathtt{I}}$ is valid.*

Proof. (If) Assume $x^{\mathtt{I}}$ is valid. Then the tableau of $x^{\mathtt{I}}$ has a branch that is grounded. Then, by Lemma 5.2, that branch has no cycles. Let $x_1^{\mathtt{I}}, \ldots, x_n^{\mathtt{I}}$ the sequence of all the formulae labeled with I in the order in which they occur (without repetitions) in the branch, where $x_1 = x$. Since the argument in the leaf node, x_n, has no attackers, we have that x_n is acceptable w.r.t. the empty set of arguments, i.e., $x_n \in F(\emptyset)$. Then we will have $x_{n-1} \in F(\emptyset)$, if it has no attackers, or $x_{n-1} \in F^2(\emptyset) = F(F(\emptyset))$, if it has some attacker (i.e., $x_{n-1} \in F(\emptyset) \cup F^2(\emptyset) = F^2(\emptyset)$, due to the monotonicity of $F(.)$ –see [23]). Then, we inductively have $x_{n-i} \in F^{i+1}(\emptyset)$, $0 \leq i < n$, getting $x_1 \in F^n(\emptyset)$, where $F^n(\emptyset)$ is the least fixed point of $F(.)$, that is, the grounded extension of the framework (Example 5.1 illustrates the inductive process).
(Only if) It suffices to show that for every n and for every x, if $x \in F^n(\emptyset)$ then x is valid. We prove that by induction on n. For $n = 1$ the result is obvious, since every argument in $F(\emptyset)$ is free of attackers. Let now $F^k(\emptyset)$ be such that x is valid, for every $x \in F^k(\emptyset)$. Since for every $x \in F^{k+1}(\emptyset)$, x is acceptable w.r.t. $F^k(\emptyset)$, by Lemma 5.1 we have that every argument in $x \in F^{k+1}(\emptyset)$ is valid. Therefore, for every n and every argument $x \in F^n(\emptyset)$, x is valid.□

EXAMPLE **5.1** Figure 13 shows that the I labeled arguments in a grounded branch correspond to successive applications of the operator $F(.)$. $F(\emptyset) = \{c, e, g\}$ is the set of non-attacked arguments, corresponding to the sentences $c^{\mathtt{I}}$, $e^{\mathtt{I}}$, and $g^{\mathtt{I}}$, which yield – (dash). Then, $F(\{c, e, g\}) = \{c, e, g, a\}$ can be found in the

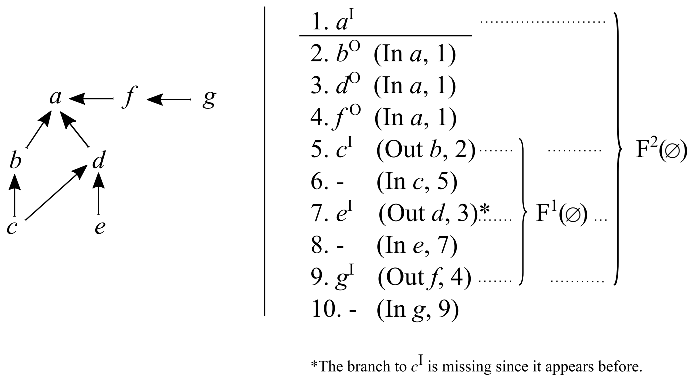


Figure 13: Example 5.1. For simplicity, we pruned the branch from $d^{\mathtt{O}}$ in 3 to $c^{\mathtt{I}}$, since it would yield the same outcome as in line 6.

tableau by tracing back the immediate ancestors of $c^{\mathtt{I}}$, $e^{\mathtt{I}}$, and $g^{\mathtt{I}}$ that are labeled with $\mathtt{I}$, i.e., just $a^{\mathtt{I}}$ in this case. Since there are no more sentences to trace back, $\{c, e, g, a\}$ is the desired least fixed point of $F(.)$.

6 Related work

The present work is related to various proof-theoretic and algorithmic approaches to argumentation frameworks.[8] One salient difference between the present approach and others is that our tableaux method allows us to check whether an argument is *out* directly, and not as a consequence of checking whether it is *in*. Dialogue based approaches, for instance [37, 14, 22], always begin with a proponent advancing an argument as a thesis to be defended, but not refused. The labeled dialectical trees of Chesñevar et al. [17, 16], as our method, avoid the mention to players, but while our tableaux proceed by labeling arguments from the root

[8]We refer the reader to the *Handbook of Formal Argumentation* [2] for various approaches, some of which we mention here.

to the leaves, dialectical trees proceed the other way round, starting by labeling the non attacked arguments as undefeated (*in*) and going bottom-up through the attack lines to the root. In this way, while dialectical trees proceed by labeling the entire tree, in our method the labeling goes backwards starting at the focused argument. The treatment of skeptical preferred acceptance given through our approach is original and can be confronted with that of dialogue games. In [21], the authors define a meta-level dialogue to find a possible preferred extension conflicting with every admissible set containing argument candidate to skeptical acceptance. In the approach of [28], the players construct and exchange entire admissible extensions rather than single arguments. [33] determine skeptical preferred acceptance by defining a two-phase game with different legal moves and protocols for each phase: the first phase allows the opponent to identify a labeling where the focal argument is not justified, and in the second phase the proponent tries to show that the previous labeling is not preferred. Lately, [35] offered an algorithm to compute skeptical preferred acceptance that, though not based on dialogue games or dialectical trees, exploits the same fact expressed in Theorem 4.2 (cf. Theorem 11 in [35]). The algorithm tries to construct an admissible set that attacks some admissible set containing the focus argument x. If such a set cannot be found, then x is skeptically justified. Otherwise, the procedure continues trying to expand the set to include x. If this cannot be done, then the process ends with a negative result. Otherwise, it continues trying with other sets in the same manner. On the other hand, our tableaux do not proceed by constructing admissible sets, but by checking membership or inclusion in some admissible set.

Dung and colleagues' *dispute trees* [22] are maybe closer to our tableaux than other dialogue approaches.

Dispute trees always begin and end with moves made by the proponent, so disputes that lead the proponent to lose are excluded by definition. The difference between finite or infinite disputes make the difference between groundedness and simple admissibility. Several derivations of dispute trees for assumption-based argumentation were also developed by Francesca Toni and colleagues [19, 25, 36, 20].

Much of the work of Martin Caminada is also close to our approach. Caminada's Socratic interpretation of discussions on preferred semantics [12] revolves around the possibility of labeling with `in` a given argument, which implies labeling with `out` its attackers, and so on. Unlike our approach, admissibility games always begin with an "in" move. Game trees yield branches both for "in" and "out" moves, which leads the author to define a pruning method for different levels of a tree as a way of finding the winning strategies of the proponent. In our approach, instead, branching is only a consequence of "out" hypothesis (corresponding to disjunctions), and the final result is visually immediate once the tableau is built. In [9, 10], the authors study strong admissibility. Originally defined in [3], it refers to the capability of a set of arguments of defending all its arguments, directly or indirectly, with non attacked arguments (i.e. no defense is cyclical). Caminada et al. define a *grounded discussion game* where four kinds of moves are defined, one for the proponent to advance a claim, and three for the opponent to advance a counterargument, concede and retract. The burden of proof is on the proponent, who is unable to defend in circles, and the opponent only has to cast sufficient doubts. Using the so called *admbuster* example [8] (Example 3.5 above), the authors show the better efficiency of the grounded game in comparison with other standard games. As we have seen, in our tableaux method the (Replay...) rule does a similar job avoiding the repetition of already computed threads. Moreover, the validity of $x^{\mathtt{I}}$ proves that x belongs to a strongly admissible set of arguments, and for every grounded branch B, $B^{\mathtt{I}}$ is strongly admissible. We can also check whether or not any non-empty set $H \subseteq A$ of arguments is weakly admissible (i.e. simply admissible) or strongly admissible. Let $S{=}\{x^{\mathtt{I}} : x \in H\}$. Then, H is weakly admissible if S is satisfiable, and is strongly admissible if S is valid. In consequence, if S is satisfiable then it is included in some preferred extension, and if S is valid then it is included in the grounded extension and, hence, in every preferred extension.

Caminada and colleagues [11] also proposed a translation from logic programs to argumentation frameworks to establish equivalences between logic programming semantics and argumentation semantics. In that way, the authors consider the possibility of applying the dialectical proof procedures for argumentation in the context of logic programming, to determine the status of a single argument without having to construct an entire logic programming model. Other authors also consider translations of argumentation frameworks to logic programs. For instance, [32] define logic programs using special predicates `in`, `out` and `und` (with the obvious meaning) and specific clauses to represent different extension semantics. In this line, an interesting subject to investigate in the future is the equivalence of our tableaux method with standard deduction methods in logic programming like resolution, via adequate translations.

We argued that our method, similar in nature to semantic tableaux, can serve as a friendly introduction to the semantics of argumentation frameworks for the logical-philosophical audience. In that vein, we can mention the work of [1], who proposed a sequent-based proof-theoretic formalism, exploiting the idea that arguments can be seen as Gentzen-style sequents, and attacks as sequent elimination rules. An important antecedent in the logical treatment of argumentation frameworks is the work of Besnard and Hunter [4, 5]. In their approach, arguments are not taken as abstract but as logically structured entities defined on a logical language, and a non-primitive notion of undercut takes the place of the attack relation. Then, an argument tree describes the various ways an argument can be counter-argued, as well as how counter-arguments can be counter-argued, and so on recursively. The authors define 'argument structures' for a formula α collecting all the argument trees supporting α and all the argument trees supporting $\neg\alpha$. In this way, argument structures can represent Dung's argumentation frameworks. However, the specific conditions on undercuts impose constraints on the attack relation in such a way that not every Dung's argumentation framework can be matched with an argument structure. Clearly, this limitation implies also a difference between Besnard and Hunter's approach and ours. Another interesting logical turn is given by Grossi [26], who uses a second-order modal logic language with which all the Dung-style semantic notions can be expressed. The strategy is that argumentation frameworks can be viewed as Kripke frames where arguments are modal states and the accessibility relation is obtained up from the attack relation. Moreover, the author introduces a two-player (proponent and opponent) model-checking game for verifying whether a formula is satisfied in a given structure. In particular, a game for skeptical preferred semantics is defined. Interestingly, one can follow the game tree with an argument-theoretic reading: the claim that argument a is in every preferred extension can be challenged by pointing a candidate preferred extension p to which a does not belong, which in turn can be challenged by the claim that p is not a preferred extension or that a belongs to p, and so on. The author highlights the different structure of the game compared with "standard argument games" (a sort to which our tableaux belongs) where nodes consist only of arguments, and which are played by selecting, at each node, only arguments that attack the current node.

Finally, among other applications of semantic tableaux methods to argumentation we can mention [31], but in this case tableaux are used to prove a conclusion via refutation through structured arguments for both deductive and defeasible reasoning.

7 Conclusion

We have proposed a tableaux method for argumentation frameworks that enables to decide, for any argument x, whether an assignment of a value in/out is satisfiable or not (i.e. whether or not there exists a labeling which assigns that value), and whether it is valid or not (i.e. whether or not it can *only* be assigned that value). We have shown that this method can be used for deciding credulous and skeptical justifications with respect to preferred and grounded semantics (this comprises the subsidiary problems of determining admissibility and complete justifications).

Unlike dialogue games, which are useful for proving whether an argument is justified or not, the tableaux method is also useful for showing whether a rejection of an argument is justified or not. This is because a dialogue game always begins with Pro advancing an argument she wants to show that is in, and proves the rejection as a failure to prove the acceptance; to our knowledge, there exists no other approach in which Pro can advance an argument she wants to show that is out. Our method, instead, enables to represent both situations. Moreover, it can easily be applied to test the acceptability of sets of arguments (introducing lists of the form $x_1^{\mathtt{I}}, \ldots, x_n^{\mathtt{I}}$) and even the joint consistency of acceptances and rejections (lists of the form $x_1^{\mathtt{I}}, \ldots, x_i^{\mathtt{I}}, x_{i+1}^{\mathtt{O}}, \ldots, x_n^{\mathtt{O}}$). We have also solved a usual problem of dialog-based procedures, which present some anomalies in the treatment of certain cases such as the *admbuster* case (Example 3.5), first discussed in [8]. For this we have considered the ancestors of each line in the tableaux for the application of the replay rule to avoid irrelevant computations.

Our method shows similar drawbacks as the other mentioned theories with respect to Dung's stable semantics (which extensions are admissible sets attacking all the external arguments). In coherent argumentation frameworks, where there exist no odd-length cycles of attack, stable semantics coincides with preferred semantics, hence the tableaux proofs for stable justifications for individual arguments are also limited to coherent argumentation frameworks.[9] But this would not be a problem for stable extensions, since the tableaux for a set of arguments sanctioned as preferred extensions will show all the arguments labeled with $\mathtt{O}$ not belonging to the set. Hence, to check that a preferred extension is also stable we have to check that all the arguments in the framework are computed in the tableau.

Another limitation of our method, like current dialogue or dispute games, is that it does not provide an efficient procedure to compute whole extensions. That would require an entire series of tableaux to be produced, implying a lot of work. The grounded extension can be found by checking validity for each argument

[9]The same can be said about *symmetric argumentation frameworks* [18], i.e., where R is symmetric, irreflexive, and nonempty, since they are always coherent.

in the framework, but this is also not efficient from a computational point of view. On the other hand, though we cannot *find* extensions efficiently, we are able to *check* given sets of arguments for satisfiability and validity. This makes our method more closely related to real-life dialogues, where one or several theses are advanced in order to justify them, but not to find *all* that one can defend.

Finally, the tableaux method that we have proposed is intuitive, it resembles analytical tableaux that are familiar to logic students, and for this reason we believe that it can be a useful tool for teaching semantics of argumentation frameworks.

References

[1] Arieli, O., and Strasser, C. Sequent-Based Logical Argumentation. *Argument and Computation 6*, 1 (2015), 73–99.

[2] Baroni, P., Gabbay, D., Giacomin, M., and van der Torre, L., Eds. *Handbook of Formal Argumentation*. College Publications, 2018.

[3] Baroni, P., and Giacomin, M. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence 171*, 10 (July 2007), 675–700.

[4] Besnard, P., and Hunter, A. A logic-based theory of deductive argumentation. *Artificial Intelligence 128*, 1-2 (May 2001), 203–235.

[5] Besnard, P., and Hunter, A. Argumentation based on classical logic. In *Argumentation in artificial intelligence*, I. Rahwan and G. Simari, Eds. Springer, 2009, pp. 133–152.

[6] Beth, E. W. Semantic entailment and formal derivability. *Mededelingen van de Konin-*

klijke Nederlandse Akademie van Wetenschappen, Afdeling Letterkunde 18, 13 (1955), 309–342.

[7] Caminada, M. On the issue of reinstatement in argumentation. In *Logics in Artificial Intelligence* (Berlin, Heidelberg, 2006), M. Fisher, W. van der Hoek, B. Konev, and A. Lisitsa, Eds., Springer Berlin Heidelberg, pp. 111–123.

[8] Caminada, M. Strong admissibility revisited. In *Computational Models of Argument* (2014), S. Parsons, N. Oren, C. Reed, and F. Cerutti, Eds., Frontiers in Artificial Intelligence and Applications, IOS Press, pp. 197–208.

[9] Caminada, M. A Discussion Game for Grounded Semantics. In *Theory and Applications of Formal Argumentation* (Cham, 2015), E. Black, S. Modgil, and N. Oren, Eds., Springer International Publishing, pp. 59–73.

[10] Caminada, M., and Dunne, P. Strong admissibility revisited: Theory and applications. *Argument & Computation 10*, 3 (2019), 277–300. Publisher: IOS Press.

[11] Caminada, M., Sá, S., Alcântara, J., and Dvořák, W. On the equivalence between logic programming semantics and argumentation semantics. *International Journal of Approximate Reasoning 58* (2015), 87–111. Special Issue of the Twelfth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2013).

[12] Caminada, M. W. A., Dvořák, W., and Vesic, S. Preferred semantics as socratic discussion. *Journal of Logic and Computation 26*, 4 (02 2014), 1257–1292.

[13] Cayrol, C., Doutre, S., and Mengin, J. Dialectical proof theories for the credulous preferred semantics of argumentation frameworks. In *Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*. Berlin: Springer, 2001, pp. 668–679.

[14] Cayrol, C., Doutre, S., and Mengin, J. On Decision Problems Related to the Preferred Semantics for Argumentation Frameworks. *Journal of Logic and Computation 13*, 3 (06 2003), 377–403.

[15] Cerutti, F., Gaggl, S., Thimm, M., and Wallner, J. Foundations of implementations for formal argumentation. In Baroni et al. [2], pp. 689–768.

[16] Chesñevar, C. I., and Simari, G. R. Modelling Inference in Argumentation through Labelled Deduction: Formalization and Logical Properties. *Logica Universalis 1*, 1 (Jan. 2007), 93–124.

[17] Chesñevar, C., Simari, G., and Godo, L. Computing dialectical trees efficiently in possibilistic defeasible logic programming. In *Proceedings of the 8th International LPNMR Conference* (2005), Springer, pp. 158–171.

[18] Coste-Marquis, S., Devred, C., and Marquis, P. Symmetric argumentation frameworks. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (Berlin, Heidelberg, 2005), L. Godo, Ed., Springer Berlin Heidelberg, pp. 317–328.

[19] Craven, R., and Toni, F. Argument graphs and assumption-based argumentation. *Artificial Intelligence 233* (2016), 1–59.

[20] Cyras, K., Fan, X., Schulz, C., and Toni, F. Assumption-based argumentation: Disputes, explanations, preferences. *FLAP 4*, 8 (2017).

[21] Doutre, S., and Mengin, J. On sceptical versus credulous acceptance for abstract argument systems. In *Logics in Artificial Intelligence* (Berlin, Heidelberg, 2004), J. J. Alferes and J. Leite, Eds., Springer Berlin Heidelberg, pp. 462–473.

[22] Dung, P., Kowalski, R., and Toni, F. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence 170*, 2 (2006), 114–159.

[23] Dung, P. M. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence 77*, 2 (1995), 321–357.

[24] Dvořák, W., and Dunne, P. Computational problems in formal argumentation and their complexity. In Baroni et al. [2], pp. 631–688.

[25] Fan, X., and Toni, F. A general framework for sound assumption-based argumentation dialogues. *Artificial Intelligence 216* (2014), 20–54.

[26] Grossi, D. Abstract argument games via modal logic. *Synthese 190* (2012), 5–29.

[27] Jakobovits, H., and Vermeir, D. Robust semantics for argumentation frameworks. *Journal of Logic and Computation 9*, 2 (04 1999), 215–261.

[28] Modgil, S., and Caminada, M. Proof theories and algorithms for abstract argumentation frameworks. In *Argumentation in Artificial Intelligence*, I. Rahwan and G. Simari, Eds. Springer Berlin Heidelberg, 2009, pp. 105–129.

[29] Nofal, S., Atkinson, K., and Dunne, P. Algorithms for decision problems in argument systems under preferred semantics. *Artificial Intelligence 207*, 1 (2014), 23–51.

[30] Pollock, J. *Knowledge and Justification.* Princeton University Press, Apr. 2016.

[31] Roos, N. A Semantic Tableau Method for Argument Construction. In *Artificial Intelligence and Machine Learning* (Cham, 2021), M. Baratchi, L. Cao, W. A. Kosters, J. Lijffijt, J. N. van Rijn, and F. W. Takes, Eds., Springer International Publishing, pp. 122–140.

[32] Sakama, C., and Rienstra, T. Representing Argumentation Frameworks in Answer Set Programming. *Fundamenta Informaticae 155*, 3 (Jan. 2017), 261–292. Publisher: IOS Press.

[33] Shams, Z., and Oren, N. A two-phase dialogue game for skeptical preferred semantics. In *Logics in Artificial Intelligence* (Cham, 2016), L. Michael and A. Kakas, Eds., Springer International Publishing, pp. 570–576.

[34] Smullyan, R. *First-Order Logic.* Dover Publications, 1995.

[35] Thimm, M., Cerutti, F., and Vallati, M. Skeptical reasoning with preferred semantics in abstract argumentation without computing preferred extensions. In *Proceedings of the Thirtieth*

International Joint Conference on Artificial Intelligence, IJCAI-21 (8 2021), Z.-H. Zhou, Ed., International Joint Conferences on Artificial Intelligence Organization, pp. 2069–2075. Main Track.

[36] Toni, F. A generalised framework for dispute derivations in assumption-based argumentation. *Artificial Intelligence 195* (2013), 1–43.

[37] Vreeswijk, G., and Prakken, H. Credulous and sceptical argument games for preferred semantics. In *Proceedings of the JELIA 2000* (Berlin, Heidelberg, 2000), Springer, pp. 239–253.