A Gentle Introduction to Argumentation Semantics

Martin Caminada
University of Luxembourg

Summer 2008

Abstract

This document presents an overview of some of the standard semantics for formal argumentation, including Dung's notions of grounded, preferred, complete and stable semantics, as well as newer notions like Caminada's semi-stable semantics and Dung, Mancarella and Toni's ideal semantics. These semantics will be treated both in their original extension-based form, as well as in the form of argument labellings. Our treatment includes a sketch of few algorithms for skeptical as well as for the credulous approach to argumentation.

1 Introduction and Overview

In this document, one of the basic building blocks of argumentation theory is treated: the argument based semantics. The idea is, roughly, that given a set of arguments where some arguments defeat others, one wants to determine which arguments can ultimately be accepted. To determine whether or not an argument can be accepted, it is not sufficient to merely look at its defeaters; what also matters is whether the defeaters are defeated themselves. Consider the following example (taken from [15]):

> Suppose Ralph normally goes fishing on Sundays, but on the Sunday which is Mother's day, he typically visits his parents. Furthermore, in the spring of each leap year his parents take a vacation, so that they cannot be visited.

Suppose it is Sunday, Mother's day and a leap year. Then, one can formulate three arguments related to whether Ralph goes fishing or not:

Argument A:
Ralph goes fishing because it is Sunday.

Argument B:
Ralph does not go fishing because it is Mother's day, so he visits his parents.

Argument C:
Ralph cannot visit his parents, because it is a leap year, so they are on vacation.

We say that an argument B *defeats* argument A iff B is a reason against A.

If one abstracts from the internal structure of an argument, as well as from the reasons *why* they defeat each other, what is left is called an *argumentation framework*. An argumentation framework simply consists of a set of (abstract) arguments and a binary defeat relation between these arguments.

Definition 1. *An* argumentation framework *is a pair* $AF = (Ar, def)$ *where* Ar *is a set of arguments and* $def \subseteq Ar \times Ar$*. We say that* A defeats B *iff* $(A, B) \in def$.

An argumentation framework essentially specifies a directed graph in which arguments are represented as nodes and the defeat relation is represented as arrows. For instance, the argumentation framework of the "Ralph goes fishing" example is shown in figure 1.

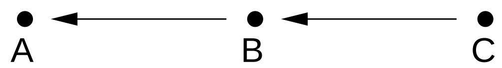

Figure 1: Arguments and reinstatement (AF_1).

An interesting question is which of the arguments should ultimately be accepted. Since A is defeated by B, it would at first seem that A should not be accepted, since it has a counterargument. If one looks further, however, it turns out that this counterargument (B) is itself defeated by an argument (C) that is not defeated by anything. So, at least C should be accepted. But if C is accepted, then B is ultimately rejected and does not form a reason against A anymore. Therefore, A should also be accepted.

In figure 1, we say that argument C *reinstates* argument A.[1] Because of the issue of reinstatement, it is necessary to state some formal criterion that takes an argumentation framework and determines which of the arguments can be accepted and which cannot. Such a criterion is called an *argument based semantics*, or simply *semantics*. The idea of a semantics is, given an argumentation framework, to specify zero or more sets of acceptable arguments. These sets are also called *argument based extensions*, or simply *extensions*.

Various argument based semantics have been stated during recent years. In this document we provide an overview of the most common ones, as well as various others that are worthwile mentioning.

2 Argument Labellings

The issue of argument based semantics is perhaps best understood using the approach of labelling each argument either `in`, `out` or `undec` according to the following conditions:

[1] Although reinstatement is directly or indirectly implemented by many systems for defeasible reasoning, it has been mentioned that in some cases reinstatement can also cause problems; see for instance [13].

- an argument is labelled `in` iff all its defeaters are labelled `out`, and
- an argument is labelled `out` iff all it has at least one defeater that is labelled `in`.

Informally, labelling an argument `in` means that one has accepted the argument, labelling an argument `out` means that one has rejected the argument and labelling the argument `undec` means that one abstains from taking a position on whether the argument is accepted or rejected. Formally, the labelling approach is described in Definition 2.

Definition 2. *Let* (Ar, def) *be an argumentation framework and* $\mathcal{L}ab : Ar \longrightarrow \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$ *be a total function. We say that* $\mathcal{L}ab$ *is a* complete labelling *iff it satisfies the following:*

- $\forall A \in Ar : (\mathcal{L}ab(A) = \mathtt{out} \equiv \exists B \in Ar : (B \; def \; A \wedge \mathcal{L}ab(B) = \mathtt{in}))$ *and*
- $\forall A \in Ar : (\mathcal{L}ab(A) = \mathtt{in} \equiv \forall B \in Ar : (B \; def \; A \supset \mathcal{L}ab(B) = \mathtt{out}))$.

We will sometimes write $\mathtt{in}(\mathcal{L}ab)$ for the set of arguments labelled `in` by $\mathcal{L}ab$, $\mathtt{out}(\mathcal{L}ab)$ for the set of arguments labelled `out` by $\mathcal{L}ab$ and $\mathtt{undec}(\mathcal{L}ab)$ for the set of arguments labelled `undec` by $\mathcal{L}ab$.

It is interesting to examine the argumentation framework of Figure 1 using the labelling approach. For argument C it holds that all its defeaters are labelled `out` (this is trivial, since C does not have any defeaters). Therefore, C must be labelled `in`. Argument B now has a defeater that is labelled `in`. Therefore, B must be labelled `out`. For argument A it holds that all its defeaters are labelled `out`, so that A itself must be labelled `in`. The overall result is hence a labelling $\mathcal{L}ab$ with $\mathcal{L}ab(A) = \mathtt{in}$, $\mathcal{L}ab(B) = \mathtt{out}$ and $\mathcal{L}ab(C) = \mathtt{out}$.

Another example that is worthwile to examine using the labelling approach is the following:

Argument A: Bert says that Ernie is unreliable, therefore everything that Ernie says cannot be relied on.

Argument B: Ernie says that Elmo is unreliable, therefore everything that Elmo says cannot be relied on.

Argument C: Elmo says that Bert is unreliable, therefore everthing that Bert says cannot be relied on.

This example yields an argumentation framework (Ar, def) where $Ar = \{A, B, C\}$ and $def = \{(A, B), (B, C), (C, A)\}$. The argumentation framework is depicted in figure 2.

For figure 2, there exists only a single labelling $\mathcal{L}ab$, with $\mathcal{L}ab(A) = \mathtt{undec}$, $\mathcal{L}ab(B) = \mathtt{undec}$ and $\mathcal{L}ab(C) = \mathtt{undec}$. That is, it is not possible to label A, B or C with `in` or `out`. This can be seen as follows. Suppose that there is a labelling where A is labelled `in`. Then all defeaters of A should be labelled `out`, which means that C should be labelled `out`. The fact that C is labelled `out` means that C should have a defeater that is labelled `in`, which implies that B is labelled `in`. The fact that B is labelled `in` means that all defeaters of B must be labelled `out`, which implies that A is labelled `out`. Contradiction.

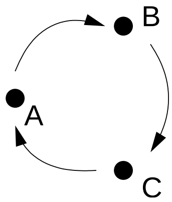

Figure 2: Bert, Ernie and Elmo

Similarly, assume that there is a labelling where A is labelled `out`. Then A must have a defeater that is labelled `in`, which implies that C is labelled `in`. The fact that C is labelled `in` means that every defeater of C must be labelled `out`, which implies that B is labelled `out`. The fact that B is labelled `out` means that B has a defeater that is labelled `in`, which implies that A is labelled `in`. Contradiction.

A labelling of an argumentation framework may not be unique. That is, there exist argumentation frameworks for which there is more than one labelling. An example of this is the well-known Nixon diamond:

Argument A: Nixon is a pacifist because be is a quaker.

Argument B: Nixon is not a pacifict because he is republican.

This example yields an argumentation framework (Ar, def) where $Ar = \{A, B\}$ and $def = \{(A, B), (B, A)\}$. The argumentation framework is graphically depicted in figure 3.

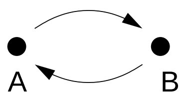

Figure 3: The Nixon diamond

For figure 3, three labellings exist:

1. $\mathcal{L}ab_1$ with $\mathcal{L}ab_1(A) =$ `in` and $\mathcal{L}ab_1(B) =$ `out`,
2. $\mathcal{L}ab_2$ with $\mathcal{L}ab_2(A) =$ `out` and $\mathcal{L}ab_2(B) =$ `in`, and
3. $\mathcal{L}ab_3$ with $\mathcal{L}ab_3(A) =$ `undec` and $\mathcal{L}ab_3(B) =$ `undec`.

In essence, one can interpret each labelling as a reasonable position someone can take regarding argument reinstatement. In the case of figure 3, there exist three such positions, one in which A is believed and B is disbelieved, one in which B is believed and A is disbelieved, and one in which a position on both A and B is abstained from. In the case of figure 2, one can only abstain from any position on A, B and C. In the case of figure 1, however, one *cannot* abstain from a position on C, since it has no defeaters and therefore must be `in`. For similar reasons, one also cannot reasonably abstain from taking a position on the arguments B and A. Thus, it can be observed that while the introduction of `undec` allows for argumentation frameworks have at least one labelling — something they would not have without `undec` — the label `undec` does not serve as a "wildcard" that one can always apply to any arbitrary argument.

Exercise 1. *Give all complete labellings of:*

(a) *figure 4*

(b) *figure 5*

(c) *figure 6*

(d) *figure 7*

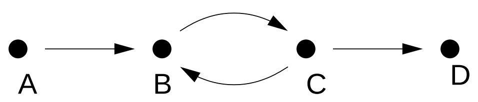

Figure 4: Argumentation framework I

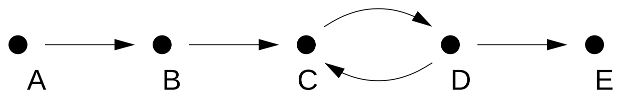

Figure 5: Argumentation framework II

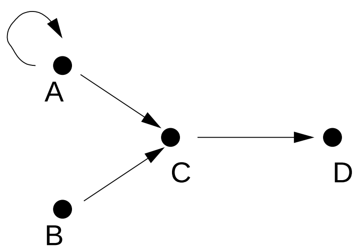

Figure 6: Argumentation framework III

3 Extension Based Semantics

The approach of complete labellings of argumentation frameweoks, although intuitive, has received only limited following in most argumentation research. The issue of whether an argument can be regarded as overall justified is usually handled my means of the concept of an *extension* of arguments. In this section, we treat five definitions according to which extensions of arguments can be determined. We show how these definitions are related not only to each other, but also to the previously discussed notion of complete labelling.

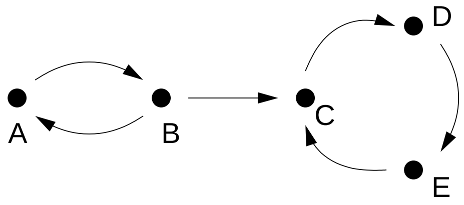

Figure 7: Argumentation framework IV

3.1 Complete Semantics

The first extension-based argument semantics to be discussed is *complete semantics* [9]. We first discuss some preliminary notions.

Definition 3. *Let (Ar, def) be an argumentation framework and let $A \in Ar$ and $\mathcal{A}rgs \subseteq Ar$.*
We define A^+ as $\{B \mid A\,def\,B\}$ and $\mathcal{A}rgs^+$ as $\{B \mid A\,def\,B$ for some $A \in \mathcal{A}rgs\}$.
We define A^- as $\{B \mid B\,def\,A\}$ and $\mathcal{A}rgs^-$ as $\{B \mid B\,def\,A$ for some $A \in \mathcal{A}rgs\}$.

A set of arguments is called *conflict-free* iff it does not contain any arguments A and B such that A defeats B.

Definition 4. *Let (Ar, def) be an argumentation framework and let $\mathcal{A}rgs \subseteq Ar$. $\mathcal{A}rgs$ is said to be* conflict-free *iff $\mathcal{A}rgs \cap \mathcal{A}rgs^+ = \emptyset$.*

A set of arguments is said to *defend* an argument C iff each defeater of C is defeated by an argument in $\mathcal{A}rgs$. This situation is depicted in Figure 8.

Definition 5. *Let (Ar, def) be an argumentation framework, $\mathcal{A}rgs \subseteq Ar$ and $B \in Ar$. $\mathcal{A}rgs$ is said to* defend B *iff $B^- \subseteq \mathcal{A}rgs^+$.*

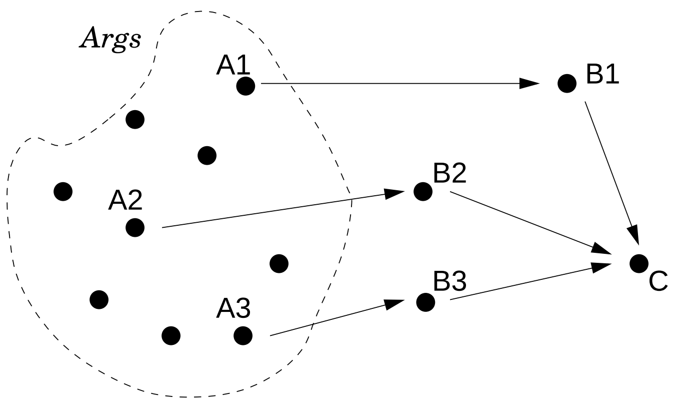

Figure 8: $\mathcal{A}rgs$ defends argument C

Exercise 2. *In figure 4:*

(a) *does* $\{A\}$ *defend* C*?*

(b) *does* $\{C\}$ *defend* C*?*

(c) *does* $\{B\}$ *defend* C*?*

The function F yields the arguments defended by a given set of arguments. That is, is specifies the set of arguments that are acceptable in the sense of [9].

Definition 6. *Let* (Ar, def) *be an argumentation framework and* $\mathcal{A}rgs \subseteq Ar$*. We introduce a function* $F : 2^{Ar} \longrightarrow 2^{Ar}$ *such that* $F(\mathcal{A}rgs) = \{A \mid A \text{ is defended by } \mathcal{A}rgs\}$*.*

Exercise 3. *In figure 7:*

(a) *give* $F(\{A\})$

(b) *give* $F(\{B\})$

(c) *give* $F(\{B, D\})$

Definition 7. *Let* (Ar, def) *be an argumentation framework and* $\mathcal{A}rgs$ *be a conflict-free set of arguments.* $\mathcal{A}rgs$ *is said to be a a* complete extension *iff* $\mathcal{A}rgs = F(\mathcal{A}rgs)$*.*

In figure 1, there exists just one complete extension: $\{A, C\}$. It is a complete extension since it is conflict-free and defends exactly itself. Notice that $\{A, B, C\}$ is also a fixpoint of F, but is not a complete extension since it is not conflict-free. In figure 3, there exists three complete extensions: $\{A\}$, $\{B\}$ and $\emptyset$. In figure 2, there exists just one complete extension: $\emptyset$.

Exercise 4. *Give all complete extensions of:*

(a) *figure 4*

(b) *figure 5*

(c) *figure 6*

(d) *figure 7*

There exists a strong connection between complete extensions and complete labellings, as has been mentioned in [6]. For every complete labelling, the set of `in` labelled arguments forms a complete extension. This is because the set of `in` labelled arguments is conflict-free and defends exactly itself. Furthermore, each complete extension $\mathcal{A}rgs$ is associated with a unique complete labelling in which all arguments in $\mathcal{A}rgs$ are labelled `in`, all arguments in $\mathcal{A}rgs^+$ are labelled `out` and all other arguments are labelled `undec`. Precise details can be found in [6]. For now, it suffices to say that complete extensions and complete labellings are basically the same things.

3.2 Grounded Semantics

Complete semantics has as a fundamental property that more than one complete extension may exist. In some situations it can have advantages to apply an argument-based semantics that is guaranteed to yield exactly one extension. Grounded semantics, a concept that has its root in Pollock's OSCAR [17] and the well-founded semantics of logic programming [21], is such a semantics. The idea is, roughly, to specifically select the complete labelling $\mathcal{L}ab$ in which $\texttt{in}(\mathcal{L}ab)$ (the set of `in`-labelled arguments) is minimal (with respect to set-inclusion)[2].

Since F is a monotonic function (that is, if $\mathcal{A}rgs \subseteq \mathcal{A}rgs'$ then $F(\mathcal{A}rgs) \subseteq F(\mathcal{A}rgs')$), it is guarenteed to have a smallest fixpoint by the Knaster-Tarski theorem. This implies that the grounded extension is well-defined.

Definition 8. *Let (Ar, def) be an argumentation framework. The grounded extension is the minimal fixpoint of F.*

The grounded extension is also conflict-free, for reasons explained in [7]. This means that the grounded extension is actually the smallest complete extension.

From the perspective of complete labellings, the grounded extension coincides with the complete labelling in which `in` is minimized (with respect to set-inclusion). This also means that in such a labelling `out` is also minimized (this is because by getting less `in` labelled arguments, one can only get less or equal `out` labelled arguments). Since both `in` and `out` are minimized, this means that `undec` is maximized. Thus, grounded semantics basically boils down to chosing the complete labelling with minimal `in`, minimal `out` and maximal `undec`.

In figure 1, the grounded extension is $\{A, C\}$. In figure 3 and figure 2, the grounded extension is $\emptyset$.

Exercise 5. *Give the grounded extension of:*

(a) *figure 4*

(b) *figure 5*

(c) *figure 6*

(d) *figure 7*

3.3 Preferred Semantics

Grounded semantics has as advantage that there always exists exactly one grounded extension. A potential disadvantage, however, is that grounded semantics is very sceptical approach. Some people have argued that what is needed is a more credulous approach. Preferred semantics is an example of such. The idea of preferred semantics is, roughly, that instead of maximizing `undec`, one maximizes `in` (and therefore also `out`).

A central notion in preferred semantics is that of admissibility. A set of arguments is admissible iff it is conflict-free and defends at least itself.

Definition 9. *Let (Ar, def) be an argumentation framework and $\mathcal{A}rgs \subseteq Ar$. $\mathcal{A}rgs$ is said to be* admissible *iff $\mathcal{A}rgs$ is conflict-free and $\mathcal{A}rgs \subseteq F(\mathcal{A}rgs)$.*

[2] In this document, whenever we mention sets that are minimal or maximal, we refer to minimality or maximality with respect to the partial ordering defined by set-inclusion.

For instance, in figure 1, $\{C\}$ is an admissible set, just like $\{A, C\}$. The set $\{B\}$, however is not admissible because it does not defend itself against C. The set $\{A\}$ is also not admissible, as it does not defend itself against B. In figure 3, $\{A\}$ and $\{B\}$ are admissible sets. The set $\{A, B\}$, however, is not admissible, since it is not conflict-free. In figure 2, the sets $\{A\}$, $\{B\}$ and $\{C\}$ are not admissible as they do not defend themselves against C, A and B, respectively.

As the empty set is conflict-free and trivially defends itself against each of its defeaters (this is because the empty set does not have any defeaters), the empty set is admissible in every argumentation framework.

Exercise 6. *Are the following sets of argumements admissible:*

(a) $\{A\}$ *in figure 4*

(b) $\{C\}$ *in figure 5*

(c) $\{A\}$ *in figure 6*

(d) $\{A, C, D\}$ *in figure 7*

The concept of a preferred extension can then be defined as follows.

Definition 10. *Let* (Ar, def) *be an argumentation framework and* $\mathcal{A}rgs \subseteq Ar$*.* $\mathcal{A}rgs$ *is said to be a* preferred extension *iff* $\mathcal{A}rgs$ *is a maximal (with respect to set-inclusion) admissible set.*

In figure 1, there exists just one preferred extension: $\{A, C\}$. In figure 3, there exist two preferred extensions: $\{A\}$ and $\{B\}$. In figure 2, there only exists one preferred extension: the empty set.

Exercise 7. *Give the preferred extensions of:*

(a) *figure 4*

(b) *figure 5*

(c) *figure 6*

(d) *figure 7*

For any argumentation framework, there exists at least one preferred extension. To prove this, we have to deal with the situation that, for an argumentation framework with an infinite set of arguments, there exist infinitly many admissible sets, and there exists at least one sequence of admissible sets $\mathcal{A}rgs_1, \mathcal{A}rgs_2, \mathcal{A}rgs_3, \ldots$ such that $\mathcal{A}rgs_{i+1} \supsetneq \mathcal{A}rgs_i$ $(i \geq 1)$. This would leave open the possibility that a preferred extension does not exist because there is no *maximal* admissible set, because for any admissible set one could always find a greater one, without there being a global maximum.

To deal with this situation, it can be mentioned that the union of an ever increasing sequence of admissible sets $\mathcal{A}rgs_1, \mathcal{A}rgs_2, \mathcal{A}rgs_3, \ldots$ is again an admissible set (say: $\mathcal{A}rgs'$). This is because this union is conflict-free (otherwise at least one $\mathcal{A}rgs_i$ $(i \geq 1)$ would not be conflict-free) and defends all its elements (otherwise at least one $\mathcal{A}rgs_i$ ($i \geq 1$) would not defend all its arguments). Using this observation, one can then apply Zorn's Lemma, which can be stated as follows: "Every non-empty partially ordered set (S) of which every totally ordered subset (T) has an upper bound contains at least one maximal element." Let S be the set of all admissible sets, where the admissible sets are ordered according to the subset relation. As every totally ordered subset T (that is: every sequence of increasing admissible sets) has an upper bound (that is: its union), one can apply Zorn's Lemma and obtain the existence of at least one maximal element (the preferred extension). Although not explicitly mentioned in [9], this is in fact the reason why there always exists a preferred extension.

It has been proved that every preferred extension is also a complete extension [9][Theorem 25]. From the perspective of complete labellings, preferred extensions coincide with those tabellings in which `in` is maximal and `out` is maximal.

3.4 Stable Semantics

Stable semantics is one of the oldest argument-based semantics available. The origins of stable semantics go back to default logic [20] and the stable model semantics of logic programming [12]. In terms of complete labellings, the idea of stable semantics is only to take into account those labellings not containing `undec`.

A set of arguments is a stable extension iff it defeats each argument which does not belong to it. We use $\mathcal{A}rgs^+$ as a shorthand for $\{A \mid A \text{ is defeated by an argument in } \mathcal{A}rgs\}$.

Definition 11. *Let (Ar, def) be an argumentation framework and $\mathcal{A}rgs \subseteq Ar$. $\mathcal{A}rgs$ is a stable extension iff $\mathcal{A}rgs^+ = Ar \backslash \mathcal{A}rgs$.*

From the above definition, it follows that a stable extension is conflict-free. This is because otherwise it would hold that $\mathcal{A}rgs \cap \mathcal{A}rgs^+ \neq \emptyset$ and therefore $\mathcal{A}rgs^+ \neq Ar \backslash \mathcal{A}rgs$. Furthermore, a stable extension is also an admissible set, as it defeats all its defeaters. Finally, a stable extension is also a *maximal* admissible set, as any strict superset of a stable extension is not conflict-free. Therefore, a stable extension is also a preferred extension.

In terms of complete labellings, stable extensions correspond with labellings without `undec`. As a stable extension defeats every argument not in it, only two lables apply: `in` (for the stable extension itself) and `out` (for all the rest).

In figure 1 only one stable extension exists: $\{A, C\}$. In figure 3 two stable extensions exist: $\{A\}$ and $\{B\}$. In figure 2 there exist no stable extensions at all.

Exercise 8. *Give the stable extensions (if any exist) of:*

(a) *figure 4*

(b) *figure 5*

(c) *figure 6*

(d) *figure 7*

3.5 Semi-Stable Semantics

The last admissibility based semantics to be discussed is that of semi-stable semantics. Where stable semantics requires that `undec` is empty, semi-stable semantics merely requires that `undec` is *minimal*.

Definition 12. *Let* (Ar, def) *be an argumentation framework and* $\mathcal{A}rgs \subseteq Ar$*.* $\mathcal{A}rgs$ *is said to be a* semi-stable extension *iff* $\mathcal{A}rgs$ *is a complete extension of which* $\mathcal{A}rgs \cup \mathcal{A}rgs^+$ *is maximal.*

When $\mathcal{A}rgs \cup \mathcal{A}rgs^+$ is maximized, it follows that the associated complete labelling must have `undec` minimized.

In figure 1 only one semi-stable extension exists: $\{A, C\}$. In figure 3 two semi-stable extensions exist: $\{A\}$ and $\{B\}$. In figure 2 only one semi-stable extension exists: $\emptyset$.

Exercise 9. *Give the semi-stable extensions of:*

(a) *figure 4*

(b) *figure 5*

(c) *figure 6*

(d) *figure 7*

Every stable extension is a semi-stable extension. Also, if there exists at least one stable extension then the semi-stable extensions are the same as the stable extensions. Finally, every semi-stable extension is also a preferred extension, although the converse is generally not the case.

The existence of semi-stable extensions can only be assured for argumentation frameworks where the set of arguments is finite. That is, when the set of arguments in the argumentation framework is infinite, it might be that there exists no semi-stable extension. See [3] for an example of this.

3.6 Semantics Compared

There exists a partial ordering between the various admissibility-based semantics. Every stable extension is a semi-stable extension, every semi-stable extension is a preferred extension, every preferred extension is a complete extension, and every grounded extension is a complete extension. This is depicted in Figure 9.

In essence, a complete labelling can be seen as a subjective but reasonable point of view that an agent can take with respect to which arguments are `in`, `out` or `undec`. Each such position is internally coherent in the sense that, if questioned, the agent can use its own position to defend itself. It is possible for the position to be disagreed with, but at least one cannot point out an internal inconsistency. The set of all complete labellings thus stands for all possible and reasonable positions an agent can take.

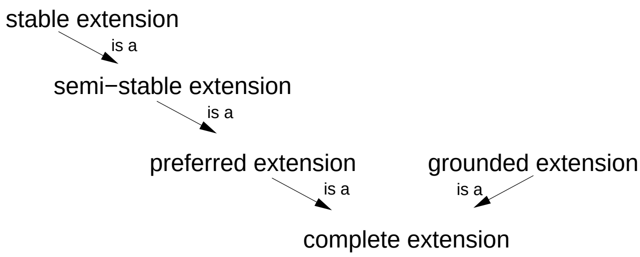

Figure 9: An overview of the different semantics.

Semantics	Description Complete Labellings	Extension Based Description
complete	all labellings	conflict-free fixpoint of F
grounded	labellings with minimal `in` labellings with minimal `out` labellings with maximal `undec`	minimal fixpoint of F minimal complete extension
preferred	labellings with maximal `in` labellings with maximal `out`	maximal admissible set maximal complete extension
semi-stable	labellings with minimal `undec`	admissible set with max. $\mathcal{A}rgs \cup \mathcal{A}rgs^+$ complete ext. with max $\mathcal{A}rgs \cup \mathcal{A}rgs^+$
stable	labellings with empty `undec`	$\mathcal{A}rgs$ defeating exactly $Ar \backslash \mathcal{A}rgs$ conflict-free $\mathcal{A}rgs$ defeating $Ar \backslash \mathcal{A}rgs$ admissible set $\mathcal{A}rgs$ defeating $Ar \backslash \mathcal{A}rgs$ complete ext. $\mathcal{A}rgs$ defeating $Ar \backslash \mathcal{A}rgs$ preferred ext. $\mathcal{A}rgs$ defeating $Ar \backslash \mathcal{A}rgs$ semi-stable ext. $\mathcal{A}rgs$ defeating $Ar \backslash \mathcal{A}rgs$

Table 1: An overview of admissibility based semantics

4 Proof Procedures

If one accepts the view that labellings correspond to the reasonable positions one can take in the presence of an argumentation framework, and one is interested whether a particular argument (say A) can be accepted, then two questions become relevant:

1. Is there *at least one* reasonable position where A is accepted? That is, is there at least one labelling $\mathcal{L}ab$ with $\mathcal{L}ab(A) = \texttt{in}$?
2. Is A accepted in *every* reasonable position? That is, does it hold that for every labelling $\mathcal{L}ab$, $\mathcal{L}ab(A) = \texttt{in}$?

The first question refers to the issue of credulous acceptance; the second question refers to the issue of sceptical acceptance.

4.1 Credulous Acceptance

When it comes to the question whether an argument A is labelled `in` in at least one complete labelling, a naive approach would be to generate all possible complete labellings and examine whether A is labelled `in` in at least one of them. For convenience, one could restrict oneself to complete labellings in which `in` is maximal (the preferred labellings), and apply an algorithm such as the one described in [8].

In this section, we will, however, treat a more sophisticated way of determining whether A is labelled `in` in every complete labelling, a way that requires us to generate not even one single complete labelling. To do so, we now introduce the notion of a *partial labelling.*

Definition 13. *A* partial labelling *is a partial function* $\mathcal{L}ab : Ar \longrightarrow \{\mathtt{in}, \mathtt{out}\}$ *such that:*

- *if* $\mathcal{L}ab(A) = \mathtt{in}$ *then for each defeater* B *of* A *it holds that* $\mathcal{L}ab(B) = \mathtt{out}$*, and*
- *if* $\mathcal{L}ab(A) = \mathtt{out}$ *then there exists a defeater* B *of* A *such that* $\mathcal{L}ab(B) = \mathtt{in}$*.*

A partial labelling differs in three respects from a complete labelling. First of all, not every argument needs to have a label, since $\mathcal{L}ab$ is a partial function. Secondly, a partial labelling does not contain `undec`. Thirdly, the conditions "iff" in Definition 2 are changed to the much weaker "if" in Definition 13.

The interesting property of partial labellings is that they correspond to admissible sets.

Theorem 1. *Let* (Ar, def) *be an argumentation framework.*

1. *If* $\mathcal{L}ab$ *is a partial labelling, then* $\mathtt{in}(\mathcal{L}ab)$ *is an admissible set.*
2. *If* $\mathcal{A}rgs$ *is an admisssible set, then there exists a partial labelling with* $\mathtt{in}(\mathcal{L}ab) = \mathcal{A}rgs$*.*

Proof.

1. Let $\mathcal{L}ab$ be a partial labelling. In order for $\mathtt{in}(\mathcal{L}ab)$ to be an admisssible set, two conditions need to hold:
   (a) $\mathtt{in}(\mathcal{L}ab)$ is conflict-free. Suppose this is not the case. Then there exist $A, B \in \mathtt{in}(\mathcal{L}ab)$ such that B defeats A. However, the fact that B defeats A means, by Definition 13, that $\mathcal{L}ab(B) = \mathtt{out}$. Contradiction.
   (b) Every argument B defeating some $A \in \mathtt{in}(\mathcal{L}ab)$ is defeated by an argument $C \in \mathtt{in}(\mathcal{L}ab)$. Let B be an arbitrary argument defeating some $A \in \mathtt{in}(\mathcal{L}ab)$. Then, according to Definition 13, it holds that $\mathcal{L}ab(B) = \mathtt{out}$. It then follows from Definition 13 that B is defeated by an argument C with $\mathcal{L}ab(C) = \mathtt{in}$. That is, $C \in \mathtt{in}(\mathcal{L}ab)$.
2. Let $\mathcal{A}rgs$ be an admissible set. Now consider the partial function $\mathcal{L}ab = \{(A, \mathtt{in}) \mid A \in \mathcal{A}rgs\} \cup \{(A, \mathtt{out}) \mid A \in \mathcal{A}rgs^+\}$. As $\mathcal{A}rgs$ is conflict-free it holds that $\mathcal{A}rgs \cap \mathcal{A}rgs^+ = \emptyset$, thus $\mathcal{L}ab$ is well-defined (it is not possible for an argument to be labelled both `in` and `out`). It holds that $\mathcal{L}ab$ is a partial labelling. This is because $\mathcal{L}ab$ satisfies the following:

(a) If $\mathcal{L}ab(A) = \texttt{in}$ then for each defeater B of A it holds that $\mathcal{L}ab(B) = \texttt{out}$.
Let $A \in Ar$ such that $\mathcal{L}ab(A) = \texttt{in}$. Let B be an arbitrary argument that defeats A. From the fact that $\mathcal{A}rgs$ is an admissible set, it follows that $\mathcal{A}rgs$ defeats B. That is, $B \in \mathcal{A}rgs^+$. This means that $\mathcal{L}ab(B) = \texttt{out}$.

(b) If $\mathcal{L}ab(A) = \texttt{out}$ then there exists a defeater B of A such that $\mathcal{L}ab(B) = \texttt{in}$.
Let $A \in Ar$ such that $\mathcal{L}ab(A) = \texttt{out}$. Then $A \in \mathcal{A}rgs^+$. This means that there exists a $B \in \mathcal{A}rgs$ such that B defeats A. As $B \in \mathcal{A}rgs$, it follows that $\mathcal{L}ab(B) = \texttt{in}$.

□

The interesting thing about a partial labelling is that it can always be extended to a complete labelling.

Theorem 2. *Let $\mathcal{L}ab$ be a partial labelling of argumentation framework (Ar, def). There exists a complete labelling $\mathcal{L}ab'$ such that $\mathcal{L}ab \subseteq \mathcal{L}ab'$.*

Proof. We first define two new functions: *extendout* and *extendin* such that:
$extendout(\mathcal{L}ab) = \mathcal{L}ab \cup \{(A, \texttt{out}) \mid (B, \texttt{in}) \in \mathcal{L}ab$ and B defeats $A\}$
$extendin(\mathcal{L}ab) = \mathcal{L}ab \cup \{(A, \texttt{in}) \mid$ for every B that defeats A: $(B, \texttt{out}) \in \mathcal{L}ab\}$
Let $\mathcal{L}ab''$ be the smallest superset of $\mathcal{L}ab$ that is closed under *extendout* and *extendin*. As an aside, $\mathcal{L}ab''$ could for a finite argumentation framework be computed by iteratively applying *extendout* and *extendin* on $\mathcal{L}ab$ until the result does not change anymore. As *extendout* and *extendin* are closed under partial labellings, $\mathcal{L}ab''$ is again a partial labelling. Let $\mathcal{L}ab' = \mathcal{L}ab'' \cup \{(A, \texttt{undec}) \mid A \in Ar$ and $(A, \texttt{in}) \notin \mathcal{L}ab''$ and $(A, \texttt{out}) \notin \mathcal{L}ab''\}$. It holds that $\mathcal{L}ab'$ is a complete labelling. For this, we have to prove that:

1. $\mathcal{L}ab(A) = \texttt{in} \Leftrightarrow$ for each defeater B of A it holds that $\mathcal{L}ab(B) = \texttt{out}$.
   The $\Rightarrow$ part follows directly from the definition of a partial labelling. The $\Leftarrow$ part follows from the fact that $\mathcal{L}ab'$ is closed under *extendin*.
2. $\mathcal{L}ab(A) = \texttt{out}$ iff A has a defeater B such that $\mathcal{L}ab(B) = \texttt{in}$.
   The $\Rightarrow$ part follows directly from the definition of a partial labelling. The $\Leftarrow$ part follows from the fact that $\mathcal{L}ab'$ is closed under *extendout*.

□

Theorem 3. *Let $\mathcal{L}ab'$ be a complete labelling of (Ar, def). There exists a partial labelling $\mathcal{L}ab$ such that $\mathcal{L}ab \subseteq \mathcal{L}ab'$.*

Proof. Partial labelling $\mathcal{L}ab$ can be constructed based on $\mathcal{L}ab'$ basically by omitting the `undec` labelled part of $\mathcal{L}ab'$. □

From Theorem 2 and 3 it follows that an argument A is labelled `in` in at least one complete labelling iff A is labelled `in` in at least one partial labelling. An interesting question, therefore, is how to determine whether an argument is labelled `in` in a partial labelling. A possible way of doing so is by means of formalized discussion, between a proponent (P) and an opponent (O) of argument A. Consider the following example:

P: I have a partial labelling where A is `in`.
O: Then, in your labelling A's defeater B must be `out`. Why?
P: B is `out` because B's defeater C is `in`.
O: Then, in your labelling C's defeater D must be `out`. Why?
⋮

In general, the debate whether an argument is `in` in a (partial/complete) labelling can be described as follows [22]:

- Proponent (P) and opponent (O) take turns; P begins.
- Each move of O is a defeater of some (not necessarily the directly preceeding) argument of P.
- Each responding move of P is a defeater of the directly preceeding argument of O.
- O is not allowed to repeat its own moves, but may repeat the proponent's moves.
- P is allowed to repeat it's own moves, but may not repeat the opponent's moves.

This can be formalized as follows.

Definition 14. *A* discussion under preferred semantics *(*p-discussion*) is a list* $[A_1, A_2, \ldots, A_n]$ *— where A_i is called a* proponent-move *iff i is odd, and an* opponent-move *iff i is even — such that:*

1. *for every proponent-move A_i ($i \geq 3$) there exists an opponent-move A_j, with $j = i - 1$, such that A_i defeats A_j;*
2. *for every opponent-move A_i ($i \geq 2$) there exists a proponent-move A_j, with $j < i$, such that A_i defeats A_j; and*
3. *there exist no two opponent-moves A_i and A_j such that $A_i = A_j$ and $i \neq j$.*
4. *there exists no proponent-move A_i and opponent-move A_j with $i > j$ and $A_i = A_j$.*

A p-discussion $[A_1, A_2, \ldots, A_n]$ is finished *iff (1) A_n is an opponent move that is equivalent to an opponent move A_i with $i < n$, or (2) there exists no A_{n+1} such that $[A_1, A_2, \ldots, A_n, A_{n+1}]$ is a discussion. A finished p-discussion is won by the proponent iff the last move is a proponent-move; it is won by the opponent iff the last move is an opponent-move.*

Theorem 4. *Let (Ar, def) be an argumentation framework and $A \in Ar$. A is labelled* `in` *in at least one partial labelling iff there exists at least one p-discussion that is won by the proponent.*

Proof.
"$\Longrightarrow$": Let $\mathcal{L}ab$ be a partial labelling with $\mathcal{L}ab(A) =$ `in`. Let $[A_1, A_2, \ldots, A_n]$ (with $A_1 = A$ and $n \geq 1$) be a discussion where the last move is a proponent-move and every proponent-move is labelled `in` and every opponent-move is labelled `out` by $\mathcal{L}ab$; such a discussion always exists since a trivial one is $[A]$. Then for every opponent-move A_{n+1} there exists a proponent-move A_{n+2} such that $[A_1, A_2, \ldots, A_n, A_{n+1}, A_{n+2}]$ is a

discussion in which every opponent-move is labelled `out` and every proponent-move is labelled `in` by $\mathcal{L}ab$. This is because A_{n+1} is labelled `out` (follows from the definition of a partial labelling and the fact that all proponent-moves in $[A_1, A_2, \ldots, A_n]$ are labelled `in`) and there exists an argument A_{n+2} that is labelled `in` and defeats A_{n+1} (also follows from the definition of a partial labelling, and the fact that A_{n+1} is labelled `out`). From the thus derived observation that every discussion where the last move is a proponent-move can be extended to a discussion where the last move is again a proponent-move (under the condition that the discussion can be extended at all), and the fact that it is not possible to infinitly extend a discussion (due to the finiteness of the argumentation framework, and the fact that the opponent is not allowed to repeat its moves), it follows that the proponent has a winning strategy.
"$\Longleftarrow$": Suppose the proponent has a winning strategy for A. Then there exists a finished discussion won by the proponent. Now define the function $\mathcal{L}ab$ such that $\mathcal{L}ab(B) = \texttt{in}$ for every proponent-argument B in this discussion, and $\mathcal{L}ab(C) = \texttt{out}$ for every opponent-argument C in this discussion. It holds that:

- each argument labelled `out` has a defeater that is labelled `in` (this follows from the fact that the last move is a proponent-move).
- for each argument labelled `in` it holds that all its defeaters are labelled `out` (this follows from the fact that the discussion is finished with a proponent-move, which means that all possible counterarguments against the proponent-moves have already been given, and they are all labelled `out`).

Thus, it follows that $\mathcal{L}ab$ is a partial labelling. $\square$

4.2 Sceptical Acceptance

The next thing to be studied is how to determine which arguments are labelled `in` (or `out`) in every complete labelling. Recall that the grounded extension is equal to the intersection of all complete extensions. As complete labellings and complete extensions are essentially equivalent, determining whether an argument is labelled `in` in every complete labelling can be done by examining whether the argument is labelled `in` in the grounded labelling.

In most cases, the grounded labelling can be calculated in a relatively straightforward way. The basic idea can be illustrated using the argumentation framework of Figure 6 (page 5). Here, argument B has no defeaters, so it must be labelled `in` in every complete labelling. This then means that argument C must be labelled `out` in every complete labelling, causing D to be labelled `in` in every complete labelling. The idea is thus to start with the arguments that have no defeaters and label them `in`, then to examine which arguments as a result of that must be labelled `out`, then to examine which arguments as a result of that must be labelled `in`, etc...

Formally, the procedure can be described as follows. Let *extendin* be a function such that for a partial labelling $\mathcal{L}ab$ it holds that $extendin(\mathcal{L}ab) = \mathcal{L}ab \cup \{(A, \texttt{in}) \mid$ for every B that defeats A it holds that $\mathcal{L}ab(B) = \texttt{out}\}$. Let *extendout* be a function such that for a partial labelling $\mathcal{L}ab$ it holds that $extendout(\mathcal{L}ab) = \mathcal{L}ab \cup \{(A, \texttt{out}) \mid$ there is a defeater B of A such that $\mathcal{L}ab(B) = \texttt{in}\}$. Let *extendinout* be the function $extendout \circ extendin$. It can be verified that if $\mathcal{L}ab$ is a partial labelling, then

$extendin(\mathcal{L}ab)$, $extendout(\mathcal{L}ab)$ and $extendinout(\mathcal{L}ab)$ are also partial labellings. Notice that *extendinout* is similar to the function F of Definition 6. The main difference is that *extendinout* is defined for partial labellings whereas F is defined for (admissible) sets.

For a finite argumentation framework the grounded labelling can be determined by successively applying *extendinout*, starting from the empty set, until the result yields nothing new anymore. That is, one can apply the following algorithm.

```
L :=∅
REPEAT
 L_old := L;
 L :=extendin(L);
 L :=extendout(L);
UNTIL(L = L_old)
```

Let $\mathcal{L}ab'$ be $\mathcal{L}ab$ in which all unlabelled arguments are labelled `undec`. That is, $\mathcal{L}ab' = \mathcal{L}ab \cup \{(A, \texttt{undec}) \mid (A, \texttt{in}) \notin \mathcal{L}ab \text{ and } (A, \texttt{out}) \notin \mathcal{L}ab\}$. It now holds that $\mathcal{L}ab'$ is the grounded labelling.

In the case of an infinite argumentation framework, the situation is slightly more complex. It holds that if an argument is `in` (`out`) in $\cup_{i=o}^{\infty} extendinout^i(\emptyset)$ then the argument is `in` (`out`) in the grounded labelling. The converse, however, does not need to be the case. A counterexample is the argumentation framework of Figure 10 (taken from [9]) where each A_i defeats A_{i+1}, and each A_i with even i defeats B. Here, the grounded labelling labels B `in` (as well as every A_i with odd i). Nevertheless, B is not labelled `in` in $\cup_{i=o}^{\infty} extendinout^i(\emptyset)$ since this would require that every A_i with ever i is labelled `out`, which is impossible to achieve in a finite number of steps.

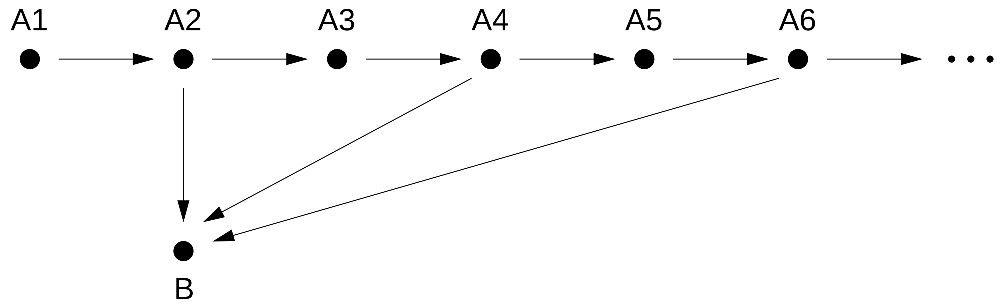

Figure 10: An argumentation framework that is not finitary.

Nevertheless, in [9] it is proved that in any finitary argumentation framework (that is, an argumentation framework where each argument has at most a finite number of defeaters) the grounded extension is equal to $\cup_{i=o}^{\infty} F^i(\emptyset)$. From this it follows that in any finitary argumentation framework, the grounded labelling is equal to $\cup_{i=o}^{\infty} extendinout^i(\emptyset)$.

The disadvantage of the above procedure is that it requires computing the entire grounded extension, even if one is only interested in whether a single argument (A)

is in it. In many cases, there exists a faster and more efficient way of determining such. The idea is to apply a dialectical argument game, much like the p-discussions that were discussed in the previous section.

The discussion whether an argument is labelled `in` in every complete labelling (a discussion under grounded semantics, or simply g-discussion) can be described as follows [19, 5, 18]:

- Proponent (P) and opponent (O) take turns; the proponent begins.
- Each move of O is a defeater of the directly preceeding argument of P.
- Each move of P is a defeater of the directly preceeding argument of O.
- P is not allowed to repeat any of its earlier moves.
- O is allowed to repeat its earlier moves.

Definition 15. *A* discussion under grounded semantics *(g-discussion) is a list* $[A_1, A_2, \ldots, A_n]$ *— where* A_i *is called a* proponent-move *iff* i *is odd, and an* opponent-move *iff* i *is even — such that:*

1. *every proponent-move* A_i $(i \geq 3)$ *defeats the directly preceeding opponent-move* A_{i-1}*;*
2. *every opponent-move* A_i $(i \geq 2)$ *defeats the directly preceeding proponent-move* A_{i-1}*; and*
3. *there exist no two proponent-moves* A_i *and* A_j *such that* $A_i = A_j$ *and* $i \neq j$*.*

A g-discussion $[A_1, A_2, \ldots, A_n]$ *is* finished *iff there exists no* A_{n+1} *such that* $[A_1, A_2, \ldots, A_n, A_{n+1}]$ *is a g-discussion. A finished discussion is won by the proponent iff the last move is a proponent-move; it is won by the opponent iff the last move is an opponent-move.*

To determine whether an argument is labelled `in` in the grounded labelling, it is not enough that there exists at least one g-discussion that is won by the proponent. What is needed is the existence of a *winning strategy.* The idea of a winning strategy is to provide a roadmap for which moves the proponent should play, taking into account every possible counter move of the opponent.

Definition 16. *Let* A *be an argument. A* winning strategy *for* A *is a tree of which the nodes are associated with arguments, the root associated with* A*, such that:*

1. *each path from the root to a leaf corresponds to a discussion won by the proponent,*
2. *each opponent-move on such a path has exactly one child, and*
3. *each proponent-move (say* B*) on such a path has a child (*C*) for every* C *such that* $[A, \ldots, B, C]$ *is a discussion.*

Theorem 5. *Let* (Ar, def) *be a finitery argumentation framework and* $A \in Ar$*.* A *is labelled* `in` *in every complete labelling iff the proponent has a winning strategy for* A *in the discussion using g-discussions.*

Proof.
"$\Longrightarrow$": Suppose A is labelled `in` in every complete labelling. Then it is also labelled `in` in the complete labelling that corresponds with the grounded extension. This means there exists some minimal i such that $extendinout^i(\emptyset)$ labels A `in`. We now prove that A has a winning strategy. This is done by induction on i.

basis Let $i = 1$. In this case, A does not have any counterarguments, which means $[A]$ is a finished discussion that is won by the proponent, which trivially implies that there exists a winning strategy for A.

step Suppose that all arguments labelled `in` in $extendinout^i(\emptyset)$ have a winning strategy. We now prove that also all arguments labelled `in` in $extendinout^{i+1}(\emptyset)$ have a winning strategy. Let A be an arbitrary argument that is labelled `in` by $extendinout^{i+1}(\emptyset)$ and let $\{B_1, B_2, \ldots, B_m\}$ be the set of defeaters of A. From the definition of *extendinout* it follows that every B_j $(1 \leq j \leq m)$ is labeled `out` in $extendinout^i(\emptyset)$. This means that for every B_j there exists some C_j that is labelled `in` in $extendinout^i(\emptyset)$ such that C_j defeats B_j. According to the induction hypothesis, there exists a winning strategy for each C_j. Therefore, there also exists a winning strategy for A. Notice that in this winning strategy the proponent indeed does not repeat any of its moves (this is because A is not labelled `in` by $extendinout^i(\emptyset)$).

"$\Longleftarrow$": Let $A \in Ar$ be an argument for which the proponent has a winning strategy. We now prove that A is labeled `in` in the grounded labeling (and therefore also in every complete labelling). This is done by induction on the depth of the winning strategy. As every discussion in the winning strategy is won by the proponent, the depth of each winning strategy is always an odd number.

basis Let $i = 1$. In this case the winning strategy consists of a single node A. This means there are no counterarguments against A. It then follows that A is labelled `in` in $extendinout(\emptyset)$ and is therefore labelled `in` in the grounded labelling.

step Suppose that for each winning strategy with a depth less or equal to i it holds that its root is labelled `in` in the grounded labelling. We now prove that each winning strategy with a depth of $i + 2$ also has its root labelled `in` in the grounded labelling. Let WS be a winning strategy, with root A, of depth less or equal to $i + 2$. Now consider all subtrees starting at a distance of 2 from the root. These are again winning strategies, of depth less or equal to i. The induction hypothesis states that the roots of these trees are labelled `in` in the grounded labelling. This means that there exists a j such that $extendinout^j(\emptyset)$ labels the roots of all these winning strategies `in`. From this it follows that that A is labelled `in` in $extendinout^{j+1}(\emptyset)$ and is therefore labelled `in` in the grounded labelling.

$\square$

5 Other Semantics

The different semantics that have been treated until now (complete, grounded, preferred, stable and semi-stable) are those that can easily be descibed in terms of complete labellings. Except for semi-stable semantics, they have all been treated in the landmark paper of Dung [9].

One interesting additional semantics, however, comes under the name of *ideal sematics* [1, 11, 10]. An ideal extension can be described as the largest admissible set (w.r.t set-inclusion) that is a subset of every preferred extension. For example, in Figure 11, the ideal extension is empty. This is because there exists two preferred extensions ($\{A, D\}$ and $\{B, D\}$) whose intersection is $\{D\}$. The largest admissible subset of $\{D\}$ is $\emptyset$. In Figure 12, the ideal extension is $\{B\}$. This is because there exists just one preferred extension ($\{B\}$) which then automatically becomes the ideal extension. It can be shown that there always exists exactly one ideal extension. Furthermore, it can also be shown that the ideal extension is also a complete extension, and is therefore a superset of (possibly equal to) the grounded extension. The value of ideal semantics is that it is a unique-extension semantics that less sceptical than grounded.

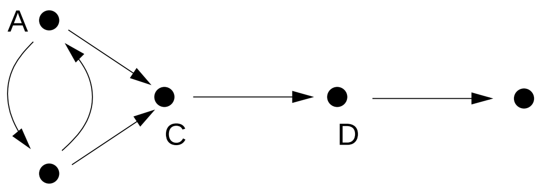

Figure 11: Here, the ideal extension is $\emptyset$.

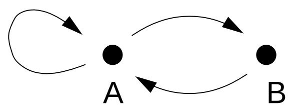

Figure 12: Here, the ideal extension is $\{B\}$.

An important design consideration when defining an argumentation semantics is whether each resulting extension is guaranteed to be an admissible set. For the semantics that have been treated until now, this is indeed the case. Every complete, grounded, preferred, stable, semi-stable or ideal extension is itself an admissible set. There also exist semantics, however, where this is not the case.

An example of a non-admissibility based semantics is the work of Jakobovits and Vermeir [14]. In their formalism, argument C in Figure 13 would be justified. In the CF2 semantics defined by Baroni et al. [2] it is on the other hand argument B that becomes justified.

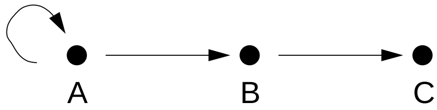

Figure 13: Argument B justified in [2]; Argument C justified in [14].

In this document, we will not treat the non-admissibility based semantics in any detail. One of the reasons for this is that by dropping the requirement of admissibility one introduces a wide range of problems and difficulties, both from technical and philosophical perspective. The technical difficulties have to do with what happens if one uses arguments that are constructed from a knowledge base, and fall outside the scope of the current document. The philosophical difficulties have to do with the nature of a justified argument, and are discussed below.

In essence, one can distinguish two kinds of justifications of a statement: model-theoretic and dialectical. A statement is justified ("true") in the model-theoretic sense if it follows from every possible model. A statement is justified in the dialectical sense if it can be defended in a rational discussion. Although most logic based research nowadays is based on the model-theoretic approach, one could in many cases equally well apply a dialectical approach to semantics. For clasical logic, such an approach has been worked out by Lorenzen and Lorenz [16], who define a rational discusion for determining the validity of a classical first order formula.

For defeasible reasoning and argumentation, the model-theoretical approach has received some following, but has not become dominant, perhaps because the kind of preferred model semantics that defeasible reasoning seems to require is not always easy and straightforward to deal with. A more natural approach seems to be to choose the dialectical way of justifying arguments, an approach that can be traced back to classical antiquity, in particular to the discourses of Socrates.

Let us consider the example of the 3-cycle of Figure 2. Here, the non-admissibility based CF2-semantics [2] would yield three extensions: $\{A\}$, $\{B\}$ and $\{C\}$. One can easily imagine a Socratic-style discussion in which the tenability of, for instance, A is questioned.

Proponent: "I hold that A"
Opponent: "Then you have to reject A's defeater C. Based on what grounds?
Proponent: "I reject C because I hold that B"
Opponent: "But then you would have to explicitly reject B's defeater A, don't you?"
Proponent: [understands that he has caught himself in] "Euhhh..."

It should be observed that the above discussion is essentially an informal p-discusion. Thus, one can see that essentially every argument that is not part of an admisible set cannot be defended in a reasonable debate, as any such attempt leads to self-refutation on the side of the proponent. It appears that that which cannot be defended should not be called justified, nor be part of any reasonable position (like an extension of arguments). This, as well as reasons related to quality postulates [4], made us decide not to treat these “advanced” forms of semantics and instead restrict ourselves to the traditional semantics where each extension is also an admisible set.

References

[1] José Júlio Alferes, Phan Minh Dung, and Luís Pereira. Scenario semantics of extended logic programs. In A. Nerode and L. Pereira, editors, *Proc. 2nd International Workshop on Logic Programming and Non-monotonic Reasoning*, pages 334–348. MIT Press, 1993.

[2] P. Baroni, M. Giacomin, and G. Guida. Scc-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168(1-2):165–210, 2005.

[3] Martin Caminada. Semi-stable semantics for formal argumentation. paper available on request (martin.caminada@uni.lu).

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

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

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

[7] M.W.A. Caminada. Semi-stable semantics. In P.E. Dunne and TJ.M. Bench-Capon, editors, *Computational Models of Argument; Proceedings of COMMA 2006*, pages 121–130. IOS Press, 2006.

[8] S. Doutre and J. Mengin. An algorithm that computes the preferred extensions of argumentation frameworks. In *ECAI'2000, Third International Workshop on Computational Dialectics (CD'2000)*, pages 55–62, August 2000.

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

[10] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. Technical report, Imperial College, 2006.

[11] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. A dialectical procedure for sceptical, assumption-based argumentation. In P.E. Dunne and TJ.M. Bench-Capon, editors, *Computational Models of Argument; Proceedings of COMMA 2006*, pages 145–156. IOS, 2006.

[12] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of the 5th International Conference/Symposium on Logic Programming*, pages 1070–1080. MIT Press, 1988.

[13] J. Horty. Argument construction and reinstatement in logics for defeasible reasoning. *Artificial Intelligence and Law*, 9:1–28, 2001.

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

[15] K. Konolige. Defeasible argumentation in reasoning about events. *Methodologies for Intelligent Systems*, 3:380–390, 1988.

[16] P. Lorenzen and K. Lorenz. Dialogische logik. *Wissenschaftliche Buchgesellschaft, Darmstadt*, 1978.

[17] J. L. Pollock. How to reason defeasibly. *Artificial Intelligence*, 57:1–42, 1992.

[18] H. Prakken. Commonsense reasoning. Technical report, Institute of Information and Computing Sciences, Utrecht University, 2004. Course material.

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

[20] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[21] Allen van Gelder, Kenneth A. Ross, and John S. Schlipf. The well-founded semantics for general logic programs. *J. ACM*, 38(3):620–650, 1991.

[22] G. A. W. Vreeswijk and H. Prakken. 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, pages 239–253, Berlin, 2000. Springer Verlag.