A Labelling Based Justification Status of Arguments

Yining Wu and **Martin Caminada** and **Mikolaj Podlaszewski**
Faculty of Sciences, Technology and Communication
University of Luxembourg, Luxembourg

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.

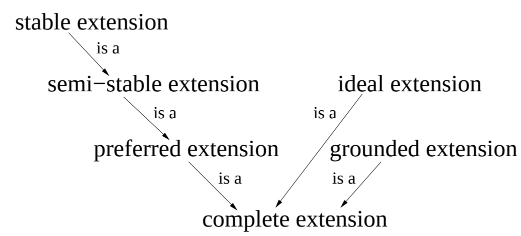

Figure 1: An overview of the different semantics

1. Introduction

The main concept in Dung's theory (Dung 1995) 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 (Dung 1995), semi-stable semantics (Caminada 2006c) or ideal semantics (Dung, Mancarella, and Toni 2007). 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 has been proved in (Caminada 2006c). 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 (Dung 1995). The ideal extension is also a complete extension (Dung, Mancarella, and Toni 2007). 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 (Pollock 1995) and of Jakobovits and Vermeir (Jakobovits and Vermeir 1999). However, for current purposes we will use the concept of complete labelling as defined by (Caminada 2006a; Caminada and Gabbay 2009).

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 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 (Caminada 2006a), 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 (Dung 1995) in the sense that it works on standard argumentation frameworks and can be implemented using existing argumentation-based proof procedures (Vreeswijk and Prakken 2000; Modgil and Caminada 2009).

The remaining part of this paper is organized as follows. We first state some preliminaries on argument semantics and argument labellings. Then we define the justification status of an argument, describe the methods for determining it and treat the related issues of computational complexity. 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 (Caminada 2006b; Caminada and Gabbay 2009) 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 ((Caminada and Gabbay 2009))**.** *A* labelling *is a function* $\mathcal{L}ab : Ar \longrightarrow \{\texttt{in}, \texttt{out}, \texttt{undec}\}$.

We write $\texttt{in}(\mathcal{L}ab)$ for $\{A \mid \mathcal{L}ab(A) = \texttt{in}\}$, $\texttt{out}(\mathcal{L}ab)$ for $\{A \mid \mathcal{L}ab(A) = \texttt{out}\}$ and $\texttt{undec}(\mathcal{L}ab)$ for $\{A \mid \mathcal{L}ab(A) = \texttt{undec}\}$. 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 $(\texttt{in}(\mathcal{L}ab), \texttt{out}(\mathcal{L}ab), \texttt{undec}(\mathcal{L}ab))$.

[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.

The idea of a *complete labelling* (Caminada 2006b; Caminada and Gabbay 2009) 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 are insufficient grounds to accept it and insufficient grounds to reject it). This is made formal in the following definition.

Definition 5 ((Caminada and Gabbay 2009))**.** *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* $\mathcal{L}ab(A) = \texttt{in}$ *then* $\forall B \in Ar : (B \ att \ A \supset \mathcal{L}ab(B) = \texttt{out})$
2. *If* $\mathcal{L}ab(A) = \texttt{out}$ *then* $\exists B \in Ar : (B \ att \ A \wedge \mathcal{L}ab(B) = \texttt{in})$.
3. *If* $\mathcal{L}ab(A) = \texttt{undec}$ *then* $\neg\forall B \in Ar : (B \ att \ A \supset \mathcal{L}ab(B) = \texttt{out})$ *and* $\neg\exists B \in Ar : (B \ att \ A \wedge \mathcal{L}ab(B) = \texttt{in})$.

As stated in (Caminada 2006b; Caminada and Gabbay 2009), complete labellings coincide with complete extensions in the sense of (Dung 1995).

Theorem 1 ((Caminada and Gabbay 2009))**.** *Let* $AF = (Ar, att)$ *be an argumentation framework.*

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

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 (Caminada 2006b; Caminada and Gabbay 2009).

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 (Dung 1995) 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 $\mathcal{JS} : Ar \rightarrow 2^{\{\mathtt{in},\mathtt{out},\mathtt{undec}\}}$ 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 $\{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$. 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 (Caminada and Gabbay 2009)). Furthermore, it is also impossible for a justification status to be $\{\mathtt{in}, \mathtt{out}\}$, because when $\mathtt{in}$ and $\mathtt{out}$ are both included in the justification status, then $\mathtt{undec}$ 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 $\mathcal{L}ab_1(A) = \mathtt{in}$ and $\mathcal{L}ab_2(A) = \mathtt{out}$ then there exists a complete labelling $\mathcal{L}ab_3$ such that $\mathcal{L}ab_3(A) = \mathtt{undec}$.*

Proof. Let $CE_1 = \mathtt{Lab2Ext}(\mathcal{L}ab_1)$ and $CE_2 = \mathtt{Lab2Ext}(\mathcal{L}ab_2)$. From Theorem 3 of (Caminada and Gabbay 2009) it follows that CE_1 and CE_2 are complete extensions of AF. Let GE be the grounded extension of AF. From (Dung 1995) it follows that GE is the intersection of all complete extensions of AF. From $\mathcal{L}ab_2(A) = \mathtt{out}$, it follows that $A \notin CE_2$ which implies that $A \notin GE$. From $\mathcal{L}ab_1(A) = \mathtt{in}$, it follows that $\forall B \in Ar.(B\, att A \supset \mathcal{L}ab_1(B) = \mathtt{out})$. Therefore, $\forall B \in Ar.(B\, att A \supset B \notin GE)$. So $A \notin GE^+$. Let $\mathcal{L}ab_3 = \mathtt{Ext2Lab}(GE)$. 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 $\mathcal{L}ab_3(A) = \mathtt{undec}$. □

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

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

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

Proof. "⇒": Suppose $\mathcal{JS}(A) = \{\mathtt{in}\}$. Then A is labelled $\mathtt{in}$ 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 $\{\mathtt{out}\}$.

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

Proof. "⇒": Suppose $\mathcal{JS}(A) = \{\mathtt{out}\}$. Then A is labelled $\mathtt{out}$ by every complete labelling (Definition 6). So in every complete labelling, there exists at least one atttacker of A that is labelled $\mathtt{in}$ 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 $\{\mathtt{undec}\}$.

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

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

Proof. "⇒": Suppose $\mathcal{JS}(A) = \{\mathtt{undec}\}$. Then it holds that (1) A is not labelled $\mathtt{in}$ by any complete labelling and (2) A is not labelled $\mathtt{out}$ 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 $\mathtt{in}$ 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 $\mathtt{undec}$ 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 $\mathtt{in}$ 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 $\mathtt{in}$ by any complete labelling (Theorem 1) so A is not labelled $\mathtt{out}$ by any complete labelling (Definition 5). This, together with the earlier observed fact that A is not labelled $\mathtt{in}$ by any complete labelling implies that A is labelled $\mathtt{undec}$ 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 $\mathcal{JS} = \{\mathtt{undec}\}$. □

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

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

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

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

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

Proof. "⇒": Suppose $\mathcal{JS}(A) = \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$. 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.
"⇐": 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.

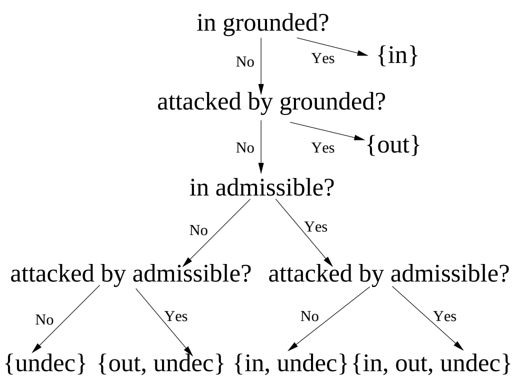

Figure 2: determining the justification status of an argument

4. An Implementation

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

Given an argumentation framework as input, the program implements two main commands: `question` and `discuss`. The command "`question` argument" gives the justification status of an argument and the command "`discuss` argument" allows the user to critically discuss this justification status.

In order to determine the justification status of an argument (the `question` command) our implementation follows the procedure of Figure 2. To determine whether an argument is in the grounded extension, the algorithm described in (Modgil and Caminada 2009) 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 (Vreeswijk and Prakken 2000; Caminada and Wu 2009) 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 is able to defend its answer (the `discuss` command) by entering a discussion game with the user. The

[2]Available at http://icr.uni.lu/∼yining

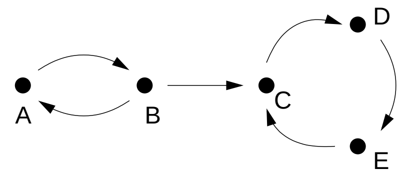

Figure 3: Argumentation Framework

precise discussion game depends on the justification status of the argument, as well as on which part of this justification status the user disagrees with. Since the justification status as calculated by the software can be assumed to be correct, the discussion will be such that the computer will always win from the user. After all, the aim of the `discuss` command is to convince the user of the correctness of the justification status as computed by the software.

Example 1. *Let AF be the argumentation framework in Figure 3.*

If the user inputs: "`question` A" then the program will give the result "$\{\texttt{in}, \texttt{out}, \texttt{undec}\}$"

If the user does not agree with the result, the program will ask which one of the followings the user does not agree with.

1. *A can be accepted.*
2. *A can be rejected.*

If the user disagrees with (1) then the admissible discussion game (Vreeswijk and Prakken 2000; Modgil and Caminada 2009) in favor of argument A will be started. If the user disagrees with (2) then an admissible discussion game (Vreeswijk and Prakken 2000; Modgil and Caminada 2009) in favor of an attacker of A that is in an admissible set will be started.

5. Computational Complexity

We now examine the computational complexity of the various problems related to assigning labelling-based justification statuses. To determine whether the justification status of an argument is $\{\texttt{in}\}$ one has to determine whether it is an element of the grounded extension (Theorem 3) which is known to be P (Dung 1995). To determine whether the justification status of an argument is $\{\texttt{out}\}$ one has to determine whether it is attacked by the grounded extension (Theorem 4) which means determining membership of the grounded extension for each of its attackers. Since there are at most n attackers (where n is the number of arguments in the argumentation framework) the complexity is n times P, which is P itself. To determine whether an argument has the justification status $\{\texttt{undec}\}$ one has to determine whether it is in an admissible set and whether it is attacked by an admissible set (Theorem 5). Determining whether it is in an admissible set is known to be NP-complete (Dimopoulos and Torres 1996). Determining whether it is attacked by an admissible set is therefore n times NP-complete, which is NP-complete itself. Using similar reasoning, one can obtain that determining whether an argument has a justification status $\{\texttt{in}, \texttt{undec}\}$ is NP-complete, $\{\texttt{out}, \texttt{undec}\}$ is

NP-complete and $\{\texttt{in}, \texttt{out}, \texttt{undec}\}$ is NP-complete. Since determining each individual justification status has a computational complexity of NP-complete or below, the overall worst-case complexity of determining a justification status is NP-complete. This puts the approach of labelling-based justification statuses in the same class as the more traditional approach of credulous preferred.

6. Discussion and Related Work

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, and examined the issue of computational complexity.

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

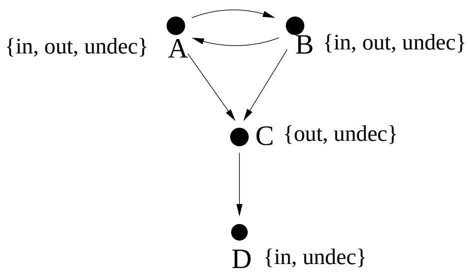

Figure 4: An example

Take the example in figure 4. Grounded semantics treats all arguments (A, B, C and D) the same (they are not 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 5, 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.

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

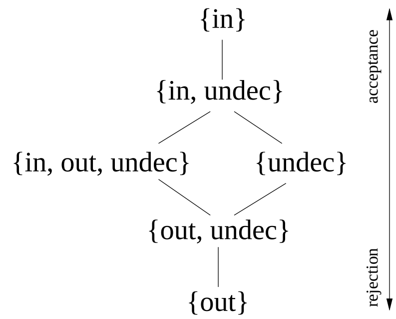

Figure 5: The hierarchy of justification statuses

grounded semantics (Dung 1995), credulous preferred semantics (Vreeswijk and Prakken 2000), sceptical preferred semantics (Cayrol, Doutre, and Mengin 2003), semi-stable semantics (Caminada 2006c) and ideal semantics (Dung, Mancarella, and Toni 2007).

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 $\{\texttt{undec}\}$ (determined borderline), which means only $\{\texttt{in}\}$ (strong accept) and $\{\texttt{in}, \texttt{out}\}$ (weak accept) remain. The validity of point 4 follows from Theorem 5 of (Caminada 2006c). 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 (Baroni and Giacomin 2007). However, in (Baroni and Giacomin 2007) 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 (Vreeswijk and Prakken 2000; Caminada and Wu 2009) to determine membership of an admissible set, and the discussion game of (Prakken and Sartor 1997;

Modgil and Caminada 2009) to determine membership of the grounded extension. An alternative would be to use the algorithm of (Vreeswijk 2006), 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.

Appendix

An ideal set in the sense of (Dung, Mancarella, and Toni 2007) 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 (Dung, Mancarella, and Toni 2007)). 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 subet of* $\{A \mid \mathcal{JS}(A) = \{\mathtt{in}\}\} \cup \{A \mid \mathcal{JS}(A) = \{\mathtt{in}, \mathtt{undec}\}\}$

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 $A \in \{A \mid \mathcal{JS}(A) = \{\mathtt{in}\}\} \cup \{A \mid \mathcal{JS}(A) = \{\mathtt{in}, \mathtt{undec}\}\}$.
"$\Leftarrow$": Let $\mathcal{A}rgs$ be an admissible subset of $\{A \mid \mathcal{JS}(A) = \{\mathtt{in}\}\} \cup \{A \mid \mathcal{JS}(A) = \{\mathtt{in}, \mathtt{undec}\}\}$. 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 $A \notin \{A \mid \mathcal{JS}(A) = \{\mathtt{in}\}\} \cup \{A \mid \mathcal{JS}(A) = \{\mathtt{in}, \mathtt{undec}\}\}$. 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

Baroni, P., and Giacomin, M. 2007. Comparing argumentation semantics with respect to skepticism. In *Proc. ECSQARU 2007*, 210–221.

Caminada, M., and Gabbay, D. 2009. A logical account of formal argumentation. *Studia Logica* 93(2-3):109–145. Special issue: New Ideas in Argumentation Theory.

Caminada, M., and Wu, Y. 2009. An argument game of stable semantics. *Logic Journal of IGPL* 17(1):77–90.

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

Caminada, M. 2006b. On the issue of reinstatement in argumentation. In Fischer, M.; van der Hoek, W.; Konev, B.; and Lisitsa, A., eds., *Logics in Artificial Intelligence; 10th European Conference, JELIA 2006*, 111–123. Springer. LNAI 4160.

Caminada, M. 2006c. Semi-stable semantics. In Dunne, P., and Bench-Capon, T., eds., *Computational Models of Argument; Proceedings of COMMA 2006*, 121–130. IOS Press.

Cayrol, C.; Doutre, S.; and Mengin, J. 2003. On Decision Problems related to the preferred semantics for argumentation frameworks. *Journal of Logic and Computation* 13(3):377–403.

Dimopoulos, Y., and Torres, A. 1996. Graph theoretical structures in logic programs and default theories. *Theoretical Computer Science* 170:209–244.

Dung, P. M.; Mancarella, P.; and Toni, F. 2007. Computing ideal sceptical argumentation. *Artificial Intelligence* 171(10-15):642–674.

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

Jakobovits, H., and Vermeir, D. 1999. Robust semantics for argumentation frameworks. *Journal of logic and computation* 9(2):215–261.

Modgil, S., and Caminada, M. 2009. Proof theories and algorithms for abstract argumentation frameworks. In Rahwan, I., and Simari, G., eds., *Argumentation in Artificial Intelligence*. Springer. 105–129.

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

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

Vreeswijk, G. A. W., and Prakken, H. 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, 239–253. Berlin: Springer Verlag.

Vreeswijk, G. 2006. An algorithm to compute minimally grounded and admissible defence sets in argument systems. In Dunne, P., and Bench-Capon, T., eds., *Computational Models of Argument; Proceedings of COMMA 2006*, 109–120. IOS.