Studies in Logic, Vol. 3, No. 4 (2010): 12–29
PII: 1674-3202(2010)-04-0012-18

A Labelling-Based Justification Status of Arguments

Yining Wu
University of Luxembourg
yining.wu@uni.lu

Martin Caminada
University of Luxembourg
martin.caminada@uni.lu

Abstract. In this paper, we define a labelling-based justification status of the arguments in an argumentation framework. Our proposal allows for a more fine-grained notion of a justification status than the traditional extensions-based approaches. In particular, we are able to distinguish different levels at which an argument can be accepted or rejected. Our approach is fully compatible with traditional abstract argumentation in the sense that it works on standard argumentation frameworks and can be implemented using existing argumentation-based proof procedures.

1. Introduction

The main concept in Dung's theory ([12]) is that of an argumentation framework, which is essentially a directed graph in which the nodes represent arguments and the arrows represent an attack relation.

Given such a graph, different sets of nodes can be accepted according to various argument based semantics such as grounded, preferred and stable semantics ([12]), semi-stable semantics ([10, 21]) or ideal semantics ([13]). Many of these semantics can be seen as restricted cases of complete semantics; an overview is provided in Figure 1. The facts that every stable extension is also a semi-stable extension and that every semi-stable extension is also a preferred extension have been proved in [10]. The facts that every preferred extension is also a complete extension and that the grounded extension is also a complete extension have been stated in [12]. The ideal extension is also a complete extension.([13]) So complete extensions can be seen as the base for describing various other semantics in abstract argumentation.

A different way of defining argumentation semantics than the traditional extensions approach is the labellings approach. Where the extensions approach only identifies the set of arguments that are accepted, the labellings approach also identifies the set of arguments that are rejected and the set of arguments that are left undecided. The concept of argument labellings goes back to work of Pollock ([18]) and of Jakobovits and Vermeir([15]). However, for current purposes we will use the concept of complete labelling as defined by [5, 8].

Essentially, a complete labelling can be seen as a subjective but reasonable position that an agent can take with respect to which arguments are accepted, rejected or

Received 2010-10-05

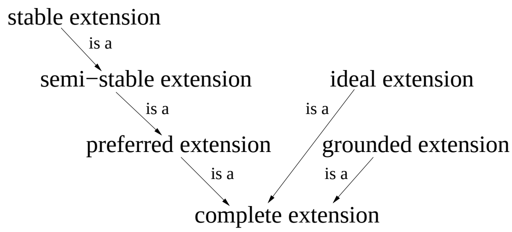

Figure 1: An overview of the different semantics

undecided. In each such position the agent can use its own position to defend itself if questioned. It is possible to disagree with a position, but at least the position is internally coherent. The set of all complete labellings thus stands for all possible and reasonable positions an agent can take.

In [8], it is stated that complete extensions and complete labellings are one-to-one related. In essence, complete extensions and complete labellings are different ways to describe the same concept.

In the current paper we will propose justification statuses of arguments based on the notion of a complete labelling. One of the main advantages of our proposal is that it allows for a more fine-grained notion of a justification status than is provided by the traditional extensions-based approaches. In particular, it allows for six distinct justification statuses (strong accept, weak accept, strong reject, weak reject, undetermined border line and determined border line) which correspond with different levels of acceptance and rejection. Furthermore, our proposal is fully compatible with [12] in the sense that it works on standard argumentation frameworks and can be implemented using existing argumentation-based proof procedures.([16, 22])

The remaining part of this paper is organized as follows. We first state some preliminaries on argument semantics and argument labellings (Section 2). Then we define the justification status of an argument (Section 3). And describe a software implementation that is able to compute this status, given an argumentation framework (Section 4). The connection between our approach and the existing argumentation semantics is discussed in Section 5. We round off with a discussion on justification statuses of conclusions in Section 6. We then round up with a discussion of how our notion of a justification state relates to existing well-known approaches.

2. Argument Semantics and Argument Labellings

In this section, we briefly restate some preliminaries regarding argument semantics and argument-labellings. For simplicity, we only consider finite argumentation frameworks.

Definition 1 An *argumentation framework* is a pair (Ar, att) where Ar is a finite set of arguments and $att \subseteq Ar \times Ar$.

We say that argument A *attacks* argument B iff $(A, B) \in att$. An argumentation framework can be represented as a directed graph in which the arguments are represented as nodes and the attack relation is represented as arrows.

Definition 2 (defense / conflict-free) Let (Ar, att) be an argumentation framework, $A \in Ar$ and $\mathcal{A}rgs \subseteq Ar$. $\mathcal{A}rgs$ is *conflict-free* iff $\neg\exists A, B \in \mathcal{A}rgs$: A attacks B. $\mathcal{A}rgs$ *defends* argument A iff $\forall B \in Ar$: (B attacks $A \supset \exists C \in \mathcal{A}rgs$: C attacks B). Let $F(\mathcal{A}rgs) = \{A \mid A$ is defended by $\mathcal{A}rgs\}$.

We say that a set of arguments $\mathcal{A}rgs$ attacks an argument B iff there exists an $A \in \mathcal{A}rgs$ that attacks B. We write $\mathcal{A}rgs^+$ for the set of arguments that are attacked by $\mathcal{A}rgs$.

Definition 3 (acceptability semantics) Let (Ar, att) be an argumentation framework. A conflict-free set $\mathcal{A}rgs \subseteq Ar$ is called an *admissible set* iff $\mathcal{A}rgs \subseteq F(\mathcal{A}rgs)$, and a *complete* extension iff $\mathcal{A}rgs = F(\mathcal{A}rgs)$.

The concept of complete semantics was originally stated in terms of sets of arguments. It is equally well possible, however, to express this concept in terms of *argument labellings*. In the current paper, we follow the approach of [5, 9] where a labelling assigns to each argument exactly one label, which can either be in, out or undec. The label in indicates that the argument is accepted, the label out indicates that the argument is rejected, and the label undec indicates that the status of the argument is undecided, meaning that one abstains from an explicit judgment whether the argument is in or out.[1]

Definition 4 ([5]) A *labelling* is a function $\mathcal{L}ab : Ar \longrightarrow \{\mathrm{in}, \mathrm{out}, \mathrm{undec}\}$.

We write $\mathrm{in}(\mathcal{L}ab)$ for $\{A \mid \mathcal{L}ab(A) = \mathrm{in}\}$, $\mathrm{out}(\mathcal{L}ab)$ for $\{A \mid \mathcal{L}ab(A) = \mathrm{out}\}$ and $\mathrm{undec}(\mathcal{L}ab)$ for $\{A \mid \mathcal{L}ab(A) = \mathrm{undec}\}$. Since a labelling can be interpreted as a partition of the set of arguments in the argumentation framework, we will sometimes write a labelling $\mathcal{L}ab$ as a triple $(\mathrm{in}(\mathcal{L}ab), \mathrm{out}(\mathcal{L}ab), \mathrm{undec}(\mathcal{L}ab))$.

The idea of a *complete labelling* ([5, 9]) is that for a labelling to be reasonable, one should be able to give reasons for each argument one accepts (all attackers are rejected), for each argument one rejects (it has at least one attacker that is accepted) and for each argument one abstains from expressing an explicit opinion about (there

[1] For instance, an argument that attacks itself (and is not attacked by any other argument) has to be labelled undec in our approach. If the argument would be labelled in then all its attackers (itself) would have to be out, and if the argument would be labelled out then it has to have an attacker (itself) that is in. Hence, the argument cannot be in and cannot be out. The situation here can be compared to the liar paradox.

are insufficient grounds to accept it and insufficient grounds to reject it). This is made formal in the following definition.

Definition 5 ([5]) Let $\mathcal{L}ab$ be a labelling of argumentation framework (Ar, att). We say that $\mathcal{L}ab$ is a *complete labelling* iff for each $A \in Ar$ it holds that:

1. If $\mathcal{L}ab(A) = \text{in}$ then
   $\forall B \in Ar : (B \ att \ A \supset \mathcal{L}ab(B) = \text{out})$
2. If $\mathcal{L}ab(A) = \text{out}$ then
   $\exists B \in Ar : (B \ att \ A \wedge \mathcal{L}ab(B) = \text{in})$.
3. If $\mathcal{L}ab(A) = \text{undec}$ then
   $\neg\forall B \in Ar : (B \ att \ A \supset \mathcal{L}ab(B) = \text{out})$ and
   $\neg\exists B \in Ar : (B \ att \ A \wedge \mathcal{L}ab(B) = \text{in})$.

As stated in [5, 9], complete labellings coincide with complete extensions in the sense of [12].

Theorem 1 ([5]) Let $AF = (Ar, att)$ be an argumentation framework.

1. If $\mathcal{L}ab$ is a complete labelling, then $\text{Lab2Ext}(\mathcal{L}ab)$ is a complete extension (where $\text{Lab2Ext}(\mathcal{L}ab) = \text{in}(\mathcal{L}ab)$)
2. If $\mathcal{A}rgs$ is a complete extension, then $\text{Ext2Lab}(\mathcal{A}rgs)$ is a complete labelling (where $\text{Ext2Lab}(\mathcal{A}rgs) = (\mathcal{A}rgs, \mathcal{A}rgs^+, Ar \backslash (\mathcal{A}rgs \cup \mathcal{A}rgs^+)))$

Moreover, when restricted to complete labellings and complete extensions, the functions Lab2Ext and Ext2Lab become bijections and each others inverses.

Theorem 1 implies that complete labellings and complete extensions are one-to-one related. In essence, a complete extension can be seen as the in-labelled part of a complete labelling. ([5, 9])

Before we proceed, we state two propositions that are used in the remaining parts of this paper.

Proposition 1 Let $AF = (Ar, att)$ be an argumentation framework and $A \in Ar$. A is in at least one complete extension iff it is in at least one admissible set.

The validity of Proposition 1 can be seen as follows. Since every complete extension is also an admissible set, it follows that if A is in a complete extension, it is also in an admissible set. Furthermore, if A is in an admissible set, then from [12] it follows that A is also in a preferred extension, and every preferred extension is also a complete extension.

Proposition 2 Let $AF = (Ar, att)$ be an argumentation framework and $A \in Ar$. A is in all complete extensions iff A is in the grounded extension.

The validity of Proposition 2 can be seen as follows. Since the grounded extension is a complete extension, it follows that if an argument is in every complete extension, it is also in the grounded extension. Furthermore, since the grounded extension is the smallest complete extension, it follows that if an argument is in the grounded extension, it is also in every complete extension.

3. Justification Statuses of Arguments

In this section we first define the justification statuses of arguments. Then we provide procedures to determine them. Intuitively, the justification status of an argument consists of the set of labels that could reasonably be assigned to the argument.

Definition 6 Let $AF = (Ar, att)$ be an argumentation framework and $A \in Ar$. The justification status of A is the outcome yielded by the function $\mathcal{JS} : Ar \to 2^{\{\text{in},\text{out},\text{undec}\}}$ such that $\mathcal{JS}(A) = \{\mathcal{L}ab(A) \mid \mathcal{L}ab$ is a complete labelling of $AF\}$.

Given the above definition, one would expect there to be eight (2^3) possible justification statuses, one for each subset of $\{\text{in}, \text{out}, \text{undec}\}$. However two of these subsets turn out not to be possible. First of all, it is not possible for a justification status to be $\emptyset$, because there always exists at least one complete labelling (the grounded labelling ([5])). Furthermore, it is also impossible for a justification status to be $\{\text{in}, \text{out}\}$, because when in and out are both included in the justification status, then undec should also be included, as is stated by the following theorem.

Theorem 2 Let $AF = (Ar, att)$ be an argumentation framework and $A \in Ar$. If AF has two complete labellings $\mathcal{L}ab_1$ and $\mathcal{L}ab_2$ such that $\mathcal{L}ab_1(A) = \text{in}$ and $\mathcal{L}ab_2(A) = \text{out}$ then there exists a complete labelling $\mathcal{L}ab_3$ such that $\mathcal{L}ab_3(A) = \text{undec}$.

Proof Let $CE_1 = \text{Lab2Ext}(\mathcal{L}ab_1)$ and $CE_2 = \text{Lab2Ext}(\mathcal{L}ab_2)$. From Theorem 3 of [5] it follows that CE_1 and CE_2 are complete extensions of AF. Let GE be the grounded extension of AF. From [12] it follows that GE is the intersection of all complete extensions of AF. From $\mathcal{L}ab_2(A) = \text{out}$, it follows that $A \notin CE_2$ which implies that $A \notin GE$. From $\mathcal{L}ab_1(A) = \text{in}$, it follows that $\forall B \in Ar.(B att A \supset \mathcal{L}ab_1(B) = \text{out})$. Therefore, $\forall B \in Ar.(B att A \supset B \notin GE)$. So $A \notin GE^+$. Let $\mathcal{L}ab_3 = \text{Ext2Lab}(GE)$. GE is a complete extension, so $\mathcal{L}ab_3$ is a complete labelling. Since $A \notin GE$ and $A \notin GE^+$, it holds that $A \in Ar\backslash(GE \cup GE^+)$. So $\mathcal{L}ab_3(A) = \text{undec}$. □

Since $\emptyset$ and $\{\text{in}, \text{out}\}$ are not possible as justification statuses, there are only 6 possible statuses left to be considered: $\{\text{in}\}$, $\{\text{out}\}$, $\{\text{undec}\}$, $\{\text{in}, \text{undec}\}$, $\{\text{out}, \text{undec}\}$ and $\{\text{in}, \text{out}, \text{undec}\}$. We now examine under which conditions these justification statuses occur.

First, we examine the conditions under which the justification status is $\{\text{in}\}$.

Theorem 3 Let $AF = (Ar, att)$ be an argumentation framework and $A \in Ar$. Then $\mathcal{JS}(A) = \{\text{in}\}$ iff A is in the grounded extension.

Proof "⇒": Suppose $\mathcal{JS}(A) = \{\text{in}\}$. Then A is labelled in by every complete labelling (Definition 6), so A is an element of each complete extension (Theorem 1) so A is in the grounded extension (Proposition 2).
"⇐": Similar as above, but the other way around. □

Next, we examine the conditions under which the justification status is $\{\text{out}\}$.

Theorem 4 Let $AF = (Ar, att)$ be an argumentation framework and $A \in Ar$. Then $\mathcal{JS}(A) = \{\text{out}\}$ iff A is attacked by the grounded extension.

Proof "⇒": Suppose $\mathcal{JS}(A) = \{\text{out}\}$. Then A is labelled out by every complete labelling (Definition 6). So in every complete labelling, there exists at least one attacker of A that is labelled in by this labelling (Definition 5). So every complete extension contains at least one attacker of A (Theorem 1). So also the grounded extension also contains an attacker of A. So A is attacked by the grounded extension.
"⇐": Similar as above, but the other way around. □

Next, we examine the conditions under which the justification status is $\{\text{undec}\}$.

Theorem 5 Let $AF = (Ar, att)$ be an argumentation framework and $A \in Ar$. Then $\mathcal{JS}(A) = \{\text{undec}\}$ iff

1. A is not in any admissible set and
2. A is not attacked by any admissible set

Proof "⇒": Suppose $\mathcal{JS}(A) = \{\text{undec}\}$. Then it holds that (1) A is not labelled in by any complete labelling and (2) A is not labelled out by any complete labelling. From (1) it follows that A is not an element of any complete extension (Theorem 1) so A is not an element of any admissible set (Proposition 1). From (2) it follows that no attacker of A is labelled in by any complete labelling (Definition 5) so no attacker of A is in any complete extension (Theorem 1) so no attacker of A is in any admissible set (Proposition 1) so A is not attacked by any admissible set. Notice that in this proof, we did not use the fact that A is labelled undec by at least one complete labelling, which after all is implied by (1) and (2) together with Theorem 2.
"⇐": Suppose that (1) A is not in any admissible set and (2) A is not attacked by any admissible set. From (1) it follows that A is not in any complete extension (Proposition 1) so A is not labelled in by any complete labelling (Theorem 1). From (2) it follows that no attacker of A is in any admissible set, so no attacker of A is in any complete extension (Proposition 1) so no attacker of A is labelled in by any complete labelling (Theorem 1) so A is not labelled out by any complete labelling (Definition

5). This, together with the earlier observed fact that A is not labelled in by any complete labelling implies that A is labelled undec by every complete labelling. Due to the fact that there always exists at least one complete labelling (since there always exists at least one complete extension), this implies that $\mathcal{JS} = \{\mathrm{undec}\}$. □

Next, we examine the conditions under which the justification status is $\{\mathrm{in}, \mathrm{undec}\}$.

Theorem 6 Let $AF = (Ar, att)$ be an argumentation framework and $A \in Ar$. Then $\mathcal{JS}(A) = \{\mathrm{in}, \mathrm{undec}\}$ iff

1. A is not in the grounded extension,
2. A is in an admissible set, and
3. A is not attacked by any admissible set.

Proof "⇒": Suppose $\mathcal{JS}(A) = \{\mathrm{in}, \mathrm{undec}\}$. Then A is labelled in by at least one complete labelling, A is labelled undec by at least one complete labelling and there exists no complete labelling that labels A out.
From the fact that A is labelled undec in at least one complete labelling it follows that there exists at least one complete extension that does not contain A (Theorem 1). So A is not in the grounded extension (Proposition 2).
From the fact that A is labelled in by at least one complete labelling it follows that A is contained in at least one complete extension (Theorem 1) and that therefore A is in at least one admissible set (Proposition 1).
From the fact that there exists no complete labelling that labels A out it follows (Definition 5) that for all arguments B that attack A, B is not labelled in by any complete labelling. Therefore, no argument B that attacks A is contained in any complete extension (Theorem 1). Therefore, no argument B that attacks A is in any admissible set (Proposition 1). That is, A is not attacked by any admissible set.
"⇐": Suppose that (1) A is not in the grounded extension, (2) A is in an admissible set and (3) A is not attacked by any admissible set.
From (2) it follows that A is in a complete extension (Proposition 1) so A is labelled in by a complete labelling (Theorem 1).
From (3) it follows that no admissible set contains an attacker of A so also no complete extension contains any attacker of A (Proposition 1). So no complete labelling labels any attacker of A in (Theorem 1), so A is not labelled out by any complete labelling (Definition 5).
From (1) it follows that there exists a complete labelling where A is not labelled in (Proposition 2 and Theorem 1). This, together with the earlier observed fact that A is not labelled out by any complete labelling, implies that A is labelled undec by at least one complete labelling. □

Next, we examine the conditions under which the justification status is $\{\mathrm{out}, \mathrm{undec}\}$.

Theorem 7 Let $AF = (Ar, att)$ be an argumentation framework and $A \in Ar$. Then $\mathcal{JS}(A) = \{\text{out}, \text{undec}\}$ iff

1. A is not in any admissible set,
2. A is attacked by an admissible set, and
3. A is not attacked by the grounded extension.

Proof "⇒": Suppose $\mathcal{JS}(A) = \{\text{out}, \text{undec}\}$. Then (1) there exists no complete labelling that labels A in, (2) there exists a complete labelling that labels A out and (3) there exists a complete labelling that labels A undec.
From (1) it follows that A is not an element of any complete extension (Theorem 1) so A is not an element of any admissible set (Proposition 1).
From (2) it follows that A is attacked by at least one complete extension (Theorem 1) so A is attacked by at least one admissible set (Proposition 1).
From (3) it follows that there exists a complete labelling where A is not labelled out, so where none of the attackers of A are labelled in (Definition 5). It then follows that there exists a complete extension that contains none of the attackers of A (Theorem 1). So none of the attackers of A are contained in the grounded extension (Proposition 2) so A is not attacked by the grounded extension.
"⇐": Suppose that (1) there exists no admissible set that contains A, (2) there is an admissible set that attacks A, and (3) A is not attacked by the grounded extension.
From (1) it follows that A is not an element of any complete extension (Proposition 1), so A is not labelled in by any complete labelling (Theorem 1).
From (2) it follows that A is attacked by a complete extension (Proposition 1) so A is labelled out by at least one complete labelling (Theorem 1).
From (3) it follows that no attacker of A is in the grounded extension. This implies that there exists a complete extension that does not contain any attacker of A (Proposition 2). So there exists a complete labelling where no attacker of A is labelled in (Theorem 1), so where A is not labelled out (Definition 5). This, together with the earlier observed fact that A is not labelled in by any complete labelling, implies that A is labelled undec by at least one complete labelling. □

Next, we examine the conditions under which the justification status is $\{\text{in}, \text{out}, \text{undec}\}$.

Theorem 8 Let $AF = (Ar, att)$ be an argumentation framework and $A \in Ar$. Then $\mathcal{JS}(A) = \{\text{in}, \text{out}, \text{undec}\}$ iff

1. A is in an admissible set
2. A is attacked by an admissible set

Proof "⇒": Suppose $\mathcal{JS}(A) = \{\text{in}, \text{out}, \text{undec}\}$. Then (1) A is labelled in by at least one complete labelling and (2) A is labelled out by at least one complete labelling.

From (1) it follows that A is an element of at least one complete extension (Theorem 1) so A is an element of at least one admissible set (Proposition 1).
From (2) it follows that there is a complete labelling that labels an attacker of A in (Definition 5). Therefore there exists a complete extension that contains an attacker of A (Theorem 1), so there exists an admissible set that contains an attacker of A (Proposition 1). That is, A is attacked by an admissible set.
"$\Leftarrow$": Suppose (1) there exists an admissible set that contains A and (2) there exists an admissible set that contains an attacker of A.
From (1) it follows that there exists a complete extension that contains A (Proposition 1). so there exists a complete labelling that labels A in (Theorem 1).
From (2) it follows that there exists a complete extension that contains an attacker of A (Proposition 1), so there exists a complete labelling that labels an attacker of A in (Theorem 1), so there exists a complete labelling where A is labelled out.
From the fact that there exists a complete labelling that labels A in and there exists a complete labelling that labels A out it follows that there also exists a complete labelling that labels A undec (Theorem 2). □

From the above theorems, it follows that membership of an admissible set and membership of the grounded extension, of the argument itself and of its attackers, is sufficient to determine the argument's justification status. The overall procedure of doing so is shown in Figure 2.

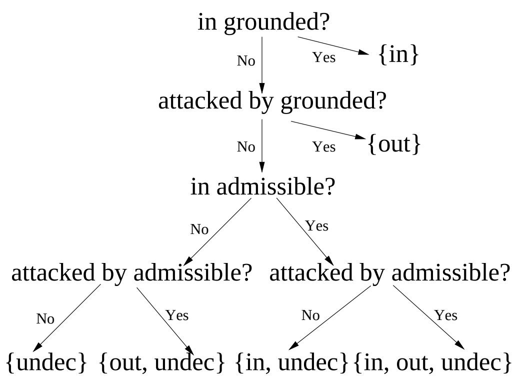

Figure 2: determining the justification status of an argument

4. An Implementation

We now demonstrate the applicability of the theory developed in the previous sections by describing our software implementation of it.[2]

[2]Authored by Mikolaj Podlaszewski. And available online at http://argulab.uni.lu/.

In order to determine the justification status of an argument, our implementation follows the procedure of Figure 2. To determine whether an argument is in the grounded extension, the algorithm described in [16] is used. This algorithm is subsequently run for the argument's attackers in oder to determine whether the argument is attacked by the grounded extension. To determine whether an argument is in an admissible set, the algorithm described in [6, 22] is used. This algorithm is subsequently run for the argument's attackers in order to determine whether the argument is attacked by an admissible set.

The software can determine the justification statuses, and is able to defend them by entering the applicable discussion game with the user. Since the software can be expected to compute the correct justification status, the game is such that in the end the software always wins from the user. Hence, the software is able to explain why its answer is correct, by entering a discussion with the user. Again, to the best of our knowledge, the current simulator is the first implementation of labelling-based justification statuses.

Our software has an easy to use graphical interface. Argumentation frameworks are displayed as graphs and arguments are labelled according to the labelling chosen by users, or according to the output of the software (given a particular semantics). The software consists of several components which (under GPL) could be re-applied in other argumentation software. Hence, someone implementing, say, an argumentation-based on-line discussion forum could apply some of our components to provide a snapshot of, for instance, the justification status of a set of arguments given a particular state of the discussion.

5. Labelling-Based Justification Statuses vs. Existing Argumentation Semantics

In this paper, we have presented the justification statuses of arguments which indicate whether an argument has to be accepted, can be accepted, has to be rejected, can be rejected, etc. We then provided some concrete guidelines for determining these justification statuses, as well as for defending them using discussion games.

We use this labelling-based approach for computing the justification statuses of arguments because it tends to yield more informative answers than the traditional extension-based approaches.

Consider the example in Figure 3. A possible interpretation would be as follows:

A: Carole hates David according to Alice.

B: Carole does not know David according to Bob.

C: Carole says that David is not reliable.

D: David is trustworthy, since he has a general reputation of being so.

Grounded semantics treats all arguments (A, B, C and D) the same (they are not

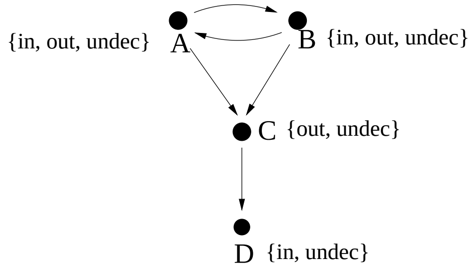

Figure 3: An example

labelled in in the grounded labelling). Credulous preferred semantics treats A, B and D the same (they are labelled in in at least one preferred labelling). Sceptical preferred semantics treats A, B and C the same (they are not labelled in in some preferred labellings). Also ideal semantics treats all arguments the same (they are not in the ideal extension).

However, our labelling-based approach for computing the justification status of an argument allows for a more fine grained distinction between arguments. According to the hierarchy of the justification statuses in Figure 4, argument D is the strongest, argument C is the weakest, A and B are in between. Unlike sceptical preferred semantics, our labelling approach does not make D completely justified although it does give it a relatively strong status.

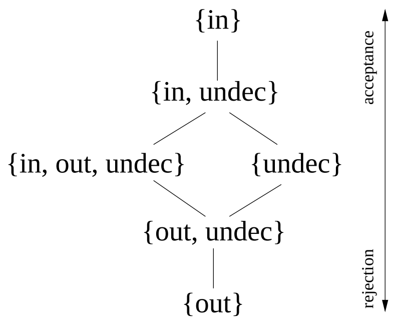

Figure 4: The hierarchy of justification statuses

We will refer to the justification status $\{\text{in}\}$ as *strong accept*, to $\{\text{in}, \text{undec}\}$ as *weak accept*, to $\{\text{in}, \text{out}, \text{undec}\}$ as *undetermined borderline*, to $\{\text{undec}\}$ as *determined borderline*, to $\{\text{out}, \text{undec}\}$ as *weak reject* and to $\{\text{out}\}$ as *strong reject*.

We now study some of the connections between our notion of justification status and a number of existing approaches. In particular, we examine the connection with grounded semantics ([12]), credulous preferred semantics ([22]), sceptical preferred semantics ([11]), semi-stable semantics ([10]) and ideal semantics ([13]).

Proposition 3 Let (Ar, att) be an argumentation framework and $A \in Ar$.

1. A is in the grounded extension iff it is strongly accepted
2. A is in at least one preferred extension iff A is strongly accepted, weakly accepted, or undetermined borderline.
3. if A is in every preferred extension then A is strongly or weakly accepted
4. if A is strongly accepted then A is in every semi-stable extension
   if A is weakly accepted then A is in at least one semi-stable extension
5. A is in an ideal set iff A is member of an admissible set consisting only of strongly or weakly accepted arguments.

The validity of point 1 follows directly from Theorem 3. The validity of point 2 follows from the fact that an argument is in a preferred extension iff it is in a complete extension, and therefore labelled in by a complete labelling. The validity of point 3 follows from the fact that sceptical preferred rules out all justification statuses containing out (strong reject, weak reject and undetermined borderline) as well as the justification status $\{\mathrm{undec}\}$ (determined borderline), which means only $\{\mathrm{in}\}$ (strong accept) and $\{\mathrm{in}, \mathrm{out}\}$ (weak accept) remain. The validity of point 4 follows from Theorem 5 of [10]. The validity of point 5 requires some more explanation, which will be provided in the appendix.

The labelling-based approach for determining justification statuses is somewhat similar to the approach described in [3]. However, in [3] the authors do not specify a concrete semantics with which to apply their approach to, and as a result of this, they do not provide any procedures regarding how to determine the justification status of an argument.

In our current implementation, we have used the discussion game of [6, 22] to determine membership of an admissible set, and the discussion game of [7, 16, 20] to determine membership of the grounded extension. An alternative would be to use the algorithm of [23], which determines both of these memberships in a single pass. Since our notion of justification status depends only on membership of an admissible set and membership of the grounded extension, one is free to apply any kind of algorithm that can determine these.

6. A Labelling-Based Justification Statuses of Conclusions

So far we only talked about justification statuses of arguments. However, what matters often are the *conclusions* of these arguments. In this section we will treat a possible approach for defining justification statuses of conclusions. If A is an argument then following the ASPIC formalism([1, 4, 17]) we write $\mathrm{Conc}(A)$ for the conclusion of A. Every argument is assumed to have exactly one conclusion which is essentially a formula in some logical language.([1, 4, 17]) It is possible for different arguments to have the same conclusion.

We first define the notion of a conclusion labelling.

Definition 7 Let $\mathcal{L}$ be a logical language. A *conclusion labelling* is a function $\mathcal{ConcLab} : \mathcal{L} \to \{\text{in}, \text{out}, \text{undec}\}$

The next step is to define how to convert an argument labelling ($\mathcal{ArgLab}$) into a conclusion labelling ($\mathcal{ConcLab}$). For this we use a function $\mathcal{ArgLab2ConcLab}$. Basically, the idea is to associate each conclusion with the label of the best possible argument that is able to produce this conclusion.[3] So if there are three arguments A_1, A_2 and A_3 with conclusion c, where $\mathcal{ArgLab}(A_1) = \text{in}$, $\mathcal{ArgLab}(A_2) = \text{out}$ and $\mathcal{ArgLab}(A_3) = \text{undec}$, then $\mathcal{ConcLab}(c) = \text{in}$, since the best argument with conclusion c (A_1) is labelled in. Similarly, if there are three arguments B_1, B_2 and B_3 with conclusion d, where $\mathcal{ArgLab}(B_1) = \text{out}$, $\mathcal{ArgLab}(B_2) = \text{undec}$ and $\mathcal{ArgLab}(B_3) = \text{out}$. Then $\mathcal{ConcLab}(d) = \text{undec}$, since the best argument with conclusion d (B_2) is labelled undec. If for a particular conclusion, there is no argument at all that produces it, then the conclusion is given the lowest possible label (out). Formally, the translation from an argument labelling to a conclusion labelling can be defined as follows.

Definition 8 Let $AF = (Ar, att)$ be an argumentation framework whose conclusions belong to logical language $\mathcal{L}$. Let $\mathcal{ArgLabs}$ be the set of all argument labellings of AF and $\mathcal{ConcLabs}$ be the set of all conclusion labellings of $\mathcal{L}$. We define a function $\mathcal{ArgLab2ConcLab} : \mathcal{ArgLabs} \to \mathcal{ConcLabs}$ such that, given a labelling $\mathcal{ArgLab}$ of AF, the associated conclusion labelling $\mathcal{ConcLab} = \mathcal{ArgLab2ConcLab}(\mathcal{ArgLab})$ is such that for every $c \in \mathcal{L}$ it holds that $\mathcal{ConcLab}(c) = \max(\{\mathcal{ArgLab}(A) \mid \text{Conc}(A) = c\} \cup \{\text{out}\})$.

We say that a conclusion labelling $\mathcal{ConcLab}$ is a *complete conclusion labelling* of AF iff there exists a complete argument labelling $\mathcal{ArgLab}$ of AF such that $\mathcal{ConcLab} = \mathcal{ArgLab2ConcLab}(\mathcal{ArgLab})$. This then allows us to define the justification status of a conclusion as the set of labels that can reasonably be assigned to it.

Definition 9 Let $AF = (Ar, att)$ be an argumentation framework whose conclusions belong to logical language $\mathcal{L}$ and $c \in \mathcal{L}$. The *justification status of* c is the outcome yielded by the function $\mathcal{ConcJS} : \mathcal{L} \to 2^{\{\text{in},\text{out},\text{undec}\}}$ such that $\mathcal{ConcJS}(c) = \{\mathcal{ConcLab}(c) \mid \mathcal{ConcLab}$ is a complete conclusion labelling of $AF\}$.

To illustrate the usefulness of justification statuses of conclusions, consider the following two examples. [4]

[3] We assume the labels to be ordered such that $\text{in} > \text{undec} > \text{out}$.

[4] Example 1 and Example 2 are taken from [19].

Example 1
A: Brygt Rykkje is Dutch since he was born in Holland.
B: Brygt Rykkje is Norwegian since he has a Norwegian name.
C: Brygt Rykkje likes ice skating since he is Norwegian.
D: Brygt Rykkje likes ice skating since he is Dutch.

Example 2
A: John says that the suspect stabbed the victim.
B: Bob says that the suspect shot the victim.
C: The suspect killed the victim since Bob says that the suspect shot the victim.
D: The suspect killed the victim since John says that the suspect stabbed the victim.

In Example 1, if we assume that a person cannot be Dutch and Norwegian at the same time, then arguments A and B attack each other. Furthermore, argument A attacks argument C and argument B attacks argument D. Similarly, in Example 2, if we assume that a person can only be killed once, then arguments A and B attack each other. Furthermore, also in Example 2, argument A attacks argument C and argument B attacks argument D. In essence, the two examples share the same argumentation framework illustrated in Figure 5. [5]

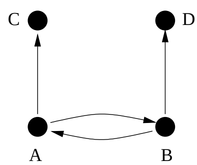

Figure 5: Floating Conclusion

Despite the fact that the two examples share the same formalization (the argumentation framework of Figure 5), in Example 1 it seems that the fact that Brygt Rykkje likes ice skating is a reasonable conclusion, whereas in Example 2 the fact that the suspect is guilty seems to be lacking sufficient support, at least from the perspective of a judge having to reach a verdict on the case. [6]

Let us now consider how this situation is handled by our approach of conclusion-based justification statuses. We first observe that in each of the examples argument C and D have the same conclusion (say conclusion c). For the argumentation framework of Figure 5, one can distinguish three different complete argument labellings:

$\mathcal{A}rg\mathcal{L}ab_1 = (\{A, D\}, \{B, C\}, \emptyset)$,
$\mathcal{A}rg\mathcal{L}ab_2 = (\{B, C\}, \{A, D\}, \emptyset)$,

[5] In essence, the two examples can be seen as instances of floating conclusions in the sense of [14, 19].
[6] The situation where the same formalization has two different interpretations with opposite desirable outcomes called a *mirror example* in [7].

$\mathcal{A}rg\mathcal{L}ab_3 = (\emptyset, \emptyset, \{A, B, C, D\})$.
Using the function $\mathcal{A}rg\mathcal{L}ab2\mathcal{C}onc\mathcal{L}ab$, one can then identify three different complete conclusion labellings (where a is the conclusion of A and b is the conclusion of B).

$\mathcal{C}onc\mathcal{L}ab_1 = (\{a, c\}, \{b\}, \emptyset)$,
$\mathcal{C}onc\mathcal{L}ab_2 = (\{b, c\}, \{a\}, \emptyset)$,
$\mathcal{C}onc\mathcal{L}ab_3 = (\emptyset, \emptyset, \{a, b, c\})$.

Hence, the justification status of conclusion c is $\{\text{in}, \text{undec}\}$ (weak accept), even though the two arguments that produce conclusion c (C and D) each have a justification status $\{\text{in}, \text{out}, \text{undec}\}$ (undetermined borderline). Thus, our approach of justification statuses makes floating conclusion *weakly accepted* but not *strongly accepted*.

Whether or not weak accept is sufficient to endorse a conclusion depends on the particular domain of reasoning. In criminal law, for instance, what matters is the suspect should be guilty beyond reasonable doubt. This can be interpreted as requiring that in every reasonable position that one can take given the information that is available (that is, in every complete labelling of the argumentation framework) the fact that the suspect is guilty is a valid conclusion (that is, there exists an accepted argument (labelled in) for the conclusion that the suspect is guilty). This condition is not fulfilled in the above example. Therefore the conclusion that the suspect is guilty should not be endured, at least not from legal perspective.

Whether or not a conclusion is endorsed depends to a great extent on the proof standard ([2]) being applied. When deciding whether to take a friend to the ice rink or the cinema (Example 1), weak accept might be sufficient. However, when deciding whether or not to put someone in jail (Example 2), one may really require strong accept before doing so. In our approach, the system simply states how strongly each statement is accepted (or rejected). And it is up to the user to decide whether this level is sufficient to act upon. The general attitude of the labellings approach is: "we report, you decide".

Appendix

An ideal set in the sense of [13] is an admissible set that is a subset of each preferred extension. It has been obtained that one can also describe an ideal set as an admissible set that is not attacked by any admissible set (Theorem 3.2 of [13]). This clears the way for proving the following lemma (which is in essence point 5 of Proposition 3).

Lemma 1 Let (Ar, att) be an argumentation framework and $\mathcal{A}rgs \subseteq Ar$. $\mathcal{A}rgs$ is an admissible set that is not attacked by any admissible set iff $\mathcal{A}rgs$ is an admissible subset of $\{A \mid \mathcal{JS}(A) = \{\text{in}\}\} \cup \{A \mid \mathcal{JS}(A) = \{\text{in}, \text{undec}\}\}$

Proof "$\Rightarrow$": Let $\mathcal{A}rgs$ be an admissible set that is not attacked by any admissible set. Let $A \in \mathcal{A}rgs$. From the fact that A is in an admissible set (and therefore also on a complete extension) it follows that A is labelled in in at least one complete labelling. From the fact that $\mathcal{A}rgs$ is not attacked by any admissible set, it follows that A is not attacked by any preferred extension and therefore not attacked by any complete extension. Hence, A is not labelled out by any complete labelling. This, together with the earlier observed fact that A is labelled in by at least one complete labelling implies that $A \in \{A \mid \mathcal{JS}(A) = \{\text{in}\}\} \cup \{A \mid \mathcal{JS}(A) = \{\text{in}, \text{undec}\}\}$.
"$\Leftarrow$": Let $\mathcal{A}rgs$ be an admissible subset of $\{A \mid \mathcal{JS}(A) = \{\text{in}\}\} \cup \{A \mid \mathcal{JS}(A) = \{\text{in}, \text{undec}\}\}$. Suppose that $\mathcal{A}rgs$ is attacked by an admissible set. That is, there is an argument $A \in \mathcal{A}rgs$ that is attacked by an admissible set. Then A is also attacked by a complete extension (since every admissible set is contained in a preferred extension, which is also a complete extension). This means that A is labelled out in at least one complete labelling. So $A \notin \{A \mid \mathcal{JS}(A) = \{\text{in}\}\} \cup \{A \mid \mathcal{JS}(A) = \{\text{in}, \text{undec}\}\}$. Contradiction. $\square$

So our labelling-based approach for defining justification statuses not only allows us to identify whether an argument is accepted according to grounded or credulous preferred semantics, it also helps to identify whether an argument is accepted according to ideal semantics. It is in an ideal set iff one can build an admissible set around it that consists only of strongly or weakly accepted arguments.

References

[1] L. Amgoud, L. Bodenstaff, M. Caminada, P. McBurney, S. Parsons, H. Prakken, J. van Veenen, and G. A. W. Vreeswijk, 2006, *Final Review and Report on Formal Argumentation System*, ASPIC Consortium.

[2] K. Atkinson and T. Bench-Capon, 2007, " Argumentation and standards of proof", in *ICAIL' 07: Proceedings of the 11th International Conference on Artificial Intelligence and Law*, ACM Press, pp. 107-116.

[3] P. Baroni and M. Giacomin, 2007, "Comparing argumentation semantics with respect to skepticism", in *Proc. ECSQARU 2007*, pp. 210-221.

[4] M. Caminada and L. Amgoud, 2007, "On the evaluation of argumentation formalisms", *Artificial Intelligence*, 171(5-6): 286-310.

[5] M. Caminada and D. Gabby, 2009, "A logical account of formal argumentation", *Studia Logica*, 93(2-3): 109-145 .

[6] M. Caminada and Y. Wu. 2009, "An argument game for stable semantics", *Logic Journal of the IGPL*, 17(1): 77-90.

[7] M. Caminada, 2004, *For the sake of the Argument. Explorations into argument-based reasoning*, Doctoral dissertation, Free University Amsterdam.

[8] M. Caminada, 2006,"On the issue of reinstatement in argumentation", *Technical Report UU-CS-2006-023*, Institute of Information and Computing Sciences, Utrecht University.

[9] M. Caminada, 2006, "On the issue of reinstatement in argumentation", in M. Fischer, W. van der Hoek, B. Konev, and A. Lisitsa (eds.), *Logics in Artificial Intelligence, 10th European Conference*, JELIA 2006, Springer, pp. 111-123.

[10] M. Caminada, 2006, “Semi-stable semantics”, in P. E. Dunne and T. J. M. Bench-Capon (eds.), *Computational Models of Argument; Proceedings of COMMA 2006*, IOS Press, pp. 121-130.

[11] C. Cayrol, S. Doutre, and J. Mengin, 2003, “On decision problems related to the preferred semantics for argumentation frameworks”, *Journal of Logic and Computation*, 13(3): 377-403.

[12] P. M. Dung, 1995, “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games”, *Artificial Intelligence*, 77: 321-357.

[13] P. M. Dung, P. Mancarella, and F. Toni, 2007, “Computing ideal sceptical argumentation”, *Artificial Intelligence*, 171(10-15): 642-674.

[14] J. Horty, 2002, “Skepticism and floating conclusions”, *Artificial Intelligence*, 135: 55-72.

[15] H. Jakobovits and D. Vermeir, 1999, “Robust semantics for argumentation frameworks”, *Journal of Logic and Computation*, 9(2): 215-261.

[16] S. Modgil and M. Caminada, 2009, “Proof theories and algorithms for abstract argumentation frameworks”, in I. Rahwan and G. Simari (eds.), *Argumentation in Artificial Intelligence*, pp. 105-129.

[17] S. Modgil and H. Prakken, 2010, “Reasoning about preferences in structured extended argumentation frameworks”, in M. Giacomin P. Baroni, F. Cerutti and G.R. Simari (eds.), *Computational Models of Argument; Proceedings of COMMA 2006*, IOS Press, pp. 347-358.

[18] J. L. Pollock, 1995, *Cognitive Carpentry: A Blueprint for How to Build a Person*, MIT Press, Cambridge, MA.

[19] H. Prakken, 2002, “Intuitions and the modelling of defeasible reasoning: some case studies”, in *Proceedings of the Ninth International Workshop on Nonmonotonic Reasoning (NMR-2002)*, Toulouse, France, pp. 91-99.

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

[21] B. Verheij, 1996, “Two approaches to dialectical argumentation: admissible sets and argumentation stages”, in J. J. Ch. Meyer and L. C. van der Gaag (eds.), *Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC’ 96)*, Utrecht, Utrecht University, pp. 357-368.

[22] G. A. W. Vreeswijk and H. Prakken, 2000, “ Credulous and sceptical argument games for preferred semantics”, in *Proceedings of the 7th European Workshop on Logic for Artificial Intelligence (JELIA-00), number 1919 in Springer Lecture Notes in AI,* Berlin, Springer Verlag, pp. 239-253.

[23] G. A. W. Vreeswijk, 2006, “An algorithm to compute minimally grounded and admissible defence sets in argument systems”, in P. E. Dunne and T. J. M. Bench- Capon, eds., *Computational Models of Argument; Proceedings of COMMA 2006*, IOS Press, pp. 109-120.

论证的基于标记的论据状态

吴伊宁
卢森堡大学
yining.wu@uni.lu
马丁·卡米那达
卢森堡大学
martin.caminada@uni.lu

摘　要

本文针对抽象论证体系中的论据状态给出了一种基于标记的定义方法。与基于扩张的传统方法相比，此定义能够更加细致准确的描述论据状态。同时，本文对论据状态的等级进行了定义和区分，并把该等级作为接受或驳斥论据的依据。由于该方法基于标准论证体系定义，在实现上使用了已有的基于论证的证明方法，因此，该定义与以往的抽象论证体系是完全兼容的。