A tableaux method for argumentation frameworks

Gustavo Adrián Bodanza
Departamento de Humanidades, Universidad Nacional del Sur and
IIESS, UNS-CONICET, Bahía Blanca, Argentina

ABSTRACT: In this work, a tableaux method is proposed as a decision procedure for argument justification in argumentation frameworks. This method enables to determine the satisfiability of sentences of the form 'argument a is in' and 'argument a is out', applying procedures similar to those of analytic tableaux for logics. This is a novelty with respect to dialogue games, which are designed only to decide if a given argument can be defended by a proponent. The method is proved to capture both, credulous and skeptical justifications for preferred semantics, and grounded semantics. Moreover, the method is defended as a useful tool for teaching on semantics for argumentation frameworks, given the familiarity of logicians with similar methods of analytic tableaux.

KEY WORDS: argumentation frameworks; tableaux methods; extension semántics; dialogue games

Un método de tablas analíticas para marcos argumentativos

RESUMEN: En este trabajo, proponemos un método de tablas analíticas para marcos argumentativos. Los marcos argumentativos están formados por un conjunto de argumentos y una relación de ataque entre éstos. El método permite decidir la justificación de proposiciones del tipo 'el argumento a es aceptado' y 'el argumento a es rechazado'. Esto es una novedad con respecto a juegos dialógicos, que son diseñados sólo para decidir is un argumento puede ser defendido por un proponente. Se demuestra que el método captura tanto las justificaciones crédulas y escépticas para la semántica preferida (preferred) como la semántica fundada (grounded). Argumentamos que el método constituye una herramienta útil para la enseñanza de semánticas de marcos argumentativos, dada la familiaridad de las tablas analíticas para la lógica.

PALABRAS CLAVES: marcos argumentativos; métodos de tableaux; semánticas de extensiones; juegos dialógicos

1 — Introduction

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 (1995) is a widely used tool for the representation of attack among arguments and the study of argument justification. Several semantics have been proposed in terms of this model. Extension semantics, as proposed by Dung and other authors, define subsets of arguments that can be defended together, according to some specific notions of defense. Another important semantics is that based on *labellings* (Jakobovits and Vermeir, 1999; Chesñevar and Simari, 2006, 2007; Caminada, 2006). Caminada (2006), for instance, defines a labelling as a function assigning to each argument a value *in*, *out*, or *undec* according to its relation of attack with the other arguments. The semantic view is useful to define acceptance and rejection of arguments, beyond the proof theory or decision method that can be implemented to determine whether a given argument is accepted or rejected. On the other hand, some proof theories have also been proposed, mainly modeled as dialogue games (Vreeswijk and Prakken, 2000; Cayrol, Doutre and Mengin, 2001). Basically, these models consider a game between two players, *Pro* and *Con*, where each player plays in turn and *Pro* plays first advancing an argument she wants to defend. In addition, Modgil and Caminada (2009) and Nofal, Atkinson and Dunne (2014) present algorithms to compute the status of arguments according to Dung's semantics.

In this paper we propose a logical representation of proofs for the acceptance/rejection of arguments. Our idea is inspired in the method of analytic tableaux for logics (Beth, 1955; Smullyan, 1998). This is a familiar decision method for logicians, and it is widely used for teaching because of its simplicity and intuitiveness. It consists of a proof procedure for deciding the satisfiability of a formula or a set of formulae, and proceeds by developing a tree (the tableau) through the application of rules that "expand" a formula according to the truth table of its principal connective and its valuation (true or false). In a finite 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 *a* is *in*' (*in* meaning 'acceptable') or 'argument *a* is *out*' (*out* meaning 'rejectable') are satisfiable. In essence, our tableaux method combines ideas from dialogue games and labelling semantics. From dialogue games -paradigmatically, TPI-disputes (Vreeswijk and Prakken, 2000)- 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 *in* or *out*. This last option introduces a novelty, since there dialogue always begin proposing an acceptance, not a rejection. From labelling semantics, on the other hand, we take the idea of labelling the arguments according to the attack relation among them. But we define labelling rules that are applied with respect to a query about the status, *in* or *out*, of a given argument. For example, given the query 'is argument *a out*?', the argument *a* is labeled with a sign for *out*, which stands for the initial hypothesis that '*a* is *out*' is defendible. This hypothesis is followed by the application of a rule representing the fact that if *a* is *out* then -according to the argumentation framework- some attacker of *a* must be *in*. This, in time, leads to develop one branch for each possible attacker of *a*, labelling each of them as *in*. The procedure continues in the same manner, labelling each possible attacker of each attacker of *a* as *out*, and so on. If at the end, that is, when no further rules can be applied, some of these branches remain "open" (i.e. without contradiction) then we say that '*a* is *out*' is satisfiable. This procedure has also reminiscenses of dialectical trees for argumentation frameworks (Chesñevar and Simari, 2006, 2007). But the tableaux proceed by labelling arguments from the root to the leaves, while dialectical trees proceed the other way round, starting by labelling 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 signing the entire graph of attacks, in our method the signs propagate backwards only from the focused argument ignoring unconnected arguments.

We define argumentation versions of the notions of satisfiability and validity with respect to an argumentation framework, and show how the tableaux method can be applied to the problem of determining credulous/skeptical acceptance/rejection for Dung's preferred and grounded semantics.

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. Section 4 shows soundness and completeness results for credulous and skeptical justification with respect to preferred semantics, while section 5 shows similar results for (skeptical) justification with respect to grounded semantics. Conclusions are exposed in section 6.

2 — Background: Dung's argumentation frameworks

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

Definition 1. An *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 only consider finite argumentation frameworks, i.e., frameworks in which A is finite.

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 1. Consider the arguments:
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 $A=\{a, b\}$ y $R=\{(b, a)\}$. This can also be diagrammed by a digraph: Figure 1.

Figure 1

1 Note that questions like "what kind of things arguments are?" or "under which conditions do attacks occur?" cannot be answered within the model. For this reason argumentation frameworks are said to be abstract.

Example 2. Consider the arguments:
a: *The government of X cannot negotiate with the government of Y because the government of Y 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 $A = \{a, b\}$ y $R = \{(a, b), (b, a)\}$ (Figure 2).

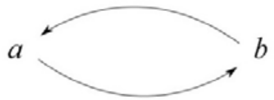

Figure 2

Intutitively, the acceptance of an argument depends on the way the arguments can be defended from attacks. The different ways to achieve this lead to corresponding notions of *justification* or *warrant*. A justification criterion can be captured by an *extension semantics* σ yielding, for every argumentation framework AF, a family os subsets of A, $\mathcal{E}_\sigma(AF) \subseteq 2A$, each one called an "extension" of AF (under σ). An argument is said "credulously justified" under a semantics σ if it belongs to some extension $E \in \mathcal{E}_\sigma(AF)$, and is said to be "skeptically justified" if it belongs to every extension $E \in \mathcal{E}_\sigma(AF)$. Dung defines grounded and preferred semantics as ways of capturing skeptical and credulous behaviours, respectively.

Definition 2. An argument a is *acceptable* w.r.t. a subset $S \subseteq A$ iff for every argument b such that $(b, a) \in R$, there exists some argument $c \in S$ such that $(c, b) \in R$. A set of arguments S is *admissible* if each $a \in S$ is acceptable w.r.t. S, and is conflict-free, i.e., the attack relation does not hold for any pair of arguments belonging to S. A *preferred extension* is any maximally (w.r.t. ⊆) admissible set of arguments of AF. A *complete extension* of AF is any conflict-free subset of arguments which is a fixed point of $F(\cdot)$, where $F(S) = \{a: a \text{ is acceptable w.r.t. } S\}$, while the grounded extension is the least (w.r.t. ⊆) complete extension.

In Example 1, $\{b\}$ is the only extension for all the above defined semantics, i.e. b is justified both skeptically and credulously under those semantics; in example 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 are credulously justified under preferred and complete semantics.

3 — Tableaux for argumentation frameworks

We will define the *tableaux scheme rules* of a generic argumentation framework $AF = \langle A, R\rangle$. We will consider a language L which formulae are of the form x^V, where $x \in A$ and $V \in \{I, O\}$ ('I' standing for *in* and 'O' standing for *out*). For every argument $x \in A$, let $Attackers(x) = \{y : (y,$

$x)\in R\}$ and let $\{y_1, \ldots, y_k\}$ be an arbitrary (but fix) enumeration of *Attackers(x)*. Then the idea is to define, for every argument $x\in A$, rules to deduce that the attackers $y_1, \ldots, y_k$ are all out if x is in, and that at least one of $y_1, \ldots, y_k$ is in if x is out.

(Scheme rules -Preliminary version)

$$\text{(In } x\text{)} \qquad \frac{x^{I}}{y1^{O} \ldots yk^{O}} \qquad\qquad \text{(Out } x\text{)} \qquad \frac{x^{O}}{y1^{I} \quad \ldots \quad yk^{I}}$$

The table is constructed by "expanding" each sentence of the form x^{I} or x^{O} by the application of the corresponding rule. The rule (In x) leads to a sequence of sentences, each one signing with 'O' an attacker y_{I} of x. This sequence must be interpreted as a conjunction. The rule (Out x) splits the proof in as many branches as attackers x has, each one signed with '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 (branches denote inclusively disjunct possibilities).

A tableau will proceed by applying these rules successively until no rule can be further applied. To make it orderly, we have to consider how to proceed in presence of sets of sentences. For instance, let $AF = \langle\{a, b, c, d, e\}, \{(b, a), (c, a), (d, b), (e, c)\}\rangle$ and assume we want to develop the tableau for a^{I}. Then we want to derive the string $b^{O}c^{O}$ (or equivalently, $c^{O}b^{O}$) applying the rule (In a) over a^{I}. Next we are going to expand the sequence $b^{O}c^{O}$ atom by atom from left to right, developing the first atom and adding the remaing string of atoms to the result[2]. In this way, we apply (Out b) on the leftmost sentence of the premise $b^{O}c^{O}$, i.e. b^{O}, yielding d^{I}, to which we add the remaining part c^{O} of the string for further development. In the fourth line we apply (In d) in a similar way on $d^{I}c^{O}$, but since d has no attackers the resulting sequence is just c^{O}. After obtaining e^{I} in the fifth line by the application of (Out c), we finish the tableau since (In e) would not produce any sentence:

a^{I}	
$b^{O}c^{O}$	(In a)
$d^{I}c^{O}$	(Out b)
c^{O}	(In d)
e^{I}	(Out c)

In order make more precise this way of handling the rules on sets of sentences, we recharacterize the scheme rules more formally as follows.

2 Note that while it is not important the order of the atoms from a logical point of view, it will be importantfrom a procedural point of view. For that reason we will often consider a set of sentences sentences as a string.

Definition 3. (Scheme rules -Final version) Given an argumentation framework AF = ⟨A, R⟩, for every argument $x \in A$ let *Attackers(x)* = $\{y : (y, x) \in R\}$ and let $\{y_1, \dots, y_k\}$ be an arbitrary (but fixed) enumeration of *Attackers(x)*. We define the scheme rules:

(In x)

$$\frac{x^{I}\ \phi}{y_1^{O} \dots y_k^{O}\ \phi}$$

(Out x)

$$\frac{x^{O}\ \phi}{y_1^{I}\ \phi \quad \dots \quad y_k^{I}\ \phi}$$

where ϕ is any (possibly empty) string of sentences[3].

The tableaux are formally defined as follows:

Definition 4. Given an argumentation framework AF = ⟨A, R⟩ and arguments $x_1, \dots, x_k \in A$, a *tableau* (*on* AF) for a sentence $\phi = x_1^{V1} \dots x_k^{VK}$ where $Vi \in \{I, O\}$ for every i, $1 \leq i \leq k$, is a labeled ordered tree[4] T where:

— labels are strings of sentences of L,
— the root of T is labeled with ϕ,
— for any node n, the label of n is a sentence ϕ' which is the result of the application of (In ...) or (Out ...) over the label of the parent node of n.

If *Attackers(x)* = ∅, then (In x) just yields ϕ; if, in addition, ϕ is empty then the rule should not be applied as it would produce nothing. On the other hand, if *Attackers(x)* = ∅ then (Out x) leads to an absurd: x cannot be *out* if it has no attackers. Then we define some extra checking (meta) rules for terminating the tableaux and adding some symbols that will help to interpret the result. If x has no attackers and ϕ is empty then we just stop applying (In x) on x^{I}. Regarding x^{O}, if x has no attackers we will introduce the symbol × —no matter if it is accompanied by other sentences— meaning that that branch is closed:

(Closure 1)

$$\frac{\phi x^{O}\ \phi'}{\times}$$

for any (possibly empty) strings ϕ and ϕ' of sentences, and where *Attackers(x)* = ∅

3 The sequence $x^{V}\phi$ is understood as the sentence conformed by the string of atoms resulting of the concatenation of x^{V} with ϕ in that order (similarly for ϕx^{V} , $\phi x^{V}\phi'$, etc.). While the order is logically irrelevant,it will be important for an ordered development of the tableau.

4 An ordered tree is a tree such that there exists a function Θ that assigns to each node x with more than one successor a sequence Θ(x) which contains no repetitions, and whose set of terms consists of all the successors of x (Smullyan, 1995).

Moreover, we have to consider other cases for closure:

(Closure 2)

$$\frac{\begin{array}{c}\phi x^{\mathrm{V}}\ \phi' \\[1em] \psi x^{\mathrm{V}}\ \psi'\end{array}}{\times}$$

for any (possibly empty) strings of sentences ϕ, ϕ', ψ, and ψ', and where V∈{I, O} and -V∈{I, O}\{V}

This rule closes the branch when the same argument x appears signed with I and with O in that same branch (no matter the position of the atoms within the strings), since that involves contradiction. This also counts for the case in which x^{I} and x^{O} occur in the same sequence.

We also need a "loop-breaking" rule to stop repeating a string of sentences in the same branch:

(Loop)

$$\frac{\begin{array}{c}\psi \\[1em] x^{\mathrm{V}}\ \phi\end{array}}{\infty\phi}$$

where V∈{I, O}, for any strings of sentences ψ and ϕ (ϕ possibly empty), and if the V-rule for x yields ψ

The Loop-breaking rule is used before the application of the V-rule to check the existence of a forthcoming loop. The mark ∞ is introduced besides the sentence as a signal of the aborted loop (this does not prevent for further development of the branch, contrarily to the case of ×). This rule will always be needed in cases like that of example 2 where we have cycles of attack of even length in the argumentation framework.

In order to avoid other spurious developments, both rules (Closure 1) and (Closure 2) will always have priority over the rule (Loop). We can also gain efficiency by simplifying steps leading to repetitions. For instance, if a string of sentences x^{V}… x^{V}… is produced, then the first occurrence of x^{V} can be eliminated.

Definition 5. A branch of a tableau T is:
— *closed* iff × occurs in it,
— *open* iff it is not closed, and
— *grounded* iff it is open and ∞ does not occur in it.

A tableau T is:
— *closed* iff every branch is closed,
— *open* iff it is not closed, and

— *grounded* iff it has a branch that is grounded.

Example 3. 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^{I} and a^{O} are open but not grounded (Figure 3).

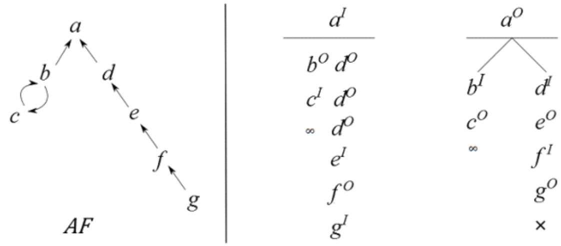

Figure 3

Example 4. Let $AF = \langle\{a, b, c, d\}, \{(b, a), (c, a), (d, c)\}\rangle$. The tableau for a^{I} has only one branch which is closed, hence the tableau is closed. The tableau for a^{O} is also open, but we have two branches, one headed by b^{I}, which is grounded, and another headed by c^{I}, which is closed (Figure 4).

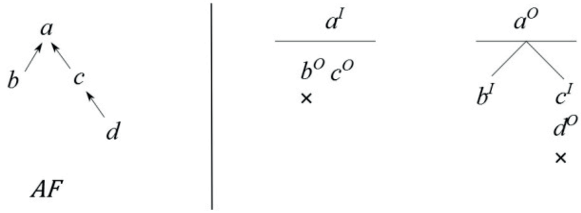

Figure 4

Definition 6. A sentence ϕ is:

— *satisfiable* (in AF) iff the tableau for ϕ is open, and
— *valid* (in AF) iff the tableau for ϕ is grounded.

Obviously, all valid sentences are satisfiable, but not vice versa.

Example 6. (Example 4 revisited). a^{I} is not satisfiable, while a^{O} is satisfiable and valid.

Example 7. (Example 3 revisited). Both a^{I} and a^{O} are satisfiable but not valid.

Theorem 1. If x^{I} (x^{O}) is valid then x^{O} (x^{I}) is not satisfiable.

Proof. We prove that if x^I is valid then x^O is not satisfiable (the proof for the other implication has a similar construction). Assume x^I is valid, then its tableau has a grounded branch. Without lost of generality (i.e. obviating possibly different permutations among the atoms of the sentences), that branch consists of a sequence x^I, $\phi y_1{}^0$, . . . , $y_{2n}{}^I$ (i.e. an odd-length sequence) where $y_{2n}{}^I$ is a non-attacked argument and ϕ refers to the remaining arguments in *Attackers(x)* all signed with O. Let us now assume by contradiction that the tableau of x^O has an open branch. Then we have two cases: either 1) the branch is grounded, or 2) the branch is signed with ∞. Case 1: the branch is grounded, hence the branch consists of a sequence x^O, $w_1{}^I$, . . . , $w_{2n-1}{}^I$ (i.e. an even-length sequence), where w_{2n-1} is a non-attacked argument. Let us now return to the tableau of x^I where we had a branchformed by the sequence x^I, $\phi y_1{}^0$, . . . , $y_{2n-1}{}^I$. Clearly, $w_1{}^0$ occurs in ϕ. Then, by the application of the corresponding rules, a sentence $w_{2n-1}{}^0\psi$ is reached before $y_{2n}{}^I$ leading to $\times$. Contradiction. Case 2: the branch is signed with ∞, hence the branch consists of a looping sequence x^0, ..., $w^V\psi$, . . . , $w^V\psi$. But then the tableau of x^I has also a "looping" branch x^I, ..., $w^{-V}\psi'$, . . . , $w^{-V}\psi'$. Note that, in this branch, a rule (In ...) is applied introducing a sentence in which $y_{2n}{}^I$ or one of its ancestors occurs in the branch x^I, $\phi y_1{}^0$, . . . , $y_{2n}{}^I$ of our hypothesis. In either case, both branches are the same branch, being grounded and not grounded at the same time. Contradiction.

To see that the converse is not true, just consider the argumentation framework $\langle\{a\},\{(a, a)\}\rangle$, where neither a^I nor a^0 are satisfiable.

4 — Justification according to the 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*I is satisfiable if, and only if, *x* is credulously justified with respect to preferred semantics. After that, we will see the necessary and sufficient conditions for skeptical preferred justification.

Proposition 1. An argument x is credulously justified w.r.t. the preferred semantics iff xI is satisfiable.

Proof. We want to show that x^I is satisfiable iff *x* belongs to some preferred extension.
(If) If x^I is satisfiable then its tableau has an open branch. Consider the set S of all the arguments y such that y^I occurs in that branch. Then S is admissible. This implies that all the arguments of S, in particular, *x*, belong to some preferred extension.
(Only if) If *x* belongs to a preferred extension then it belongs to some admissible set S of arguments. That implies that for any argument *y* such that *y* attacks *x*, there exists some argument $z \in S$ such that *z* attacks *y*. Moreover, S is conflict-free. Then, for some admissible set S such that $x \in S$, the tableau of x^I has a branch such that, for every element $y \in S$, y^I occurs in the branch. The assumption that that branch is closed implies that some of those *y* cannot be in, contradicting that S is admissible. Then the branch is open, which proves that x^I is satisfiable.

Now we turn to the problem of how to use the tableau method for proving preferred skeptical justifications. We need to prove the following result before.

Lemma 1. An argument x is skeptically justified w.r.t. the preferred semantics only if x^O is not satisfiable.

Proof. By contraposition, we show that if x^O is satisfiable then there exists some preferred extension to which x does not belong. If x^O is satisfiable then there exists some open branch in its tableau. Since the branch is open, all the arguments signed with I in the branch belong to some admissible set S. But S includes an argument y such that y attacks x. Clearly, y^I is introduced in the tableau immediately after x^O. Now, since S∪{x} is not conflict-free we have that S⊆E and not S∪{x}⊆E for some preferred extension E. Therefore x does not belong to E.

This lemma establishes a necessary but not sufficient condition for an argument to be skeptically justified. The reason is that x^I can be satisfiable while x^O not, even if x does not belong to some preferred extension (see the case of a in Example 8). Moreover, when every preferred extension is stable the condition is also sufficient (i.e. when the argumentation framework is coherent [9]). On the other hand, to prove that x is skeptically justified w.r.t. preferred semantics it suffices to show that x^I is satisfiable and that y^I is not satisfiable, for every argument y∈A such that y^O appears in every open branch of the tableau of x^I. The reason is that if such y^I is satisfiable, then y belongs to some preferred extension to which x does not belong. In Fig. 3, for instance, a is not skeptically justified though a^I is satisfiable and a^O is not: d^O appears in every open branch of a^I but d^I is satisfiable (d belongs to a preferred extension to which a does not belong).

Proposition 2. An argument x is skeptically justified w.r.t. the preferred semantics iff
1. x^I is satisfiable,
2. x^O is not satisfiable, and
3. y^I is not satisfiable, for every argument y∈A such that y^O appears in every open branch of the tableau of x^I.

Proof. (If) We prove that the conjunction of only 1) and 3) implies that x is skeptically justified w.r.t. the preferred semantics. Assume 1) and let us show by contraposition that if x does not belong to some preferred extension, then y^I is satisfiable for some y such that y^O appears in every open branch of the tableau of x^I. Let E be a preferred extension such that x∉E. Then, since E is maximally admissible, E∪{x} is not admissible, which means that either (i) x is in conflict with some y∈E, or (ii) x is not acceptable w.r.t. E, which means that there exists some attacker w of x which is not attacked from E. If (i) is the case, then there exists some y∈E such that y^O appears in the tableau of x^I and y^I is satisfiable. Assume now that (ii) is the case. Since by 1) x^I is satisfiable, then x belongs to some preferred extension E', hence any attack on x is counter-attacked from E'. This implies that some argument of E' which defends x is in conflict with some argument y of E. Clearly, y^O occurs in the tableau of x^I. But, since y belongs to E, then y^{I} is satisfiable. *(Only if)* Assume that x belongs to every preferred extension. Then 1) is obvious while 2) follows from Lemma 1. Assume now that y^{O} appears in every open branch of the tableau of x^{I}. Then x and y cannot belong to the same preferred extension. But, since x belongs to every preferred extension by hypothesis, it follows that y does not belong to any preferred extension. Then, by Proposition 1, y^{I} is not satisfiable.

Example 8. See *AF* in Figure 5. There are two preferred extensions: $\{c, a\}$ and $\{d\}$. No argument is skeptically justified w.r.t. preferred semantics.

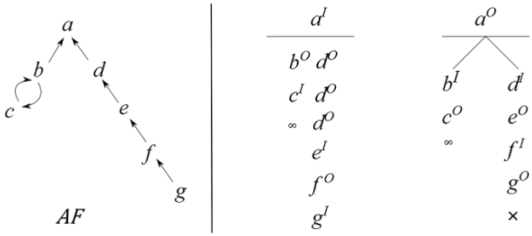

Figure 5

This example shows that a is not justified even when a^{O} is not satisfiable. In labelling semantics this is clear from the fact that there exists a labelling that assigns *undec* to a and *in* to d (i.e. there exists a non empty preferred extension, $\{d\}$, that does not contain a).

5 — Justification according to the grounded semantics

Dung's grounded semantics models a skeptical behaviour which does not coincide in general with skeptical preferred semantics: while the grounded extension is always included in the intersection of all the preferred extensions, the converse is not always the case.

Proposition 3. An argument x is justified w.r.t. the grounded semantics iff x^{I} is valid.

Proof. (If) Assume x^{I} is valid. Then the tableau of x^{I} has a branch that is grounded, i.e. neither × nor ∞ occur in it. This last case implies that the attack relation among the arguments in that branch has no cycles, and the sequence $x_1^{\mathrm{I}}, x_2^{\mathrm{O}}\phi_2, \ldots x_{n-2}^{\mathrm{I}}\phi_{n-2}, x_{n-1}^{\mathrm{O}}, x_n^{\mathrm{I}}$ of all the sentences in the exact order in which they appear in the branch is such that $x_1 = x$ and n is odd. Note that every argument x_j $(1 \leq j \leq n)$ which is signed with I appears in an odd position. Now, since x_n has no attackers we have $x_n \in \mathrm{F}(\emptyset)$. Then, inductively we have $x_{n-2i} \in \mathrm{F}^{i+1}(\emptyset)$, $0 \leq i \leq (n\text{-}1)/2$, getting $x_1 \in \mathrm{F}^{(n+1)/2}(\emptyset)$, where $\mathrm{F}^{(n+1)/2}(\emptyset)$ is the least fixed point of F, that is, the grounded extension of the framework.

(Only if) By contraposition, assume that x^{I} is not valid and let us prove that x is not justified w.r.t. the grounded semantics. The hypothesis implies that no branch of the tableau of x^{I} is grounded. The case that x^{I} is not satisfiable clearly leads to the conclusion that x does not belong to the grounded extension. Otherwise, if x is satisfiable (and x^{I} being not valid), x^{I} should appear in an open branch marked with ∞. Then neither *x* nor any of its defenders is a non-attacked argument (i.e. the defense of *x* is always involved in a cycle of attacks). Then none of those arguments are in F(∅) and, then, they cannot belong to the least fixed point of F neither. Therefore, *x* does not belong to the grounded extension.

6 — 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 labelling 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*, while there is no way to begin a game with Pro advancing an argument she wants to show that is *out*. Our method, instead, enables to represent both situations.

On the other hand, our method shows the same 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 are no odd-length cycles of attack, stable semantics coincides with preferred semantics, hence the tableaux proofs for stable justifications are also limited to coherent argumentation frameworks. Moreover, our tableaux method is not computationally better than dialogue games, though it is not worse either.

Finally, in our opinion, the tableaux method we have proposed is intuitive, resembling the analytic tableaux which are familiar for logics students, and for this reason we think it can be a useful tool for teaching argumentation frameworks semantics.

ACKNOWLEDGMENTS

This paper was supported by ANPCyT (National Agency for the Promotion of Science and Technology), PICT 2013 – 1489, and Universidad Nacional del Sur, PGI 24/I223, Argentina.

REFERENCES

BETH, E., *Semantic Entailment and Formal Derivability*, Mededelingen van de Koninklijke Nederlandse Akademie van Wetenschappen, Afdeling Letterkunde 18 (13) (1955), 309-342. Reprinted in J. Hintikka (ed.) *The Philosophy of Mathematics*, Oxford University Press, 1969.

CAMINADA, M., On the Issue of Reinstatement in Argumentation. *In*: *Proceedings of Logics in Artificial Intelligence*, 10th European Conference, JELIA, LNCS 4160, Springer, 2006, pp. 111-123.

CAYROL, C., DOUTRE, S. & MENGIN, J., Dialectical Proof Theories for the Credulous Preferred Semantics of Argumentation Frameworks. *In: Proceedings of the 6th ECSQARU*, LNAI 2143, Springer. 2001, pp. 668-679.

CHESÑEVAR, C.I. & SIMARI G.R., An Abstract Model for Computing Warrant in Skeptical Argumentation Frameworks. *In: Proceedings of the 11th International Workshop on Nonmonotonic Reasoning (NMR 2006)*, IfI Technical Report Series, Clausthal University 2006, pp. 411-419.

CHESÑEVAR, C.I. & SIMARI G.R., A Lattice-based Approach to Computing Warranted Belief in Skeptical Argumentation Frameworks. *In: Proceedings of the IJCAI*, 2007, pp. 280-285.

DUNG, P.M., On the Acceptability of Arguments and its Fundamental Role in Non-Monotonic Reasoning, Logic Programming, and n-Person Games. *Artificial Intelligence* **77**: 321-357, 1995.

JAKOBOVITS, H. & VERMEIR, D. Robust Semantics for Argumentation Frameworks. *Journal of Logic and Computation* **9** (2): 215-261, 1999.

MODGIL, S. & CAMINADA, M., Proof Theories and Algorithms for Abstract Argumentation Frameworks. *In:* Rahwan, I., & Simari, G. (eds.), *Argumentation in Artificial Intelligence*, Springer, 2009, pp. 105-129.

NOFAL, S., ATKINSON, K. & DUNNE, P., Algorithms for Decision Problems in Argument Systems under Preferred Semantics. *Artificial Intelligence* **207**, 23-51, 2014.

SMULLYAN, R., *First Order-Logic*. Dover Publications, 1998.

VREESWIJK, G. & PRAKKEN, H., Credulous and Sceptical Argument Games for Preferred Semantics. *In*: *Proceedings of the JELIA 2000*, LNAI 1919, Springer, 239-253, 2000.