

RESEARCH ARTICLE

A Labelling Approach for Ideal and Stage Semantics

Martin Caminada

University of Luxembourg

(Received 00 Month 200x; final version received 00 Month 200x)

In this document, we describe the concepts of ideal semantics and stage semantics for abstract argumentation in terms of argument labellings. The difference between the traditional extensions approach and the labelling approach is that where the former only identifies the sets of accepted arguments, the latter also identifies the rejected arguments as well as the arguments that are neither accepted nor rejected. So far, the labellings approach has been successfully applied to complete, grounded, preferred, stable and semi-stable semantics, as well as to the concept of admissibility. In the current paper, we continue this line of research by showing that also ideal semantics and stage semantics can be described in terms of argument labellings.

Keywords: formal models of argumentation; argumentation frameworks; argumentation semantics

1. Introduction

Formal argumentation has become a popular approach for purposes varying from nonmonotonic reasoning (3, 9), multi-agent communication (1) and reasoning in the semantic web (20). Although some research on formal argumentation can be traced back to the early 1990s (like for instance the work of Vreeswijk (24) and of Simari and Loui (21)) the topic really started to take off with Dung's theory of abstract argumentation (13). Here, arguments are seen as abstract entities (although they can be instantiated using approaches like (9) and (18)) among which an attack relationship is defined. The thus formed *argumentation framework* can be represented as a directed graph in which the arguments serve as nodes and the attack relation as the arrows.

Given such a graph, an interesting question is which sets of nodes can reasonably be accepted. Several criteria of acceptance have been stated, including Dung's original approaches of grounded, complete, preferred and stable semantics (13), as well as other approaches like semi-stable semantics¹ (6, 22) and ideal semantics (14).

The concepts of admissibility, as well as that of complete, grounded, preferred, stable or semi-stable semantics were originally stated in terms of sets of arguments. It is equally well possible, however, to express these concepts using argument labellings. This approach was pioneered by Pollock (17) and has subsequently been applied by Verheij (22), Jakobovits and Vermeir (15), Caminada (4, 7, 10) and Vreeswijk (25). In this paper we follow the approach of (4, 7, 10) where each argument is assigned a label, which can either be in, out or undec. The label in indicates that the argument is explicitly accepted, the label out indicates that the argument is explicitly rejected, and the label undec indicates that the status of the argument

¹Semi-stable semantics was described in terms of *admissible stage extensions* in (22).

is undecided, meaning that one abstains from an explicit judgment whether the argument is in or out.

An advantage of the labellings approach is that it becomes possible to describe complete, preferred, grounded, stable and semi-stable semantics without having to refer to technical concepts like admissibility or fixpoints of the characteristic function F , as is explained in (4, 10). Moreover, the labelling approach has also been used for defining argumentation based algorithms, for instance for algorithms that, given an argumentation framework, compute its stable, semi-stable or preferred labellings (7, 16) (of which the sets of in-labelled arguments correspond to the stable, semi-stable and grounded extensions, respectively). Another recent application of argument labellings is in the field of judgement aggregation. In essence, a (complete) labelling can be seen as a reasonable position an agent can take in the presence of the conflicting information expressed in the argumentation framework. The question then becomes how several of such positions can be aggregated to a collective opinion. This has been the topic of work of Caminada and Pigozzi (11), as well as of Rahwan and Tohmé (19).

In previous work (4, 5, 10) labellings have been specified for complete, stable, semi-stable, preferred and grounded semantics. The key concept is that of a complete labelling, which is shown to implement complete semantics (4, 10). The other semantics can then be described in terms of complete labellings in which the occurrence of a particular type of label is maximized or minimized. A preferred labelling, for instance, can be described as a complete labelling in which the set of in-labelled arguments is maximal (w.r.t. set-inclusion). Similarly, the grounded labelling can be described as a complete labelling in which the set of in-labelled arguments is minimal, and a semi-stable labelling as a complete labelling in which the set of undec-labelled arguments is minimal. In the current paper, we apply the labellings approach to two additional semantics: ideal (14) and stage (22). Although these semantics cannot be described by simply maximizing or minimizing the occurrence of a particular type of label using complete labellings (as is the case for grounded, preferred and semi-stable semantics) we show that the labelling approach can nevertheless be meaningfully applied. We show that ideal semantics can be described in terms of judgement aggregation. In particular, we show that the ideal labelling is the most committed admissible (or complete) labelling that is compatible with each admissible, complete or preferred labelling. Where the key concept of ideal semantics is that of compatibility with other labellings (Section 3) the key concept of stage semantics is that of maximal consistency (Section 4). In essence, a stage labelling is a stable labelling of a maximal subgraph of the argumentation framework that has at least one stable labelling, augmented by labelling all arguments undec that are outside of the subgraph. However, before being able to explain ideal and stage labellings in more detail, we first provide some formal preliminaries in Section 2.

2. Formal Preliminaries

An *argumentation framework* (13) consists of a set of arguments and an attack relation on these arguments. In order to simplify the discussion, we consider only finite argumentation frameworks.

Definition 2.1: An *argumentation framework* is a pair (Ar, att) where Ar is a set of arguments and $att \subseteq Ar \times Ar$.

We say that (1) an argument A *attacks* an argument B iff $(A, B) \in att$, (2) an argument A attacks a set of arguments $Args$ iff A attacks an argument in $Args$,

(3) a set of arguments attacks an argument A iff an argument in $Args$ attacks A , and (4) a set of argument $Args_1$ attacks a set of arguments $Args_2$ iff an argument in $Args_1$ attacks an argument in $Args_2$.

The shorthand notation A^+ and A^- stands for, respectively, the set of arguments attacked by argument A and the set of arguments that attack argument A . Likewise, if $Args$ is a set of arguments, then we write $Args^+$ for the set of arguments that are attacked by at least one argument in $Args$, and $Args^-$ for the set of arguments that attack at least one argument in $Args$. In the definition below, $F(Args)$ stands for the set of arguments that are acceptable in the sense of (13).

Definition 2.2 defense / conflict-free Let (Ar, att) be an argumentation framework, $A \in Ar$ and $Args \subseteq Ar$.

We define A^+ as $\{B \mid A \text{ att } B\}$ and $Args^+$ as $\{B \mid A \text{ att } B \text{ for some } A \in Args\}$.

We define A^- as $\{B \mid B \text{ att } A\}$ and $Args^-$ as $\{B \mid B \text{ att } A \text{ for some } A \in Args\}$.

$Args$ is *conflict-free* iff $Args \cap Args^+ = \emptyset$.

$Args$ *defends* an argument A iff $A^- \subseteq Args^+$.

We define the function $F : 2^{Ar} \rightarrow 2^{Ar}$ as

$F(Args) = \{A \mid A \text{ is defended by } Args\}$.

In the definition below, notions of grounded, preferred and stable semantics are described in terms of complete semantics. These descriptions are not literally the same as the ones provided by Dung (13), but as was stated in (4), these are in fact equivalent to Dung's original versions of grounded, preferred and stable semantics.

Definition 2.3 acceptability semantics Let (Ar, att) be an argumentation framework and let $Args \subseteq Ar$ be a conflict-free set of arguments.

- $Args$ is *admissible* iff $Args \subseteq F(Args)$.
- $Args$ is a *complete* extension iff $Args = F(Args)$.
- $Args$ is a *grounded* extension iff $Args$ is the minimal (w.r.t. set-inclusion) complete extension.
- $Args$ is a *preferred* extension iff $Args$ is a maximal (w.r.t. set-inclusion) complete extension.
- $Args$ is a *stable* extension iff $Args$ is a complete extension that attacks every argument in $Ar \setminus Args$.
- $Args$ is a *semi-stable* extension iff $Args$ is a complete extension where $Args \cup Args^+$ is maximal (w.r.t. set-inclusion).

In (4, 10) it is described how the concepts of complete, grounded, preferred, stable and semi-stable semantics, as well as the concept of admissibility, can be characterised in terms of argument labellings.

Definition 2.4: Let (Ar, att) be an argumentation framework. A *labelling* is a total function $Lab : Ar \rightarrow \{\text{in}, \text{out}, \text{undec}\}$.

We write $\text{in}(Lab)$ for $\{A \mid Lab(A) = \text{in}\}$, $\text{out}(Lab)$ for $\{A \mid Lab(A) = \text{out}\}$ and $\text{undec}(Lab)$ for $\{A \mid Lab(A) = \text{undec}\}$. Sometimes, we write a labelling Lab as a triple $(Args_1, Args_2, Args_3)$ where $Args_1 = \text{in}(Lab)$, $Args_2 = \text{out}(Lab)$ and $Args_3 = \text{undec}(Lab)$.

We introduce two functions (Lab2Ext and Ext2Lab) that allows one to convert a labelling to an extension, and an extension to a labelling. When converting a labelling to an extension, one basically selects the set of in-labelled arguments. That is, $\text{Lab2Ext}(Lab) = \text{in}(Lab)$. When converting an extension to a labelling, one labels the arguments of the extension in, the arguments attacked by the extension out and the other arguments undec. That is, $\text{Ext2Lab}(Args) = (Args, Args^+, Ar \setminus (Args \cup Args^+))$, where Ar is the set of all arguments in the argumentation framework.

Notice that although Lab2Ext is defined for any labelling, Ext2Lab is defined only for sets of arguments that are conflict-free (otherwise the result would not be a correct labelling).

Definition 2.5: Let (Ar, att) be an argumentation framework. An *admissible labelling* is a labelling such that for every $A \in Ar$ it holds that:

- (1) if A is labelled in then all attackers of A are labelled out
- (2) if A is labelled out then A has an attacker that is labelled in

Admissible labellings correspond to admissible sets, although the relationship is not one-to-one.¹

Theorem 2.6: Let $AF = (Ar, att)$ be an argumentation framework. It holds that:

- (1) if $Args$ is an admissible set of AF then $\text{Ext2Lab}(Args)$ is an admissible labelling of AF , and
- (2) if Lab is an admissible labelling of AF then $\text{Lab2Ext}(Lab)$ is an admissible set of AF .

Definition 2.7: Let (Ar, att) be an argumentation framework. A *complete labelling* is a labelling such that for every $A \in Ar$ it holds that:

- (1) if A is labelled in then all attackers of A are labelled out
- (2) if A is labelled out then A has an attacker that is labelled in
- (3) if A is labelled undec then not all attackers of A are labelled out and A does not have an attacker that is labelled in

Notice that the first two conditions of a complete labelling are the same conditions as of an admissible labelling. Whereas an admissible labelling requires one to be able to explain why an argument is labelled in and why an argument is labelled out, a complete labelling also requires one to explain why an argument is labelled undec.

It is also possible to describe a complete labelling in a different way.

Proposition 2.8 (11): Let Lab be a labelling of an argumentation framework $AF = (Ar, att)$. Lab is a complete labelling iff for each argument $A \in Ar$ it holds that:

- (1) A is labelled in iff all its attackers are labelled out, and
- (2) A is labelled out iff it has at least one attacker that is labelled in.

The difference between Definition 2.7 and Proposition 2.8 is that the former describes a complete labelling using three if-statements, and the latter describes it using two iff-statements. Although the latter description does not explicitly mention the undec label, it follows that for an argument to be labelled undec, it cannot have all its attackers labelled out (otherwise it would have to be labelled in according to point 1 of Proposition 2.8). Similarly, it cannot have an attacker that is labelled in (otherwise it would have to be labelled out according to point 2 of Proposition 2.8). These two observations, when put together, are equivalent with point 3 of Definition 2.7. Hence, the description of complete labellings of Proposition 2.8 puts restrictions on the use of the undec label, even without explicitly referring to it.

¹For instance, in the argumentation framework of Figure 2, both $(\{A\}, \emptyset, \{B, C\})$ and $(\{A\}, \{B\}, \{C\})$ are admissible labellings that are associated with the same admissible set $\{A\}$.

Lemma 2.9 (11) *Let $\mathcal{L}ab_1$ and $\mathcal{L}ab_2$ be complete labellings of argumentation framework $AF = (Ar, att)$. It holds that:*

- (1) $\text{in}(\mathcal{L}ab_1) \subseteq \text{in}(\mathcal{L}ab_2)$ iff $\text{out}(\mathcal{L}ab_1) \subseteq \text{out}(\mathcal{L}ab_2)$,
- (2) $\text{in}(\mathcal{L}ab_1) \subsetneq \text{in}(\mathcal{L}ab_2)$ iff $\text{out}(\mathcal{L}ab_1) \subsetneq \text{out}(\mathcal{L}ab_2)$ and
- (3) $\text{in}(\mathcal{L}ab_1) = \text{in}(\mathcal{L}ab_2)$ iff $\text{out}(\mathcal{L}ab_1) = \text{out}(\mathcal{L}ab_2)$.

Complete labellings correspond to complete extensions, as is stated by the following theorem.

Theorem 2.10: *Let $AF = (Ar, att)$ be an argumentation framework. It holds that:*

- (1) *if $Args$ is a complete extension of AF then $\text{Ext2Lab}(Args)$ is a complete labelling of AF , and*
- (2) *if $\mathcal{L}ab$ is a complete labelling of AF then $\text{Lab2Ext}(\mathcal{L}ab)$ is a complete extension of AF .*

In (11) it is also proved that the relationship between complete labellings and complete extensions is one-to-one. That is, each complete labelling is associated to exactly one complete extension and vice versa. This property can be seen as a result of Lemma 2.9.

Using the concept of complete labellings, one can then also describe the concept of grounded, preferred, stable and semi-stable semantics in terms of argument labellings, as has been done in (11). A summary is provided in the following definition.

Definition 2.11: *Let $\mathcal{L}ab$ be a complete labelling of argumentation framework $AF = (Ar, att)$.*

- *$\mathcal{L}ab$ is a preferred labelling iff $\text{in}(\mathcal{L}ab)$ is maximal w.r.t. set-inclusion among all complete labellings.*
- *$\mathcal{L}ab$ is the grounded labelling iff $\text{in}(\mathcal{L}ab)$ is minimal w.r.t. set-inclusion among all complete labellings.*
- *$\mathcal{L}ab$ is a semi-stable labelling iff $\text{undec}(\mathcal{L}ab)$ is minimal w.r.t. set-inclusion among all complete labellings.*
- *$\mathcal{L}ab$ is a stable labelling iff $\text{undec}(\mathcal{L}ab) = \emptyset$.*

Preferred, grounded, semi-stable and stable labellings correspond to preferred, grounded, semi-stable and stable extensions, respectively, as is stated in the following theorem.

Theorem 2.12: *Let $AF = (Ar, att)$ be an argumentation framework. It holds that:*

- (1) *if $Args$ is a preferred extension of AF then $\text{Ext2Lab}(Args)$ is a preferred labelling of AF , and if $\mathcal{L}ab$ is a preferred labelling of AF then $\text{Lab2Ext}(\mathcal{L}ab)$ is a preferred extension of AF .*
- (2) *if $Args$ is the grounded extension of AF then $\text{Ext2Lab}(Args)$ is the grounded labelling of AF , and if $\mathcal{L}ab$ is the grounded labelling of AF then $\text{Lab2Ext}(\mathcal{L}ab)$ is the grounded extension of AF .*
- (3) *if $Args$ is a semi-stable extension of AF then $\text{Ext2Lab}(Args)$ is a semi-stable labelling of AF , and if $\mathcal{L}ab$ is a semi-stable labelling of AF then $\text{Lab2Ext}(\mathcal{L}ab)$ is a semi-stable extension of AF .*
- (4) *if $Args$ is a stable extension of AF then $\text{Ext2Lab}(Args)$ is a stable labelling of AF , and if $\mathcal{L}ab$ is a stable labelling of AF then $\text{Lab2Ext}(\mathcal{L}ab)$ is a stable extension of AF .*

Since for complete semantics, the relation between labellings and extensions is one-to-one, it follows that for preferred, grounded, stable and semi-stable semantics the relation between labellings and extensions is also one-to-one.

It turns out that there are different ways to describe the concepts of preferred, grounded, semi-stable and stable labellings. For instance, a preferred labelling can also be described as a complete labelling where the set of out-labelled arguments is maximal. Similarly, the grounded labelling can also be described as the complete labelling where the set of out-labelled arguments is minimal, or where the set of undec-labelled arguments is maximal.

restriction complete labellings	Dung-style semantics
no restrictions	complete semantics
empty undec	stable semantics
maximal in	preferred semantics
maximal out	preferred semantics
maximal undec	grounded semantics
minimal in	grounded semantics
minimal out	grounded semantics
minimal undec	semi-stable semantics

Table 1. Argument labellings and Dung-style semantics (10).

Table 1, taken from (4, 10), examines what happens when one focusses on complete labellings in which one particular label has been maximized or minimized. It turns out that in all but one case, one obtains a correspondence with one of the semantics described in (13). The only exception is when one minimizes undec, in which one obtains correspondence with a semantics that was introduced in (6).

Although Table 1 nicely shows how the traditional semantics of (13) plus the one from (6) fit together, it only treats semantics that can be expressed using maximality or minimality of one of the labels. In the current paper, we will therefore treat two other semantics that are not defined using maximality or minimality conditions of complete labellings: ideal semantics and stage semantics. We show that the labelling approach is nevertheless capable not only of capturing these semantics, but also in contributing to understanding what these semantics are essentially all about.

3. Ideal Semantics

The intuition behind *ideal semantics*¹ (14) can perhaps be best explained as a form of judgement aggregation(11), where a set of judgements (labellings) is combined to reach an aggregated judgement (labelling). The idea is to start with a group of people who individually try to accept as much as possible. The process of judgment aggregation then examines what they all agree on and whether the aggregated judgment (labelling) is still reasonable (admissible). If not, then some argument that were accepted (in) or rejected (out) are abstained from having an opinion about (that is, they are relabelled to undec). This process is repeated until the resulting overall judgment (labelling) is reasonable (admissible). The result is the *ideal labelling*.

¹Please notice that we do not intend to give any value judgement by adopting the term “ideal semantics”.

In order to formally define the concept of the ideal labelling, we first need to treat some preliminaries from (11).

Definition 3.1 (11:) Let $\mathcal{L}ab_1$ and $\mathcal{L}ab_2$ be labellings of argumentation framework $AF = (Ar, att)$. We say that $\mathcal{L}ab_2$ is *more or equally committed* than $\mathcal{L}ab_1$ (written as $\mathcal{L}ab_1 \sqsubseteq \mathcal{L}ab_2$) iff $\text{in}(\mathcal{L}ab_1) \subseteq \text{in}(\mathcal{L}ab_2)$ and $\text{out}(\mathcal{L}ab_1) \subseteq \text{out}(\mathcal{L}ab_2)$. We say that $\mathcal{L}ab_2$ is *compatible* with $\mathcal{L}ab_1$ (written as $\mathcal{L}ab_1 \approx \mathcal{L}ab_2$) iff $\text{in}(\mathcal{L}ab_1) \cap \text{out}(\mathcal{L}ab_2) = \emptyset$ and $\text{out}(\mathcal{L}ab_1) \cap \text{in}(\mathcal{L}ab_2) = \emptyset$.

It holds that “ \sqsubseteq ” defines a partial order (reflexive, anti-symmetric, transitive) on the labellings of an argumentation framework. We can therefore talk about a labelling being “bigger” or “smaller” than another labelling with respect to “ \sqsubseteq ”. The relation “ \approx ”, although reflexive and symmetric, is not an equivalence relation, since it does not satisfy transitivity.¹ It holds that “ \sqsubseteq ” is at least as strong as “ \approx ”; that is, if $\mathcal{L}ab_1 \sqsubseteq \mathcal{L}ab_2$ then $\mathcal{L}ab_1 \approx \mathcal{L}ab_2$.²

The idea of “ \sqsubseteq ” is to define what it means for a labelling to be more committed than another labelling. For instance, the grounded labelling is the least committed labelling among all complete labellings. The idea of “ \approx ” is to define when a labelling of one person might still be acceptable to another person. To see this, first consider that by requiring that $\text{in}(\mathcal{L}ab_1) \cap \text{out}(\mathcal{L}ab_2) = \emptyset$ and $\text{out}(\mathcal{L}ab_1) \cap \text{in}(\mathcal{L}ab_2) = \emptyset$, the relation “ \approx ” does not allow for conflicts between in and out. That is, if there is an argument that is accepted by agent A but rejected by agent B (or vice versa) then their labellings are not compatible. However, there is no problem in having conflicts between in and undec, or between out and undec. Thus, the idea of compatibility is to provide an indication of how easy or difficult it is to have to defend a position that is not one’s own. It is easier to do this for a labelling that is compatible than for a labelling that is not compatible. In the former case the worst that can happen is that one has to abstain from something one accepts or rejects (or have to accept or reject something where one did not have an explicit opinion about). In the latter case, however, one has to make statements that go directly against one’s private position.

To come back to the informal description of ideal semantics, we assume a meeting in which every preferred labelling is represented. The meeting then discusses each argument, one by one, with the aim to define an *initial labelling*. If everybody agrees that the argument is labelled in (that is, the argument is labelled in in every preferred labelling) then the argument is also labelled in in the initial labelling. If everybody agrees that the argument is labelled out (that is, the argument is labelled out in every preferred labelling) then the argument is labelled out in the initial labelling. In all other cases, the argument is labelled undec in the initial labelling. After this process is over, and the initial labelling has been finished, the meeting goes to the second phase, in which the initial labelling is “watered down” in order to become an admissible labelling. This is done by iteratively relabelling each argument that is illegally in or illegally out to undec. When there are no more arguments left that are illegally in or illegally out, the result is the *ideal labelling*. It was proved in (11) that this process results in constructing the most committed (“biggest”) labelling that is less or equally committed to each preferred labelling. This leads to the following definition of ideal semantics.

Definition 3.2: Let $AF = (Ar, att)$ be an argumentation framework. The ideal

¹As a counterexample, consider an argumentation framework $AF = (\{A, B\}, \{(A, B), (B, A)\})$. Let $\mathcal{L}ab_1 = (\{A\}, \{B\}, \emptyset)$, $\mathcal{L}ab_2 = (\emptyset, \emptyset, \{A, B\})$ and $\mathcal{L}ab_3 = (\{B\}, \{A\}, \emptyset)$. It holds that $\mathcal{L}ab_1 \approx \mathcal{L}ab_2$ and $\mathcal{L}ab_2 \approx \mathcal{L}ab_3$ but $\mathcal{L}ab_1 \not\approx \mathcal{L}ab_3$.

²This is because $\mathcal{L}ab_1 \approx \mathcal{L}ab_2$ iff $\text{in}(\mathcal{L}ab_1) \subseteq \text{in}(\mathcal{L}ab_2) \cup \text{undec}(\mathcal{L}ab_2)$ and $\text{out}(\mathcal{L}ab_1) \subseteq \text{out}(\mathcal{L}ab_2) \cup \text{undec}(\mathcal{L}ab_2)$.

labelling is the biggest admissible labelling that is smaller or equal to each preferred labelling.

The uniqueness of the ideal labelling has been proved in (11).¹ It also holds that the ideal labelling is a complete labelling (11). Since the grounded labelling is the smallest complete labelling (w.r.t. “ \approx ”) it directly follows that the ideal labelling is bigger or equal to the grounded labelling.

Proposition 3.3: *Let (Ar, att) be an argumentation framework, let $\mathcal{L}ab_{grounded}$ be its grounded extension and $\mathcal{L}ab_{ideal}$ be its ideal extension. It holds that $\mathcal{L}ab_{grounded} \sqsubseteq \mathcal{L}ab_{ideal}$.*

In the current paper, we extend the results of (11) by proving that the ideal labelling is also the *biggest* admissible labelling that is compatible with each admissible/complete/preferred labelling.

Theorem 3.4: *Let $\mathcal{L}ab$ be a labelling of argumentation framework $AF = (Ar, att)$. The following statements are equivalent.*

- (1) $\mathcal{L}ab$ is the biggest admissible labelling that is smaller or equal to each preferred labelling (that is, $\mathcal{L}ab$ is the ideal labelling)
- (2) $\mathcal{L}ab$ is the biggest admissible labelling that is compatible with each admissible labelling
- (3) $\mathcal{L}ab$ is the biggest admissible labelling that is compatible with each complete labelling
- (4) $\mathcal{L}ab$ is the biggest admissible labelling that is compatible with each preferred labelling

Proof: *From 1 to 2:*

Let $\mathcal{L}ab$ be the biggest admissible labelling that is smaller or equal to each preferred labelling. Then $\mathcal{A}rgs = \text{in}(\mathcal{L}ab)$ is the biggest admissible set that is a subset of each preferred extension. This can be seen as follows. First of all, $\mathcal{A}rgs$ is an admissible set that is a subset of each preferred extension (this is because from $\mathcal{L}ab \sqsubseteq \mathcal{L}ab'$ it follows that $\text{in}(\mathcal{L}ab) \subseteq \text{in}(\mathcal{L}ab')$). We now prove that $\mathcal{A}rgs$ is also the *biggest* admissible set that is a subset of each preferred extension. Let $\mathcal{A}rgs'$ be an admissible set that is a subset of each preferred extension. Then $\mathcal{L}ab' = \text{Ext2Lab}(\mathcal{A}rgs')$ is an admissible labelling that is smaller or equal to each preferred labelling (this follows from Lemma 2.9). So $\mathcal{L}ab' \sqsubseteq \mathcal{L}ab$ (this is because $\mathcal{L}ab$ is the biggest admissible labelling that is smaller or equal to each preferred labelling). So $\text{in}(\mathcal{L}ab') \subseteq \text{in}(\mathcal{L}ab)$ so $\mathcal{A}rgs' \subseteq \mathcal{A}rgs$. So $\mathcal{A}rgs$ is indeed the *biggest* (w.r.t. set-inclusion) admissible set that is a subset of each preferred extension.

We can then apply the results from (14, Theorem 3.3): If $\mathcal{A}rgs$ is the biggest admissible set that is a subset of each preferred extension, then $\mathcal{A}rgs$ is the biggest admissible set that is not attacked by any admissible set. We now prove that $\mathcal{L}ab$ is an admissible labelling that is compatible with each admissible labelling. Let $\mathcal{L}ab_{adm}$ be an arbitrary admissible labelling and let $\mathcal{A}rgs_{adm} = \text{in}(\mathcal{L}ab_{adm})$. We now prove the following.

- (1) $\text{in}(\mathcal{L}ab) \cap \text{out}(\mathcal{L}ab_{adm}) = \emptyset$
This follows from the fact that $\mathcal{A}rgs_{adm}$ does not attack $\mathcal{A}rgs$.
- (2) $\text{out}(\mathcal{L}ab) \cap \text{in}(\mathcal{L}ab_{adm}) = \emptyset$
This follows from the fact that for admissible sets the attack relation is symmetrical.² So from the fact that $\mathcal{A}rgs_{adm}$ does not attack $\mathcal{A}rgs$ it follows

¹The idea is to perform the sceptical judgment aggregation procedure of (11) on all preferred labellings.

²That is, if $\mathcal{A}rgs_1$ and $\mathcal{A}rgs_2$ are admissible sets and $\mathcal{A}rgs_1$ attacks $\mathcal{A}rgs_2$ then $\mathcal{A}rgs_2$ also attacks $\mathcal{A}rgs_1$.

that $Args$ does not attack $Args_{adm}$.

We now prove that $\mathcal{L}ab$ is also the *biggest* admissible labelling that is compatible with each admissible labelling. Let $\mathcal{L}ab'$ be an admissible labelling that is compatible with each admissible labelling. Then $Args' = \text{in}(\mathcal{L}ab')$ is an admissible set that is not attacked by any admissible set. So $Args' \subseteq Args$. It then follows that also $\mathcal{L}ab' \sqsubseteq \mathcal{L}ab$. This can be seen as follows.

- $\text{in}(\mathcal{L}ab') \subseteq \text{in}(\mathcal{L}ab)$
This is because $Args' \subseteq Args$ and $\text{in}(\mathcal{L}ab') = Args'$ and $\text{in}(\mathcal{L}ab) = Args$
- $\text{out}(\mathcal{L}ab') \subseteq \text{out}(\mathcal{L}ab)$
Let $A \in \text{out}(\mathcal{L}ab')$. Then there is an argument $B \in \text{in}(\mathcal{L}ab')$ that attacks A . From the fact that $\text{in}(\mathcal{L}ab') \subseteq \text{in}(\mathcal{L}ab)$ it follows that $B \in \text{in}(\mathcal{L}ab)$. From the fact that $\mathcal{L}ab$ is a complete labelling (see (11)) it follows that $A \in \text{out}(\mathcal{L}ab)$.

So $\mathcal{L}ab$ is the biggest admissible labelling that is compatible with each admissible labelling.

From 2 to 1:

Let $\mathcal{L}ab$ be the biggest admissible labelling that is compatible with each admissible labelling. Let $\mathcal{L}ab'$ be the biggest admissible labelling that is smaller or equal to each preferred labelling. It then follows (“from 1 to 2”) that $\mathcal{L}ab'$ is also the biggest admissible labelling that is compatible with each admissible labelling. From the uniqueness of the biggest admissible labelling that is compatible with each admissible labelling (which is basically the biggest element of a partially ordered set) it then follows that $\mathcal{L}ab' = \mathcal{L}ab$.

From 2 to 3:

Let $\mathcal{L}ab$ be the biggest admissible labelling that is compatible with each admissible labelling. From the fact that each complete labelling is also an admissible labelling, it follows that $\mathcal{L}ab$ is also compatible with each complete labelling. We now prove that $\mathcal{L}ab$ is also the *biggest* admissible labelling that is compatible with each complete labelling. Let $\mathcal{L}ab'$ be an arbitrary admissible labelling that is compatible with each complete labelling. We will now prove that $\mathcal{L}ab' \sqsubseteq \mathcal{L}ab$. We do this by proving that $\mathcal{L}ab'$ is compatible with each admissible labelling. Let $\mathcal{L}ab''$ be an arbitrary admissible labelling. Let $\mathcal{L}ab'''$ be the up-complete labelling of $\mathcal{L}ab''$, as defined in (11). It then holds that $\mathcal{L}ab'''$ is a complete labelling with $\mathcal{L}ab'' \sqsubseteq \mathcal{L}ab'''$ (11). From the fact that $\mathcal{L}ab'$ is compatible with each complete labelling it then follows that $\mathcal{L}ab'$ is compatible with $\mathcal{L}ab'''$. From this, together with the fact that $\mathcal{L}ab'' \sqsubseteq \mathcal{L}ab'''$, it follows that $\mathcal{L}ab'$ is compatible with $\mathcal{L}ab''$. Since this holds for any arbitrary admissible labelling $\mathcal{L}ab''$, it follows that $\mathcal{L}ab'$ is compatible with each admissible labelling. Since $\mathcal{L}ab$ is the *biggest* admissible labelling that is compatible with each admissible labelling, it then follows that $\mathcal{L}ab' \sqsubseteq \mathcal{L}ab$.

From 3 to 2:

This follows from “from 2 to 3” together with the uniqueness of 3. The proof is similar to the proof of “from 2 to 1”.

From 3 to 4:

Let $\mathcal{L}ab$ be the biggest admissible labelling that is compatible with each complete labelling. From the fact that each preferred labelling is a complete labelling, it follows that $\mathcal{L}ab$ is also compatible with each preferred labelling. We now prove that $\mathcal{L}ab$ is also the *biggest* admissible labelling that is compatible with each preferred labelling. Let $\mathcal{L}ab'$ be an arbitrary admissible labelling that is compatible with each preferred labelling. We will now prove that $\mathcal{L}ab' \sqsubseteq \mathcal{L}ab$. We do this by proving that $\mathcal{L}ab'$ is compatible with each complete labelling. Let $\mathcal{L}ab''$ be an arbitrary complete labelling. Let $\mathcal{L}ab'''$ be a preferred labelling such that $\mathcal{L}ab'' \sqsubseteq \mathcal{L}ab'''$ (the existence of such a $\mathcal{L}ab'''$ follows from the fact that for every complete extension E'' there

exists a preferred extension E''' with $E'' \subseteq E'''$ (13), which together with Lemma 2.9 implies that $\text{Ext2Lab}(E'') \sqsubseteq \text{Ext2Lab}(E''')$. From this, together with the fact that $\mathcal{L}ab'$ is compatible with $\mathcal{L}ab'''$ (because $\mathcal{L}ab'$ is compatible with each preferred labelling) it follows that $\mathcal{L}ab'$ is compatible with $\mathcal{L}ab''$. Since this holds for any arbitrary complete labelling $\mathcal{L}ab''$, it follows that $\mathcal{L}ab'$ is compatible with each complete labelling. Since $\mathcal{L}ab$ is the biggest admissible labelling that is compatible with each complete labelling it then follows that $\mathcal{L}ab' \sqsubseteq \mathcal{L}ab$.

From 4 to 3:

This follows from “from 3 to 4” together with the uniqueness of 4. The proof is similar to the proof of “from 2 to 1”. \square

We will now treat the notion of ideal semantics as described in (14).¹

Definition 3.5: Let $AF = (Ar, att)$ be an argumentation framework. An admissible set $Args$ is called *ideal* iff it is a subset of each preferred extension. The ideal extension is a maximal (w.r.t. set-inclusion) ideal set.

It turns out that the ideal extension is unique (which implies that it is also the biggest ideal set) and that it is also a complete extension (14).

There are several ways of describing the ideal extension.

Proposition 3.6: Let $AF = (Ar, att)$ be an argumentation framework, and let $Args \subseteq Ar$. The following statements are equivalent.

- (1) $Args$ is the biggest admissible set that is a subset of each preferred extension (that is, $Args$ is the ideal extension)
- (2) $Args$ is the biggest admissible set that is not attacked by any admissible set
- (3) $Args$ is the biggest admissible set that is not attacked by any complete extension
- (4) $Args$ is the biggest admissible set that is not attacked by any preferred extension

In Proposition 3.6 the equivalence between points 1 and 2 follows from (14, Theorem 3.3). The equivalence between points 2, 3 and 4 follows from the fact that an argument (or set) is attacked by an admissible set iff it is attacked by a complete extension iff it is attacked by a preferred extension.

It turns out that the ideal labelling corresponds to the ideal extension.²

Theorem 3.7: Let $AF = (Ar, att)$ be an argumentation framework. It holds that:

- (1) if $Args$ is the ideal extension of AF then $\text{Ext2Lab}(Args)$ is the ideal labelling of AF , and
- (2) if $\mathcal{L}ab$ is the ideal labelling of AF then $\text{Lab2Ext}(\mathcal{L}ab)$ is the ideal extension of AF .

Proof:

- (1) Let $Args$ be the ideal extension of AF . From Proposition 3.6 it follows that $Args$ is the biggest admissible set that is not attacked by any complete extension. Since the ideal extension is also a complete extension, it follows that $Args$ is the biggest complete extension that is not attacked by any complete extension. Since attacks among admissible sets are symmetric (and therefore attacks among complete extensions are also symmetric) it

¹We use the term *ideal extension* to refer to the biggest ideal set.

²Point 2 (but not point 1) of Theorem 3.7 has been proved in (11).

follows that $Args$ is also the biggest complete extension that does not attack any complete extension. It then follows that $Ext2Lab(Args)$ is the biggest complete labelling that is compatible with each complete labelling. It then follows that $Ext2Lab(Args)$ is also the biggest admissible labelling that is compatible with any complete labelling (this follows from point 3 of Theorem 3.4, together with the uniqueness of the ideal labelling and the fact that the ideal labelling is a complete labelling). Therefore, $Ext2Lab(Args)$ is the ideal labelling.

- (2) Let Lab be the ideal labelling of AF . From the fact that the ideal labelling is a complete labelling, together with the one-to-one relationship between complete labellings and complete extensions, it follows that $Ext2Lab(Lab2Ext(Lab)) = Lab$. Let $Args = Lab2Ext(Lab)$. From the fact that $Ext2Lab(Args) = Lab$, together with the fact that the ideal extension is unique, it follows (from point 1) that $Args$ is the ideal extension.

□

Since for complete semantics, the relation between labellings and extensions is one-to-one, it follows that for ideal semantics the relation is one-to-one as well (this is because the ideal extension/labelling is also a complete extension/labelling).

4. Stage Semantics

The concept of stage semantics goes back to work of Verheij (22). In essence, a stage extension can be described as a conflict-free set of arguments $Args$, where $Args \cup Args^+$ is maximal w.r.t. set-inclusion.¹

In order to describe a stage extension in terms of labellings, we first introduce the concept of a conflict-free labelling

Definition 4.1: Let Lab be a labelling of argumentation framework $AF = (Ar, att)$. Lab is *conflict-free* iff for each $A \in Ar$ it holds that:

- (1) if A is labelled in then it does not have an attacker that is labelled in
- (2) if A is labelled out then it has at least one attacker that is labelled in

When comparing a conflict-free labelling to an admissible labelling it can be noticed that the condition on out labelled arguments (point 2) is essentially the same. However, the condition for in-labelled arguments (point 1) is weaker for conflict-free labellings than for admissible labellings. It then follows that every admissible labelling is also a conflict-free labelling (just like every admissible set is also a conflict-free set by definition).

The idea of a *stage labelling* can then be described as a conflict-free labelling where undec is minimal.

Definition 4.2: Let $AF = (Ar, att)$ be an argumentation framework. A labelling Lab is called a *stage labelling* iff it is a conflict-free labelling where $undec(Lab)$ is minimal (w.r.t. set-inclusion) among all conflict-free labellings.

It holds that every stable labelling is also a stage labelling.²

Theorem 4.3: Let Lab be a labelling of argumentation framework $AF = (Ar, att)$. If Lab is a stable labelling of AF then Lab is also a stage labelling of AF .

¹Recall that $Args^+$ stands for the set of arguments attacked by $Args$.

²A similar observation was made in (22).

Proof: Let $\mathcal{L}ab$ be a stable labelling of AF . Then $\mathcal{L}ab$ is a complete labelling with $\text{undec}(\mathcal{L}ab) = \emptyset$. From the fact that $\mathcal{L}ab$ is a complete labelling (points 1 and 2 from Proposition 2.8), it follows (Definition 4.2) that $\mathcal{L}ab$ is a conflict-free labelling (points 1 and 2 from Definition 4.1). Since $\text{undec}(\mathcal{L}ab) = \emptyset$, it directly follows that $\text{undec}(\mathcal{L}ab)$ is minimal among all conflict-free labellings. Hence, $\mathcal{L}ab$ is a stage labelling. \square

If there exists at least one stable labelling, then each stage labelling is also a stable labelling.¹

Theorem 4.4: *Let $AF = (Ar, att)$ be an argumentation framework. If there exists at least one stable labelling of AF then every stage labelling is also a stable labelling.*

Proof: Let $\mathcal{L}ab$ be a stable labelling of AF . Then from Theorem 4.3 it follows that $\mathcal{L}ab$ is also a stage labelling. In particular, $\mathcal{L}ab$ is a stage labelling with $\text{undec}(\mathcal{L}ab) = \emptyset$. Now, let $\mathcal{L}ab'$ be an arbitrary stage labelling of AF . It then follows that $\text{undec}(\mathcal{L}ab')$ is minimal among all conflict-free labellings. Since $\mathcal{L}ab$ is a conflict-free labelling with $\text{undec}(\mathcal{L}ab) = \emptyset$ it follows that $\text{undec}(\mathcal{L}ab')$ must also be \emptyset . So $\mathcal{L}ab'$ is a conflict-free labelling with $\text{undec}(\mathcal{L}ab') = \emptyset$. In order to show that $\mathcal{L}ab'$ is a stable labelling, we first show that it is a complete labelling, hence that the three points of Definition 2.7 are satisfied:

- (1) “if A is labelled in then all attackers of A are labelled out”

This follows from point 1 of Definition 4.1, together with the fact that $\text{undec}(\mathcal{L}ab) = \emptyset$ (therefore from the fact that A does not have any defeaters that are labelled in it follows that all defeaters of A are labelled out).

- (2) “if A is labelled out then all attackers of A are labelled in”

This follows from point 2 of Definition 4.1.

- (3) “if A is labelled undec then not all attackers of A are labelled out and A does not have an attacker that is labelled in”

This is trivially satisfied, since $\text{undec}(\mathcal{L}ab') = \emptyset$.

So $\mathcal{L}ab'$ is a complete labelling with $\text{undec}(\mathcal{L}ab') = \emptyset$. Hence (Definition 2.11) $\mathcal{L}ab'$ is a stable labelling. \square

There also exists an alternative way to describe the concept of a stage labelling. In essence, what a stage labelling does is to take a maximal subgraph of the argumentation framework that has a stable labelling. A stage labelling is then a stable labelling of such a maximal subgraph, augmented with undec labels for the arguments that did not make their way into the subgraph.

In the following lemma, $AF|_{\mathcal{A}rgs}$ stands for $(Ar \cap \mathcal{A}rgs, att \cap (\mathcal{A}rgs \times \mathcal{A}rgs))$ (assuming that $AF = (Ar, att)$) and $\mathcal{L}ab|_{\mathcal{A}rgs}$ stands for $\mathcal{L}ab \cap (\mathcal{A}rgs \times \{\text{in}, \text{out}, \text{undec}\})$.

Lemma 4.5: *Let $\mathcal{L}ab$ be a labelling of argumentation framework $AF = (Ar, att)$. The following two statements are equivalent.*

- (1) $\mathcal{L}ab$ is a conflict-free labelling.
- (2) $\mathcal{L}ab|_{\mathcal{A}rgs}$ is a stable labelling of $AF|_{\mathcal{A}rgs}$, where $\mathcal{A}rgs = \text{in}(\mathcal{L}ab) \cup \text{out}(\mathcal{L}ab)$.

Proof:

from 1 to 2:

Let $\mathcal{L}ab$ be a conflict-free labelling. We now prove that $\mathcal{L}ab|_{\mathcal{A}rgs}$ is a stable labelling of $AF|_{\mathcal{A}rgs}$. We do this by proving that $\mathcal{L}ab|_{\mathcal{A}rgs}$ fulfils the three conditions of a complete labelling (Definition 2.7).

¹A similar observation, in the context of DEFLOG, was made in (23, page 337).

- (1) “If A is labelled in then all attackers of A are labelled out”
This follows from point 1 of Definition 4.1, together with the fact that $\text{undec}(\mathcal{L}ab|_{\mathcal{A}rgs}) = \emptyset$.
- (2) “if A is labelled out then A has an attacker that is labelled in”
This follows from point 2 of Definition 4.1.
- (3) “if A is labelled undec then not all attackers of A are labelled out and A does not have an attacker that is labelled in”
This is trivially satisfied since $\text{undec}(\mathcal{L}ab|_{\mathcal{A}rgs}) = \emptyset$.

From the validity of point 1, 2 and 3 it follows that $\mathcal{L}ab|_{\mathcal{A}rgs}$ is a complete labelling of $AF|_{\mathcal{A}rgs}$. From the fact that $\text{undec}(\mathcal{L}ab|_{\mathcal{A}rgs}) = \emptyset$ it follows that $\mathcal{L}ab|_{\mathcal{A}rgs}$ is also a stable labelling of $AF|_{\mathcal{A}rgs}$.

from 2 to 1:

Let $\mathcal{L}ab$ be a labelling of AF such that $\mathcal{L}ab|_{\mathcal{A}rgs}$ is a stable labelling of $AF|_{\mathcal{A}rgs}$. We now prove that $\mathcal{L}ab$ is a conflict-free labelling of AF . According to Definition 4.1 we need to prove two things:

- (1) “if A is labelled in then it does not have an attacker that is labelled in”
Let B be an arbitrary attacker of A in AF . We distinguish two cases:
 - a) $B \in \mathcal{A}rgs$. Then B also attacks A in $AF|_{\mathcal{A}rgs}$. From the fact that $\mathcal{L}ab|_{\mathcal{A}rgs}$ is a stable labelling (and therefore also a complete labelling) of $AF|_{\mathcal{A}rgs}$ it follows that B is labelled out by $\mathcal{L}ab|_{\mathcal{A}rgs}$, and therefore that B is labelled out by $\mathcal{L}ab$.
 - b) $B \notin \mathcal{A}rgs$. Then $B \notin \text{in} \cup \text{out}$ so $B \in \text{undec}(\mathcal{L}ab)$.
 In both cases, B is not labelled in by $\mathcal{L}ab$. Therefore A does not have an attacker in AF that is labelled in by $\mathcal{L}ab$.
- (2) “If A is labelled out then it has at least one attacker that is labelled in”
Let A be an argument that is labelled out by $\mathcal{L}ab$. Then $A \in \mathcal{A}rgs$, so the fact that $\mathcal{L}ab|_{\mathcal{A}rgs}$ is a stable (and therefore also complete) labelling of $AF|_{\mathcal{A}rgs}$ implies that there exists an argument B labelled in by $\mathcal{L}ab|_{\mathcal{A}rgs}$ that attacks A in $AF|_{\mathcal{A}rgs}$, so B is also labelled in by $\mathcal{L}ab$ and also attacks A in AF .

□

Theorem 4.6: *Let $\mathcal{L}ab$ be a labelling of argumentation framework $AF = (Ar, att)$. The following two statements are equivalent.*

- (1) $\mathcal{L}ab$ is a conflict-free labelling where $\text{undec}(\mathcal{L}ab)$ is minimal (w.r.t. set inclusion) among all conflict-free labellings (that is, $\mathcal{L}ab$ is a stage labelling)
- (2) $\mathcal{A}rgs = \text{in}(\mathcal{L}ab) \cup \text{out}(\mathcal{L}ab)$ is a maximal subset of Ar such that $AF|_{\mathcal{A}rgs}$ has a stable labelling, and $\mathcal{L}ab|_{\mathcal{A}rgs}$ is a stable labelling of $AF|_{\mathcal{A}rgs}$.

Proof:

from 1 to 2:

Let $\mathcal{L}ab$ be a conflict-free labelling where $\text{undec}(\mathcal{L}ab)$ is minimal. From Lemma 4.5 it follows that $\mathcal{L}ab|_{\mathcal{A}rgs}$ is a stable labelling of $AF|_{\mathcal{A}rgs}$. We now prove that $\mathcal{A}rgs$ is a maximal subset of Ar such that $AF|_{\mathcal{A}rgs}$ has a stable labelling. Suppose $\mathcal{A}rgs' \supsetneq \mathcal{A}rgs$ is a subset of Ar such that $AF|_{\mathcal{A}rgs'}$ has a stable labelling (say $\mathcal{L}ab'$). Let $\mathcal{L}ab'' = \mathcal{L}ab \cup \{(A, \text{undec}) \mid A \in Ar \setminus \mathcal{A}rgs'\}$. It holds that $\mathcal{L}ab' = \mathcal{L}ab''|_{\mathcal{A}rgs'}$. From Lemma 4.5 it follows that $\mathcal{L}ab''$ is a conflict-free labelling of AF . Moreover, it holds that $\text{undec}(\mathcal{L}ab'') \subsetneq \text{undec}(\mathcal{L}ab)$, so $\text{undec}(\mathcal{L}ab)$ is not minimal. Contradiction.

from 2 to 1:

Let $\mathcal{A}rgs = \text{in}(\mathcal{L}ab) \cup \text{out}(\mathcal{L}ab)$ be a maximal subset of Ar such that $AF|_{\mathcal{A}rgs}$ has a stable labelling, and that $\mathcal{L}ab|_{\mathcal{A}rgs}$ is a stable labelling of $AF|_{\mathcal{A}rgs}$. From

Lemma 4.5 it follows that $\mathcal{L}ab$ is a conflict-free labelling of AF . We now prove that $\text{undec}(\mathcal{L}ab)$ is minimal. Let $\mathcal{L}ab'$ be a conflict-free labelling of AF with $\text{undec}(\mathcal{L}ab') \subsetneq \text{undec}(\mathcal{L}ab)$ and let $\mathcal{A}rgs' = \text{in}(\mathcal{L}ab') \cup \text{out}(\mathcal{L}ab')$. Then, according to Lemma 4.5 it follows that $\mathcal{L}ab'|_{\mathcal{A}rgs'}$ is a stable labelling of $AF|_{\mathcal{A}rgs'}$. However, since $\mathcal{A}rgs' \supsetneq \mathcal{A}rgs$ it follows that $\mathcal{A}rgs$ is not a maximal subset of Ar such that $AF|_{\mathcal{A}rgs}$ has a stable labelling. Contradiction. \square

As was mentioned at the beginning of the current section, stage semantics can also be expressed using the traditional approach of argument extensions.

Definition 4.7: Let $AF = (Ar, att)$ be an argumentation framework. A stage extension is a conflict-free set $\mathcal{A}rgs \subseteq Ar$ where $\mathcal{A}rgs \cup \mathcal{A}rgs^+$ is maximal (w.r.t. set inclusion) among all conflict-free sets.

Lemma 4.8: Let $AF = (Ar, att)$ be an argumentation framework and $\mathcal{A}rgs \subseteq Ar$. The following two statements are equivalent:

- (1) $\mathcal{A}rgs$ is a conflict-free set of AF
- (2) $\mathcal{A}rgs$ is a stable extension of $AF|_{\mathcal{A}rgs \cup \mathcal{A}rgs^+}$

Proof: Similar to Lemma 4.5. \square

Theorem 4.9: Let $AF = (Ar, att)$ be an argumentation framework and $\mathcal{A}rgs \subseteq Ar$. The following two statements are equivalent.

- (1) $\mathcal{A}rgs$ is a conflict-free set where $\mathcal{A}rgs \cup \mathcal{A}rgs^+$ is maximal (w.r.t. set inclusion) among all conflict-free sets (that is, $\mathcal{A}rgs$ is a stage extension).
- (2) $\mathcal{A}rgs \cup \mathcal{A}rgs^+$ is a maximal subset of Ar such that $AF|_{\mathcal{A}rgs \cup \mathcal{A}rgs^+}$ has a stable extension, and $\mathcal{A}rgs$ is a stable extension of $AF|_{\mathcal{A}rgs \cup \mathcal{A}rgs^+}$.

Proof: Similar to the proof of Theorem 4.6. \square

Before discussing the connection between stage labellings and stage extensions, we first state the connection between conflict-free labellings and conflict-free sets.

Proposition 4.10: Let $AF = (Ar, att)$ be an argumentation framework. It holds that:

- (1) if $\mathcal{A}rgs$ is a conflict-free set of AF then $\text{Ext2Lab}(\mathcal{A}rgs)$ is a conflict-free labelling of AF
- (2) if $\mathcal{L}ab$ is a conflict-free labelling of AF then $\text{Lab2Ext}(\mathcal{L}ab)$ is a conflict-free set of AF

The relation between conflict-free sets and conflict-free labellings is not one-to-one, however. There can be several conflict-free labellings which are associated with the same conflict-free set.¹

Theorem 4.11: Let $AF = (Ar, att)$ be an argumentation framework. It holds that:

- (1) if $\mathcal{A}rgs$ is a stage extension of AF then $\text{Ext2Lab}(\mathcal{A}rgs)$ is a stage labelling of AF
- (2) if $\mathcal{L}ab$ is a stage labelling of AF then $\text{Lab2Ext}(\mathcal{L}ab)$ is a stage extension of AF

Proof: This follows from Proposition 4.10. \square

¹An example would be an argumentation framework $(\{A, B\}, \{(A, B)\})$ with $\mathcal{L}ab_1 = (\{A\}, \{B\}, \emptyset)$ and $\mathcal{L}ab_2 = (\{A\}, \emptyset, \{B\})$.

Stage extensions and stage labellings stand in a one-to-one relation to each other. That is, when restricted to stage extensions and stage labellings, the functions Ext2Lab and Lab2Ext become bijections and each other's inverse.

Theorem 4.12: *Let $AF = (Ar, att)$ be an argumentation framework, se its set of stage extensions and sl its set of stage labellings. Let $\text{Ext2Lab}^s : se \rightarrow sl$ be a function such that $\text{Ext2Lab}^s(\mathcal{A}rgs) = \text{Ext2Lab}(\mathcal{A}rgs)$ (that is, Ext2Lab^s is the function Ext2Lab where the domain/range is restricted to stage extensions/labellings) and $\text{Lab2Ext}^s : sl \rightarrow se$ be a function such that $\text{Lab2Ext}^s(\mathcal{L}ab) = \text{Lab2Ext}(\mathcal{L}ab)$ (that is, Lab2Ext^s is the function Lab2Ext where the domain/range is restricted to stage labellings/extensions). The functions Ext2Lab^s and Lab2Ext^s are bijective and each other's inverse.*

Proof: As every function that has an inverse is bijective, we only need to prove that Lab2Ext^s and Ext2Lab^s are each other's inverses. That is $(\text{Lab2Ext}^s)^{-1} = \text{Ext2Lab}^s$ and $(\text{Ext2Lab}^s)^{-1} = \text{Lab2Ext}^s$. For this, we prove the following two things:

- (1) “For every stage labelling $\mathcal{L}ab$ it holds that $\text{Ext2Lab}^s(\text{Lab2Ext}^s(\mathcal{L}ab)) = \mathcal{L}ab$.”

Let $\mathcal{L}ab$ be a stage labelling of AF . Let $\mathcal{A}rgs = \text{in}(\mathcal{L}ab) \cup \text{out}(\mathcal{L}ab)$. Then according to Theorem 4.6 $\mathcal{L}ab|_{\mathcal{A}rgs}$ is a stable labelling of $AF|_{\mathcal{A}rgs}$ and since stable labellings and stable extensions are one-to-one related (this follows from the fact that complete labellings and complete extensions are one-to-one related, as proved in (10)) it follows that $\text{Ext2Lab}(\text{Lab2Ext}(\mathcal{L}ab|_{\mathcal{A}rgs})) = \mathcal{L}ab|_{\mathcal{A}rgs}$. It then follows that for every $A \in Ar$:

$\text{Ext2Lab}^s(\text{Lab2Ext}^s(\mathcal{L}ab))(A) = \text{in}$ iff $\mathcal{L}ab(A) = \text{in}$

$\text{Ext2Lab}^s(\text{Lab2Ext}^s(\mathcal{L}ab))(A) = \text{out}$ iff $\mathcal{L}ab(A) = \text{out}$

It then directly follows that: $\text{Ext2Lab}^s(\text{Lab2Ext}^s(\mathcal{L}ab))(A) = \text{undec}$ iff $\mathcal{L}ab(A) = \text{undec}$

So $\text{Ext2Lab}^s(\text{Lab2Ext}^s(\mathcal{L}ab)) = \mathcal{L}ab$.

- (2) “For every stage extension E it holds that $\text{Lab2Ext}^s(\text{Ext2Lab}^s(E)) = E$.”

Let E be a stage extension of AF . We now prove two things:

- a) $\text{Lab2Ext}^s(\text{Ext2Lab}^s(E)) \subseteq E$

Let $A \in \text{Lab2Ext}^s(\text{Ext2Lab}^s(E)) \subseteq E$. Then A is labelled in by $\text{Ext2Lab}^s(E)$. Therefore $A \in E$.

- b) $E \subseteq \text{Lab2Ext}^s(\text{Ext2Lab}^s(E))$

Let $A \in E$. Then A is labelled in by $\text{Ext2Lab}^s(E)$. Therefore $A \in \text{Lab2Ext}^s(\text{Ext2Lab}^s(E))$.

□

In order to understand the difference between stage semantics and the admissibility based semantics, it is useful to make an analogy with classical logic. In the presence of a potentially inconsistent knowledge base one could do two things:

- (1) Take the maximal consistent subsets of the knowledge base, and examine what is entailed by all of these. That is, take the (classical) models of the maximal subsets of the knowledge base that have classical models.
- (2) Define a new semantics such that the entire knowledge base will have models. This is the approach that is, for instance, taken in the field of paraconsistent logic (2, 12).

Solution 1 (applying the original semantics to maximal subsets of the original problem description) is comparable to stage semantics, whereas solution 2 (redefining

the semantics so that it can meaningfully be applied to a bigger class of knowledge bases) is comparable with the admissibility based semantics.

4.1. Stage Semantics versus Semi-Stable Semantics

The concept of stage semantics is similar to that of semi-stable semantics, as both of these semantics aim to maximize the range ($Args \cup Args^+$) of their extensions.¹ However, where semi-stable semantics aims to maximize the range under the condition of admissibility, stage semantics tries to maximize the range under the weaker condition of conflict-freeness. It turns out that the approach of stage semantics is equivalent to taking the stable extensions (labellings) of the biggest subframework that has at least one stable extension (labelling). Hence, the approach of stage semantics is comparable with the approach of handling inconsistent knowledge bases, where one can select maximal consistent subsets of the knowledge base, and then examine what holds in all of them (in the union of all their models). That is, it is as if stage semantics interprets the absence of stable extensions as some form of “inconsistency”, which needs to be handled taking the “maximal consistent subframeworks”.

In semi-stable semantics, on the other hand, one still takes all the arguments of the argumentation framework into account. That is, one does not ignore any information provided by the argumentation framework. An example is shown in Figure 1. Here, semi-stable semantics yields a single extension \emptyset , corresponding with a labelling $(\emptyset, \emptyset, \{A, B\})$, whereas stage semantics yields a single extension $\{B\}$, corresponding with a labelling $(\{B\}, \emptyset, \{A\})$. In essence, what stage semantics does is to ignore argument A , since this argument causes the framework not to have any stable extensions.



Figure 1. Stage semantics differs from semi-stable semantics

Another example to illustrate the difference between stage semantics and semi-stable semantics is given in Figure 2. Here, semi-stable semantics yields a single extension $\{A\}$, corresponding to a labelling $(\{A\}, \{B\}, \{C\})$. Stage semantics yields two extensions, the first one being equivalent to the one yielded by semi-stable semantics, the second one being $\{B\}$, corresponding to a labelling $(\{B\}, \{C\}, \{A\})$. The first stage extension (labelling) is the result of ignoring argument C , the second stage extension (labelling) is the result of ignoring argument A . For both possibilities, the remaining argumentation framework is a maximal one that has at least one stable extension (labelling). It can therefore be observed that under stage semantics, even an argument without any attackers (like argument A in Figure 2) is not always labelled in.² With semi-stable semantics, however, an argument without any attackers is *always* labelled in.

Although examples like illustrated by Figure 2 pose a serious problem for stage semantics, the situation is not hopeless. A positive aspect is that each argumentation framework has at least one stage labelling that is at least as committed (“bigger or equal”) as the grounded labelling. It then directly follows that each argumentation

¹Or, alternatively, they try to minimize the set of undec-labelled arguments of their labellings.

²We would like to thank Pietro Baroni and Massimiliano Giacomin for this observation.

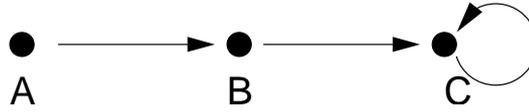


Figure 2. Stage semantics does not always endorse non-attacked arguments

framework has at least one stage labelling that labels each argument without any attackers in. In order to prove this, we need the following lemma. In this lemma, as well as in the subsequent theorem, the *range* of a labelling stands for its set of in-labelled or out-labelled arguments (that is, $\text{range}(\mathcal{L}ab) = \text{in}(\mathcal{L}ab) \cup \text{out}(\mathcal{L}ab)$).

Lemma 4.13: *Let AF be an argumentation framework with grounded labelling \mathcal{G} such that $\text{undec}(\mathcal{G}) = \emptyset$. Let \mathcal{C} be an arbitrary conflict-free labelling of AF . It holds that if $\text{range}(\mathcal{G}) \subseteq \text{range}(\mathcal{C})$ then $\mathcal{C} = \mathcal{G}$.*

Proof: From the fact that $\text{range}(\mathcal{G}) \subseteq \text{range}(\mathcal{C})$, together with the fact that $\text{undec}(\mathcal{G}) = \emptyset$, it follows that $\text{range}(\mathcal{G}) = \text{range}(\mathcal{C})$. So $\text{undec}(\mathcal{C}) = \emptyset$, so \mathcal{C} is a stable labelling. The grounded labelling is less or equally committed than every complete labelling (and is therefore also less or equally committed than every stable labelling). That is, $\mathcal{G} \sqsubseteq \mathcal{C}$. Since $\text{undec}(\mathcal{G}) = \emptyset$ it then follows that $\mathcal{G} = \mathcal{C}$. \square

Theorem 4.14: *Let $AF = (Ar, att)$ be an argumentation framework and \mathcal{G} be its grounded labelling. There exists at least one stage labelling \mathcal{S} such that $\mathcal{G} \sqsubseteq \mathcal{S}$.*

Proof: Let \mathcal{C}_{max} be an element of $\{\mathcal{C} \mid \mathcal{C} \text{ is a conflict-free labelling of } AF \text{ and } \mathcal{G} \sqsubseteq \mathcal{C}\}$ with a maximal range within this set. Since the set is finite (because AF is finite) and non-empty (because it contains at least \mathcal{G}) such an element with maximal range (\mathcal{C}_{max}) does exist. Now suppose that \mathcal{C}_{max} is not a stage labelling. Then there exists a conflict-free labelling \mathcal{D}_{max} of AF such that $\text{range}(\mathcal{C}_{max}) \subsetneq \text{range}(\mathcal{D}_{max})$. Let $AF' = AF|_{\text{range}(\mathcal{G})}$, $\mathcal{G}' = \mathcal{G}|_{\text{range}(\mathcal{G})}$, $\mathcal{C}'_{max} = \mathcal{C}_{max}|_{\text{range}(\mathcal{G})}$, and $\mathcal{D}'_{max} = \mathcal{D}_{max}|_{\text{range}(\mathcal{G})}$. From the fact that $\mathcal{G} \sqsubseteq \mathcal{C}_{max}$ it follows that $\mathcal{G}' = \mathcal{C}'_{max}$. Also, from $\text{range}(\mathcal{C}_{max}) \subsetneq \text{range}(\mathcal{D}_{max})$ it follows that $\text{range}(\mathcal{C}'_{max}) \subseteq \text{range}(\mathcal{D}'_{max})$. Furthermore, it holds that $\text{range}(\mathcal{G}') \subseteq \text{range}(\mathcal{C}'_{max})$ (since $\mathcal{G}' = \mathcal{C}'_{max}$), so by transitivity we obtain $\text{range}(\mathcal{G}') = \text{range}(\mathcal{D}'_{max})$. From Lemma 4.13 it then follows that $\mathcal{G}' = \mathcal{D}'_{max}$, from which it follows that $\mathcal{G} \sqsubseteq \mathcal{D}_{max}$. So \mathcal{D}_{max} is an element of $\{\mathcal{C} \mid \mathcal{C} \text{ is a conflict-free labelling of } AF \text{ and } \mathcal{G} \sqsubseteq \mathcal{C}\}$. But since $\text{range}(\mathcal{C}_{max}) \subsetneq \text{range}(\mathcal{D}_{max})$ it then follows that \mathcal{C}_{max} does not have a maximal range among the elements of this set. Contradiction. \square

Hence, a possible way of dealing with problems such as the one illustrated in Figure 2 is simply to delete all stage labellings that do not label all arguments without attackers in. Theorem 4.14 guarantees that after doing so, there will be at least one stage labelling left.

4.2. Verheij's Original Formalization of Stage Semantics

In (22), stage semantics was originally introduced not in terms of traditional argument extensions, nor in terms of labellings in the sense of Definition 2.4. Instead, it was introduced using what we will refer to as *argumentation tuples*, which in essence is a type of labelling in which undec is not explicitly represented. In the current section we treat the overlap between our formalization and Verheij's original work.

We first state Definition 3 of (22), although we have slightly changed the terminology in order to be compatible with what is nowadays commonly used in the field of formal argumentation.

Definition 4.15 Definition 3 of (22)

- (1) An argumentation tuple (IN, OUT) is a pair of disjoint sets of arguments. The union of IN and OUT is called the range of the argumentation tuple.
- (2) A stage tuple is an argumentation tuple (IN, OUT) such that for each argument $A \in IN \cup OUT$ it holds that:
 $A \in OUT$ iff there exists an argument $A' \in IN$ that attacks A
- (3) A stage tuple extension is a stage tuple (IN, OUT) such that there is no stage tuple with a larger range.

There exists a one-to-one relationship between the notion of stage tuples (Definition 4.15) and the notion of a conflict-free labellings (Definition 4.1).

Theorem 4.16: *Let (Ar, att) be an argumentation framework. It holds that (IN, OUT) is a stage tuple iff $(IN, OUT, Ar \setminus (IN \cup OUT))$ is a conflict-free labelling.*

Proof: “ \Rightarrow ”: Let (IN, OUT) be a stage tuple. We now prove the following.

- (1) “If $A \in IN$ then it does not have an attacker B such that $B \in IN$.”
 Let B be an attacker of A . Then according to point 2 of Definition 4.15, $B \in OUT$. Since IN and OUT are disjoint, it follows that $B \notin IN$.
- (2) “If $A \in OUT$ then it has at least one attacker B such that $B \in IN$.”
 This follows directly from point 2 of Definition 4.15.

“ \Leftarrow ”: Let $(IN, OUT, Ar \setminus (IN \cup OUT))$ be a conflict-free labelling. That is, for each argument $A \in Ar$ it holds that:

- (1) If $A \in IN$ then it does not have an attacker B such that $B \in IN$.
- (2) If $A \in OUT$ then it has at least one attacker B such that $B \in IN$.

We first observe that IN and OUT are disjoint (if $A \in IN$ and $A \in OUT$ then from 1 and 2 a contradiction follows) thus satisfying point 2 of Definition 4.15. Moreover, for each $A \in IN \cup OUT$ it holds that:

- if $A \in OUT$ then there exists an $A' \in IN$ that attacks A (follows directly from 2)
- if $A \notin OUT$ then there does not exist an $A' \in IN$ that attacks A (follows from 1, together with the fact that $A \in IN$ when $A \in IN \cup OUT$ and $A \notin OUT$)

From these two implications, point 2 of Definition 4.15 follows. \square

Similarly, there exists a one-to-one correspondence between the notion of stage tuple extensions (Definition 4.15) and the notion of stage labellings (Definition 4.2).

Theorem 4.17: *Let (Ar, att) be an argumentation framework. It holds that (IN, OUT) is a stage tuple extension iff $(IN, OUT, Ar \setminus (IN \cup OUT))$ is a stage labelling.*

Proof: “ \Rightarrow ”: Let (IN, OUT) be a stage extension. It then follows that $(IN, OUT, Ar \setminus (IN \cup OUT))$ is a conflict-free labelling (Theorem 4.16). The fact that $(IN, OUT, Ar \setminus (IN \cup OUT))$ is also a stage labelling (that is, a conflict-free labelling with minimal $Ar \setminus (IN \cup OUT)$) follows from the fact that $IN \cup OUT$ is maximal (from the fact that (IN, OUT) is a stage tuple extension, it follows that it has a maximal range).

“ \Leftarrow ”: Let $(IN, OUT, Ar \setminus (IN \cup OUT))$ be a stage labelling. It then follows that (IN, OUT) is a stage tuple (Theorem 4.16). The fact that (IN, OUT) is a stage tuple extension (that is, a stage tuple with maximal range $IN \cup OUT$) then follows from the fact that $Ar \setminus (IN \cup OUT)$ is minimal (since $(IN, OUT, Ar \setminus (IN \cup OUT))$ is

a stage labelling). □

Since stage extensions, stage labellings and stage tuple extensions express essentially the same concept (though in different manifestations) we can use either form without losing essential information.

Although Verheij shows that, given a stage tuple (IN, OUT) the set IN is an admissible set (and even a stable extension) of the argumentation framework restricted to $IN \cup OUT$ (Theorem 3 of (22)) he does not go all the way and state that a stage tuple extension is essentially about taking a maximal subframework that has at least one stable extension (labelling), as we have stated in Theorem 4.9 and Theorem 4.6.

The concept of semi-stable semantics corresponds with what Verheij calls “admissible stage extensions”.¹ It was ten years later that Caminada (initially unaware of this correspondence) reinvented the concept under the name of “semi-stable semantics”, and proved various of its properties, such as the fact that every semi-stable extension is also a complete extension, the fact that semi-stable semantics satisfies the postulate of relevance, and a number of conditions under which an argument is credulously or sceptically endorsed under semi-stable semantics.

5. Roundup

In the current work, we have extended the results of (4, 10) by showing that the labelling approach is applicable to two additional semantics from the existing argumentation literature: ideal semantics and stage semantics. Moreover, there turns out to be a one-to-one correspondence between the extensions and labellings of these two semantics.

Ideal semantics can be described in terms of judgment aggregation. In (11) it was shown that the ideal labelling is the result of the sceptical judgment aggregation procedure based on all preferred labellings, or alternatively, the result of the credulous or super-credulous judgment aggregation procedure on all admissible, complete or preferred labellings. In the current paper, we go one step further and show that this result is also the biggest (w.r.t. “ \sqsubseteq ”) admissible and complete labelling that is compatible with all admissible, complete or preferred labellings.

As for stage semantics, it was found that this semantics takes the stable labellings/extensions of the maximal subframework(s) that have at least one stable labelling/extension. Here, we again see that the properties of a semantics (in this case stage semantics) can be expressed both in terms of extensions and in terms of labellings.

Although the labellings approach and the extensions approach are equivalent (after all, an extension is basically the in-labelled part of a labelling) the labellings approach offers several advantages. First of all, the labellings approach can offer some technical advantages. For instance, the algorithm for stage semantics (8) has been stated in terms of labellings, and would be very difficult to rewrite using only the traditional extensions approach. A similar observation holds for the algorithm for semi-stable semantics (4, 16). What makes labellings suitable for constructing these algorithms is the fact that labellings can be defined in terms of relatively small and modular properties that hold independently from each other. The algorithms

¹Verheij also expresses the notions of preferred and stable semantics, as well as the notion of admissibility, in terms of argumentation tuples.

then uses the fine-grainedness of these properties to work to satisfy all of them in an incremental way.

The second and in our view even most important advantage of the labellings approach is that it tends to make formal argumentation more easy to understand, even for non-experts. For instance, for complete semantics (Proposition 2.8) the essential rule is that an argument has to be accepted iff all its attackers are rejected, and an argument has to be rejected iff it has at least one attacker that is accepted. Thus, argumentation can be explained without referring to things like admissibility or fixpoints of Dung's characteristic function. In a gunfight, one stays alive iff all attackers are dead, and one dies iff at least one attacker is still alive. Those who can understand this have basically understood what complete semantics is all about. This understanding can then also be used to explain the basic intuitions behind other semantics, like preferred, grounded, stable and semi-stable (see Table 1). As for ideal semantics, the labellings approach allows it to be described (Proposition Theorem 3.4) as the most committed reasonable (admissible/complete) position one can take that is still compatible with each reasonable (admissible/complete) position. As for stage semantics, the labellings approach also provides an intuitive description (Theorem 4.6): one wants to have a black-and-white (in or out) view of the world, in which there is no room for shades of gray (undec), even if one has to ignore part of the available information to achieve this (that is, one wants to have a stable labelling of a maximal part of the argumentation framework).

Overall, by treating two additional semantics (stage and ideal) in terms of labellings, we hope to have contributed to promoting labellings as a general way to characterize argumentation semantics, that has advantages of both technical and conceptual nature.

Thanks

We would like to thank Yining Wu for helping to develop the concept of an ideal labelling.

References

- [1] L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-00)*, pages 31–38, Boston, MA, 2000.
- [2] O. Arieli and A. Avron. The value of the four values. *Artificial Intelligence*, 102:97–141, 1998.
- [3] ASPIC-consortium. Deliverable D2.5: Draft formal semantics for ASPIC system, June 2005.
- [4] 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.
- [5] M.W.A. Caminada. On the issue of reinstatement in argumentation. Technical Report UU-CS-2006-023, Institute of Information and Computing Sciences, Utrecht University, 2006.
- [6] M.W.A. Caminada. Semi-stable semantics. In P.E. Dunne and T.J.M. Bench-Capon, editors, *Computational Models of Argument; Proceedings of COMMA 2006*, pages 121–130. IOS Press, 2006.
- [7] M.W.A. Caminada. An algorithm for computing semi-stable semantics. In *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (EC-SQARU 2007)*, number 4724 in Springer Lecture Notes in AI, pages 222–234, Berlin, 2007. Springer Verlag.
- [8] M.W.A. Caminada. A labelling approach for ideal and stage semantics. submitted, 2010.
- [9] M.W.A. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5-6):286–310, 2007.
- [10] M.W.A. Caminada and D.M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009. Special issue: new ideas in argumentation theory.
- [11] M.W.A. Caminada and G. Pigozzi. On judgment aggregation in abstract argumentation. *Journal of Autonomous Agents and Multi-Agent Systems*, 2009.
- [12] W. Carnielli, M.E. Coniglio, and J. Marcos. Logics of formal inconsistency. In D.M. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic, second edition*, volume 14, pages 15–114. Springer Verlag, 2002.

- [13] 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.
- [14] P. M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(10-15):642–674, 2007.
- [15] H. Jakobovits and D. Vermeir. Robust semantics for argumentation frameworks. *Journal of logic and computation*, 9(2):215–261, 1999.
- [16] S. Modgil and M.W.A. Caminada. Proof theories and algorithms for abstract argumentation frameworks. In I. Rahwan and G.R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 105–129. Springer, 2009.
- [17] J. L. Pollock. *Cognitive Carpentry. A Blueprint for How to Build a Person*. MIT Press, Cambridge, MA, 1995.
- [18] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7:25–75, 1997.
- [19] I. Rahwan and F. Tohmé. Collective argument evaluation as judgement aggregation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, 2010.
- [20] I. Rahwan, F. Zablith, and C. Reed. Laying the foundations for a world wide argument web. *Artificial Intelligence*, 171(10-15):897–921, 2007.
- [21] G.R. Simari and R.P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53:125–157, 1992.
- [22] B. Verheij. Two approaches to dialectical argumentation: admissible sets and argumentation stages. In J.-J.Ch. Meyer and L.C. van der Gaag, editors, *Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC'96)*, pages 357–368, Utrecht, 1996. Utrecht University.
- [23] B. Verheij. DEFLOG: on the logical interpretation of prima facie justified assumptions. *Journal of Logic and Computation*, 13:319–346, 2003.
- [24] G.A.W. Vreeswijk. Studies in defeasible argumentation. *PhD thesis at Free University of Amsterdam*, 1993.
- [25] G.A.W. Vreeswijk. An algorithm to compute minimally grounded and admissible defence sets in argument systems. In P.E. Dunne and T.J.M. Bench-Capon, editors, *Computational Models of Argument; Proceedings of COMMA 2006*, pages 109–120. IOS, 2006.