Martin Caminada
Dov M. Gabbay

A Logical Account of Formal Argumentation

Abstract. In the current paper, we re-examine how abstract argumentation can be formulated in terms of labellings, and how the resulting theory can be applied in the field of modal logic. In particular, we are able to express the (complete) extensions of an argumentation framework as models of a set of modal logic formulas that represents the argumentation framework. Using this approach, it becomes possible to define the grounded extension in terms of modal logic entailment.

Keywords: abstract argumentation, argument labellings, modal logic, grounded semantics.

1. Introduction

Formal argumentation has become a popular approach for purposes varying from nonmonotonic reasoning [2, 11], multi-agent communication [1] and reasoning in the semantic web [23]. Although some research on formal argumentation can be traced back to the early 1990s (like for instance the work of Vreeswijk [27] and of Simari and Loui [24]) the topic really started to take off with Dung's theory of abstract argumentation [12]. Here, arguments are seen as abstract entities (although they can be instantiated using approaches like [11] and [22]) 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 grounded, complete, preferred and stable semantics [12], as well as more recent approaches like semi-stable semantics [9] and ideal semantics [13].

Despite its relative popularity, formal argumentation has been criticized for its lack of meta-theory [6]. Although quite extensive work has been done describing the properties of the various argument-based semantics [4, 3], what is lacking is a comprehensive way of expressing properties of argumentation using existing logical approaches.

In this paper, we describe several alternative ways to express argumentation (and in particular argumentation semantics) other then the traditional

Special Issue: New Ideas in Argumentation Theory
Edited by **Dov M. Gabbay and Leendert van der Torre**

Studia Logica (2009) 93: 109–145
DOI: 10.1007/s11225-009-9218-x

extensions approach. In particular, we focus on argument labellings, classical logic and various forms of modal logic.

The remaining part of this paper is structured as follows. First, in Section 2, we treat some basic concepts of abstract argumentation, including the traditional extensions approach. Then, in Section 3, we describe the labellings approach and show how this can be used an alternative way to describe the standard admissibility based argumentation semantics. Section 4 then describes how argumentation can be expressed in modal logic. In section 5 argumentation is described using classical logic, and in Section 6 we introduce an alternative approach for using modal logic to describe argumentation. We then round off with a discussion of related work in Section 7.

2. Argumentation Preliminaries

An *argumentation framework* [12] 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 1. Let U be the universe of all possible arguments. An *argumentation framework* is a pair (Ar, att) where Ar is a finite subset of U and $att \subseteq Ar \times Ar$.

We say that an argument A *attacks* an argument B iff $(A, B) \in att$.

An argumentation framework can be depicted as a directed graph in which the arguments are represented as nodes and the attack relation is represented as arrows. For instance, argumentation framework (Ar, att) where $Ar = \{A, B, C, D, E\}$ and $att = \{(A, B), (B, A), (B, C), (C, D), (D, E), (E, C)\}$ is represented in Figure 1.

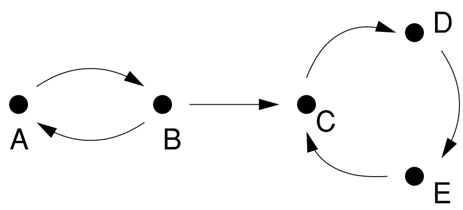

Figure 1. An argumentation framework represented as a directed graph.

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 $\mathcal{A}rgs$ is a set of arguments, then we write $\mathcal{A}rgs^+$

for the set of arguments that are attacked by at least one argument in $\mathcal{A}rgs$, and $\mathcal{A}rgs^-$ for the set of arguments that attack at least one argument in $\mathcal{A}rgs$. In the definition below, $F(\mathcal{A}rgs)$ stands for the set of arguments that are acceptable in the sense of [12].

DEFINITION 2 (defense / conflict-free). Let (Ar, att) be an argumentation framework, $A \in Ar$ and $\mathcal{A}rgs \subseteq Ar$.
We define A^+ as $\{B \mid A \ att \ B\}$ and $\mathcal{A}rgs^+$ as $\{B \mid A \ att \ B$ for some $A \in \mathcal{A}rgs\}$.
We define A^- as $\{B \mid B \ att \ A\}$ and $\mathcal{A}rgs^-$ as $\{B \mid B \ att \ A$ for some $A \in \mathcal{A}rgs\}$.
$\mathcal{A}rgs$ is *conflict-free* iff $\mathcal{A}rgs \cap \mathcal{A}rgs^+ = \emptyset$.
$\mathcal{A}rgs$ *defends* an argument A iff $A^- \subseteq \mathcal{A}rgs^+$.
We define the function $F : 2^{Ar} \rightarrow 2^{Ar}$ as
$F(\mathcal{A}rgs) = \{A \mid A$ is defended by $\mathcal{A}rgs\}$.

In the definition below, definitions of grounded, preferred and stable semantics are described in terms of complete semantics, which has the advantage of making the proofs in the remainder of this paper more straightforward. These descriptions are not literally the same as the ones provided by Dung [12], but as was first stated in [7], these are in fact equivalent to Dung's original versions of grounded, preferred and stable semantics.

DEFINITION 3 (acceptability semantics). Let (Ar, att) be an argumentation framework and let $\mathcal{A}rgs \subseteq Ar$ be a conflict-free set of arguments.

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

As an example, in the argumentation framework of Figure 1 $\{B, D\}$ is a stable extension, $\{A\}$ is a preferred extension which is neither stable nor semi-stable, $\emptyset$ is the grounded extension, and $\{B\}$ is an admissible set which is not a complete extension.

It is known that for every argumentation framework, there exists at least one admissible set (the empty set), exactly one grounded extension, one or more complete extensions, one or more preferred extensions and zero or more stable extensions. Moreover, when the set of arguments in the argumentation framework is finite, as is assumed in the current paper, there also exist one or more semi-stable extensions.

An overview of how the various extensions are related to each other is provided in Figure 2. The fact that every stable extension is also a semi-stable extension, and that every semi-stable extension is also a preferred extension was first stated in [9]. All other relations shown in Figure 2 have originally been stated in [12].

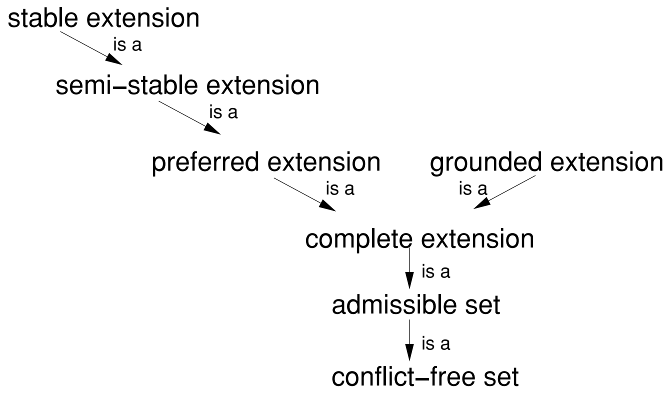

Figure 2. An overview of argumentation semantics (extension based).

3. Argument Labellings

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 [21] and Jakobovits and Vermeir [20], and has more recently been extended by Caminada [7, 10], Vreeswijk [28] and Verheij [26]. The idea of a labelling is to associate with each argument exactly one 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 is undecided, meaning that one abstains from an explicit judgment whether the argument is `in` or `out`.

Definition 4. Let (Ar, att) be an argumentation framework. A *labelling* is a total function $\mathcal{L} : Ar \longrightarrow \{\texttt{in}, \texttt{out}, \texttt{undec}\}$.

We write $\texttt{in}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \texttt{in}\}$, $\texttt{out}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \texttt{out}\}$ and $\texttt{undec}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \texttt{undec}\}$. Sometimes, we write a labelling $\mathcal{L}$ as a triple $(\mathcal{A}rgs_1, \mathcal{A}rgs_2, \mathcal{A}rgs_3)$ where $\mathcal{A}rgs_1 = \texttt{in}(\mathcal{L})$, $\mathcal{A}rgs_2 = \texttt{out}(\mathcal{L})$ and $\mathcal{A}rgs_3 = \texttt{undec}(\mathcal{L})$.

We distinguish three special kinds of labellings. The *all-in labelling* is a labelling that labels every argument `in`. The *all-out labelling* is a labelling that labels every argument `out`. The *all-undec labelling* is a labelling that labels every argument `undec`. We say that a labelling is *conflict-free* if no `in`-labelled argument attacks an (other or the same) `in`-labelled argument.

3.1. Complete Labellings

We will now define the concept of a complete labelling and show its relationship with Dung's concept of a complete extension.

Definition 5. 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 all attackers of A are labelled `out` then A is labelled `in`
3. if A is labelled `out` then A has a attacker that is labelled `in`, and
4. if A has a attacker that is labelled `in` then A is labelled `out`

Conditions 1 and 2 essentially form a bi-implication ("A is labelled `in` iff all its attacker are labelled `out`"), just like conditions 3 and 4 ("A is labelled `out` iff it has a attacker that is labelled `in`").

It is also possible to characterize a complete labelling as a labelling without arguments that are illegally `in`, illegally `out`, or illegally `undec`.

Definition 6. Let $\mathcal{L}$ be a labelling of argumentation framework (Ar, att) and let $A \in Ar$. We say that:

1. A is *illegally* `in` iff A is labelled `in` but not all its attackers are labelled `out`
2. A is *illegally* `out` iff A is labelled `out` but it does not have an attacker that is labelled `in`
3. A is *illegally* `undec` iff A is labelled `undec` but either all its attackers are labelled `out` or it has a attacker that is labelled `in`.

We say that an argument is *legally* `in` iff it is labelled `in` and is not illegally `in`. We say that an argument is *legally* `out` iff it is labelled `out` and is not illegally `out`. We say that an argument is *legally* `undec` iff it is labelled undec and is not illegally undec.

THEOREM 7. *Let $\mathcal{L}$ be a labelling of argumentation framework (Ar, att). It holds that $\mathcal{L}$ is a complete labelling iff*

- *every* `in`*-labelled argument is legally* `in`*,*
- *every* `out`*-labelled argument is legally* `out`*, and*
- *every* `undec`*-labelled argument is legally* `undec`*.*

PROOF. "$\Longrightarrow$": The fact that each `in`-labelled argument is legally `in` follows from point 1 of Definition 5. The fact that each `out`-labelled argument is legally `out` follows from point 3 of Definition 5. We will now prove that each `undec`-labelled argument is legally `undec`. Let A be an argument that is labelled `undec`. Then from point 2 of Definition 5 it follows that not all attackers of A are labelled `out` and from point 4 of Defintion 5 it follows that A does not have a attacker that is labelled `in`. Hence, A is legally `undec`.
"$\Longleftarrow$": Point 1 of Definition 5 follows from the fact that each `in`-labelled argument is legally `in`. Point 3 of Definition 5 follows from the fact that each `out`-labelled argument is legally `out`. Point 2 of Definition 5 can be proved as follows. Let A be an argument of which all attackers are labelled `out`. Then A cannot be labelled `out` (otherwise A would be illegally `out`) and A cannot be labelled `undec` (otherwise A would be illegally `undec`). Therefore, A can only be labelled `in`. Point 4 of Definition 5 can be proved as follows. Let A be an argument that has an attacker that is labelled `in`. Then A cannot be labelled `in` (otherwise A would be illegally `in`) and A cannot be labelled `undec` (otherwise A would be illegally `undec`). Hence, A can only be labelled `out`. ■

Using the results of Theorem 7 we can restate the concept of a complete labelling as follows.

PROPOSITION 1. *Let $\mathcal{L}$ be a labelling of argumentation framework (Ar, att). It holds that $\mathcal{L}$ is a complete labelling iff for each argument A it holds that:*

- *if A is labelled* `in` *then all its attackers are labelled* `out`*,*
- *if A is labelled* `out` *then it has at least one attacker that is labelled* `in`*, and*
- *if A is labelled* `undec` *then it has at least one attacker that is labelled* `undec` *and it does not have an attacker that is labelled* `in`*.*

LEMMA 1. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be complete labellings of argumentation framework $AF = (Ar, att)$. It holds that:*

- $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ *iff* $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$, *and*
- $\texttt{in}(\mathcal{L}_1) \subsetneq \texttt{in}(\mathcal{L}_2)$ *iff* $\texttt{out}(\mathcal{L}_1) \subsetneq \texttt{out}(\mathcal{L}_2)$

PROOF. We first prove that $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ iff $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$.
"$\Longrightarrow$": Suppose $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$. Let $A \in \texttt{out}(\mathcal{L}_1)$. From point 3 of Definition 5 it then follows that A has an attacker (say B) that is labelled in by $\mathcal{L}_1$. That is, $B \in \texttt{in}(\mathcal{L}_1)$. From the fact that $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ it then follows that $B \in \texttt{in}(\mathcal{L}_2)$. From point 4 of Definition 5 it then follows that A is labelled out by $\mathcal{L}_2$. That is, $A \in \texttt{out}(\mathcal{L}_2)$.
"$\Longleftarrow$": Suppose $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$. Let $A \in \texttt{in}(\mathcal{L}_1)$. From point 1 of Definition 5 it then follows that each attacker of A must be labelled out by $\mathcal{L}_1$. From the fact that $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$ it then follows that each attacker of A is also labelled out by $\mathcal{L}_2$. From point 2 of Definition 5 it then follows that A is labelled in by $\mathcal{L}_2$. That is, $A \in \texttt{in}(\mathcal{L}_2)$.
We now prove that $\texttt{in}(\mathcal{L}_1) \subsetneq \texttt{in}(\mathcal{L}_2)$ $\texttt{out}(\mathcal{L}_1) \subsetneq \texttt{out}(\mathcal{L}_2)$.
"$\Longrightarrow$": Suppose $\texttt{in}(\mathcal{L}_1) \subsetneq \texttt{in}(\mathcal{L}_2)$. This means that $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ and not $\texttt{in}(\mathcal{L}_2) \subseteq \texttt{in}(\mathcal{L}_1)$. It then follows that $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$ and not $\texttt{out}(\mathcal{L}_2) \subseteq \texttt{out}(\mathcal{L}_1)$. This means that $\texttt{out}(\mathcal{L}_1) \subsetneq \texttt{out}(\mathcal{L}_2)$.
"$\Longleftarrow$": Suppose $\texttt{out}(\mathcal{L}_1) \subsetneq \texttt{out}(\mathcal{L}_2)$. This means that $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$ and not $\texttt{out}(\mathcal{L}_2) \subseteq \texttt{out}(\mathcal{L}_1)$. It then follows that $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ and not $\texttt{in}(\mathcal{L}_2) \subseteq \texttt{in}(\mathcal{L}_1)$. This means that $\texttt{in}(\mathcal{L}_1) \subsetneq \texttt{in}(\mathcal{L}_2)$. ■

Lemma 1 implies that a complete labelling is uniquely defined by the in-labelled part, as well as by the out-labelled part.

LEMMA 2. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be complete labellings of argumentation framework (Ar, att).*

1. *if* $\texttt{in}(\mathcal{L}_1) = \texttt{in}(\mathcal{L}_2)$ *then* $\mathcal{L}_1 = \mathcal{L}_2$
2. *if* $\texttt{out}(\mathcal{L}_1) = \texttt{out}(\mathcal{L}_2)$ *then* $\mathcal{L}_1 = \mathcal{L}_2$

PROOF.

1. Suppose $\texttt{in}(\mathcal{L}_1) = \texttt{in}(\mathcal{L}_2)$. Then $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ and $\texttt{in}(\mathcal{L}_2) \subseteq \texttt{in}(\mathcal{L}_1)$, so from Lemma 1 it follows that $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$ and $\texttt{out}(\mathcal{L}_2) \subseteq \texttt{out}(\mathcal{L}_1)$, so $\texttt{out}(\mathcal{L}_1) = \texttt{out}(\mathcal{L}_2)$. From the fact that $\texttt{in}(\mathcal{L}_1) = \texttt{in}(\mathcal{L}_2)$ and $\texttt{out}(\mathcal{L}_1) = \texttt{out}(\mathcal{L}_2)$ it then follows that $\texttt{undec}(\mathcal{L}_1) = \texttt{undec}(\mathcal{L}_2)$ so $\mathcal{L}_1 = \mathcal{L}_2$.

2. Suppose $\mathtt{out}(\mathcal{L}_1) = \mathtt{out}(\mathcal{L}_2)$. Then $\mathtt{out}(\mathcal{L}_1) \subseteq \mathtt{out}(\mathcal{L}_2)$ and $\mathtt{out}(\mathcal{L}_2) \subseteq \mathtt{out}(\mathcal{L}_1)$, so from Lemma 1 it follows that $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$ and $\mathtt{in}(\mathcal{L}_2) \subseteq \mathtt{in}(\mathcal{L}_1)$, so $\mathtt{in}(\mathcal{L}_1) = \mathtt{in}(\mathcal{L}_2)$. From the fact that $\mathtt{in}(\mathcal{L}_1) = \mathtt{in}(\mathcal{L}_2)$ and $\mathtt{out}(\mathcal{L}_1) = \mathtt{out}(\mathcal{L}_2)$ it then follows that $\mathtt{undec}(\mathcal{L}_1) = \mathtt{undec}(\mathcal{L}_2)$ so $\mathcal{L}_1 = \mathcal{L}_2$. ■

It turns out that there is a one-to-one relationship between complete labellings and complete extensions, and that it is relatively straightforward to convert a complete labelling to a complete extension and vice versa.

DEFINITION 8. Let $AF = (Ar, att)$ be an argumentation framework, *clabellings* be its set of all conflict-free labellings and *csets* be its set of all conflict-free sets.
We define a function $\mathtt{Ext2Lab}_{AF}$: *csets* $\rightarrow$ *clabellings* such that $\mathtt{Ext2Lab}_{AF}$ $(\mathcal{A}rgs) = \{(A, \mathtt{in}) \mid A \in \mathcal{A}rgs\} \cup \{(A, \mathtt{out} \mid A \in \mathcal{A}rgs^{+}\} \cup \{(A, \mathtt{undec} \mid A \notin \mathcal{A}rgs$ and $A \notin \mathcal{A}rgs^{+}\}$.
We define a function $\mathtt{Lab2Ext}_{AF}$: *clabellings* $\rightarrow$ *csets* such that $\mathtt{Lab2Ext}_{AF}$ $(\mathcal{L}) = \mathtt{in}(\mathcal{L})$.

Sometimes, when the argumentation framework is either clear or not relevant, we write Ext2Lab and Lab2Ext instead of $\mathtt{Ext2Lab}_{AF}$ and $\mathtt{Lab2Ext}_{AF}$.

THEOREM 9. *Let $AF = (Ar, att)$ be an argumentation framework and let $\mathcal{L}$ be a complete labelling. Then $\mathtt{Lab2Ext}_{AF}(\mathcal{L})$ is a complete extension.*

PROOF. Let $\mathcal{A}rgs = \mathtt{Lab2Ext}_{AF}(\mathcal{L})$. We now prove that $\mathcal{A}rgs$ is a fixpoint of F.

$\mathcal{A}rgs \subseteq F(\mathcal{A}rgs)$**:** Let $A \in \mathcal{A}rgs$. Then $\mathcal{L}(A) = \mathtt{in}$. The fact that $\mathcal{L}$ is a complete labelling implies (point 1 of Definition 5) that each attacker B of A is labelled out. Point 3 of Definition 5 then implies that each such B has an attacker (say C) that is labelled in. From the definition of $\mathtt{Lab2Ext}_{AF}$ it then follows that $C \in \mathcal{A}rgs$. This means that for each attacker B of A, there is a $C \in \mathcal{A}rgs$ that attacks B. Therefore, $A \in F(\mathcal{A}rgs)$.

$F(\mathcal{A}rgs) \subseteq \mathcal{A}rgs$**:** Let $A \in F(\mathcal{A}rgs)$. Then each B that attacks A is attacked by some $C \in \mathcal{A}rgs$. From the definition of $\mathtt{Lab2Ext}_{AF}$ it follows that C is labelled in. The fact that $\mathcal{L}$ is a complete labelling then implies (point 4 of Definition 5) that each such B is labelled out, which then implies (point 2 of Definition 5) that A is labelled in. Therefore, by definition of $\mathtt{Lab2Ext}_{AF}$, it holds that $A \in \mathcal{A}rgs$.

The fact that $\mathcal{A}rgs$ is a fixpoint of F, together with the fact that $\mathcal{A}rgs$ is conflict-free (which follows from the fact that each complete labelling is also a conflict-free labelling) implies that $\mathcal{A}rgs$ is a complete extension. ■

THEOREM 10. *Let $AF = (Ar, att)$ be an argumentation framework and let $\mathcal{A}rgs$ be a complete extension. Then* $\texttt{Ext2Lab}_{AF}(\mathcal{A}rgs)$ *is a complete labelling.*

PROOF. We first observe that $\texttt{Ext2Lab}_{AF}(\mathcal{A}rgs)$ is well-defined because the fact that $\mathcal{A}rgs$ is a complete extension implies that it is conflict-free. We now prove the four properties of Definition 5.

1. *"if A is labelled* `in` *then all attackers of A are labelled* `out`*"*
   Let A be an argument that is labelled `in`. From the definition of $\texttt{Ext2Lab}_{AF}$ it then follows that $A \in \mathcal{A}rgs$. The fact that $\mathcal{A}rgs$ is a complete extension implies that it is an admissible set. That is, $\mathcal{A}rgs$ attacks every attacker of A. From the definition of $\texttt{Ext2Lab}_{AF}$ it then follows that every attacker of A is labelled `out`.
2. *"if all attackers of A are labelled* `out` *then A is labelled* `in`*"*
   Let A be an argument such that every attacker of A is labelled `out`. From the definition of $\texttt{Ext2Lab}_{AF}$ it then follows that every attacker of A is an element of $\mathcal{A}rgs^+$. This means that A is defended by $\mathcal{A}rgs$ ($A \in F(\mathcal{A}rgs)$). From the fact that $\mathcal{A}rgs$ is a complete extension it then follows that $A \in \mathcal{A}rgs$. From the definition of $\texttt{Ext2Lab}_{AF}$ it then follows that A is labelled `in`.
3. *"if A is labelled* `out` *then A has an attacker that is labelled* `in`*"*
   Let A be an argument that is labelled `out`. From the definition of $\texttt{Ext2Lab}_{AF}$ it then follows that $A \in \mathcal{A}rgs^+$, so there is an argument $B \in \mathcal{A}rgs$ that attacks A. From the definition of $\texttt{Ext2Lab}_{AF}$ it follows that B is labelled `in`.
4. *"if A has an attacker that is labelled* `in` *then A is labelled* `out`*"*
   Let A be an argument that has an attacker (say B) that is labelled `in`. From the definition of $\texttt{Ext2Lab}_{AF}$ it follows that $B \in \mathcal{A}rgs$, so $A \in \mathcal{A}rgs^+$. From the definition of $\texttt{Ext2Lab}_{AF}$ it then follows that A is labelled `out`. ■

When the domain and range of $\texttt{Lab2Ext}_{AF}$ are restricted to complete labellings and complete extensions, and the domain and range of $\texttt{Ext2Lab}_{AF}$ are restricted to complete extensions and complete labellings, then the resulting functions (call them $\texttt{Lab2Ext}^c_{AF}$ and $\texttt{Ext2Lab}^c_{AF}$) are bijective and each other's inverse.

THEOREM 11. *Let $AF = (Ar, att)$ be an argumentation framework, cextensions its set of complete extensions and clabellings its set of complete labellings. Let* $\texttt{Ext2Lab}^c_{AF}$ *: cextensions → clabellings be a function such that* $\texttt{Ext2Lab}^c_{AF}(\mathcal{A}rgs) = \texttt{Ext2Lab}_{AF}(\mathcal{A}rgs)$ *and* $\texttt{Lab2Ext}^c_{AF}$ *: clabellings → cextensions be a function such that* $\texttt{Lab2Ext}^c_{AF}(\mathcal{L}) = \texttt{Lab2Ext}_{AF}(\mathcal{L})$*. The functions* $\texttt{Ext2Lab}^c_{AF}$ *and* $\texttt{Lab2Ext}^c_{AF}$ *are bijective and each other's inverse.*

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

1. For every complete labelling $\mathcal{L}$ it holds that $\texttt{Ext2Lab}^c_{AF}(\texttt{Lab2Ext}^c_{AF}(\mathcal{L})) = \mathcal{L}$.
   Let $\mathcal{L}$ be a complete labelling of AF and let $A \in Ar$.
   If $\mathcal{L}(A) = \texttt{in}$ then $A \in \texttt{Lab2Ext}^c_{AF}(\mathcal{L})$, so $\texttt{Ext2Lab}^c_{AF}(\texttt{Lab2Ext}^c_{AF}(\mathcal{L}))(A) = \texttt{in}$.
   If $\mathcal{L}(A) = \texttt{out}$ then A is attacked by $\texttt{Lab2Ext}^c_{AF}(\mathcal{L})$, so $\texttt{Ext2Lab}^c_{AF}(\texttt{Lab2Ext}^c_{AF}(\mathcal{L}))(A) = \texttt{out}$.
   If $\mathcal{L}(A) = \texttt{undec}$ then $A \notin \texttt{Lab2Ext}^c_{AF}(\mathcal{L})$ and A is not attacked by $\texttt{Lab2Ext}^c_{AF}(\mathcal{L})$, so $\texttt{Ext2Lab}^c_{AF}(\texttt{Lab2Ext}^c_{AF}(\mathcal{L}))(A) = \texttt{undec}$.
2. For every complete extension $\mathcal{A}rgs$ it holds that $\texttt{Lab2Ext}^c_{AF}(\texttt{Ext2Lab}^c_{AF}(\mathcal{A}rgs)) = \mathcal{A}rgs$.
   Let $\mathcal{A}rgs$ be a complete extension of AF. We now prove two things:
   (a) $\texttt{Lab2Ext}^c_{AF}(\texttt{Ext2Lab}^c_{AF}(\mathcal{A}rgs)) \subseteq \mathcal{A}rgs$
   Let $A \in \texttt{Lab2Ext}^c_{AF}(\texttt{Ext2Lab}^c_{AF}(\mathcal{A}rgs))$. Then A is labelled in by $\texttt{Ext2Lab}^c_{AF}(\mathcal{A}rgs)$. Therefore $A \in \mathcal{A}rgs$.
   (b) $\mathcal{A}rgs \subseteq \texttt{Lab2Ext}^c_{AF}(\texttt{Ext2Lab}^c_{AF}(\mathcal{A}rgs))$
   Let $A \in \mathcal{A}rgs$. Then A is labelled in by $\texttt{Ext2Lab}^c_{AF}(\mathcal{A}rgs)$. Therefore $A \in \texttt{Lab2Ext}^c_{AF}(\texttt{Ext2Lab}^c_{AF}(\mathcal{A}rgs))$. ■

From Theorem 10 it follows that complete labellings and complete extensions stand in a one-to-one relationship to each other. In essence, complete labellings and complete extensions are different ways of describing the same thing.

Given the one-to-one relationship between complete extensions and complete labellings, the next step would be to examine what kind of labellings are associated with stable, grounded, preferred and semi-stable extensions, all of which are essentially special forms of complete extensions. What would be the properties of the labellings associated with these types of extensions? This question is studied in the following sections.

3.2. Stable Labellings

We start the discussion with examining the labellings that are associated with stable extensions. These labellings will be called *stable labellings*.

Definition 12. Let $AF = (Ar, att)$ be an argumentation framework. A stable labelling is a complete labelling $\mathcal{L}$ such that $\texttt{Lab2Ext}_{AF}(\mathcal{L})$ is a stable extension.

The fact that for complete labellings and complete extensions, `Lab2Ext` and `Ext2Lab` are inverse functions implies that if $\mathcal{A}rgs$ is a stable extension, then $\texttt{Ext2Lab}(\mathcal{A}rgs)$ is a stable labelling.

Stable labellings can also be characterized as complete labellings without `undec`.

Theorem 13. *Let $AF = (Ar, att)$ be an argumentation framework. The following statements are equivalent:*

1. *$\mathcal{L}$ is a complete labelling such that* $\texttt{undec}(\mathcal{L}) = \emptyset$
2. *$\mathcal{L}$ is a stable labelling*

Proof.

from 1 to 2: Suppose $\mathcal{L}$ is a complete labelling such that $\texttt{undec}(\mathcal{L}) = \emptyset$. Let $\mathcal{A}rgs$ be $\texttt{Lab2Ext}_{AF}(\mathcal{L})$. We now prove that $\mathcal{A}rgs$ is a stable extension (Definition 3). Let $A \in Ar \backslash \mathcal{A}rgs$. From the fact that $A \notin \mathcal{A}rgs$ it follows that A is not labelled `in` by $\mathcal{L}$. From the fact that $\texttt{undec}(\mathcal{L}) = \emptyset$ it follows that A is not labelled `undec` either. Therefore, A is labelled `out` by $\mathcal{L}$. From the fact that $\mathcal{L}$ is a complete labelling, it follows that A is attacked by an argument (say B) that is labelled `in`. From the fact that B is labelled `in` it follows that $B \in \mathcal{A}rgs$. This means that A is attacked by $\mathcal{A}rgs$.

from 2 to 1: Suppose $\mathcal{L}$ is a stable labelling, meaning that $\mathcal{A}rgs = \texttt{Lab2Ext}_{AF}(\mathcal{L})$ is a stable extension. We now prove that $\texttt{undec}(\mathcal{L}) = \emptyset$. Let $A \in Ar$. We distinguish two cases:

1. $A \in \mathcal{A}rgs$. Then A is labelled `in` by $\mathcal{L}$.
2. $A \notin \mathcal{A}rgs$. Then from the fact that $\mathcal{A}rgs$ is a stable extension, it follows that A is attacked by $\mathcal{A}rgs$, so A is labelled `out` by $\mathcal{L}$.

In both cases, $A \notin \texttt{undec}(\mathcal{L})$. Since this folds for an arbitrary $A \in Ar$, it follows that $\texttt{undec}(\mathcal{L}) = \emptyset$. ■

As an aside, one can also raise the question whether there exist labellings with empty in or empty out, and if so, what would be the meaning of these labellings. As for complete labellings with empty in it follows that these also have empty out, so each argument has to be labelled undec. Thus, a complete labelling with empty in is essentially the grounded labelling (see Definition 14 and Theorem 15) in an argumentation framework where each argument has at least one attacker.

As for complete labellings with empty out, it follows that each argument has to be labelled either in or undec. This means that no argument that is labelled in attacks any (other) argument. In essence, a complete labelling with empty out is the grounded labelling in an argumentation framework where each argument without attackers also does not attack any argument itself.

3.3. The Grounded Labelling

The next thing to be examined are the properties of the labelling associated with the grounded extension.

DEFINITION 14. Let $AF = (Ar, att)$ be an argumentation framework. A grounded labelling is a complete labelling $\mathcal{L}$ such that $\texttt{Lab2Ext}_{AF}(\mathcal{L})$ is the grounded extension.

The fact that the grounded extension is unique, and that for complete labellings and complete extensions, Lab2Ext and Ext2Lab are inverse functions, implies that the grounded labelling is unique, and that if $\mathcal{A}rgs$ is the grounded labelling, then $\texttt{Ext2Lab}(\mathcal{A}rgs)$ is the grounded extension.

The grounded labelling can be characterized as the complete labelling with minimal in, as the complete labelling with minimal out, or as the complete labelling maximal undec.

THEOREM 15. *Let $AF = (Ar, att)$ be an argumentation framework. The following statements are equivalent:*

1. *$\mathcal{L}$ is a complete labelling where $\texttt{in}(\mathcal{L})$ is minimal (w.r.t. set inclusion) among all complete labellings of AF*
2. *$\mathcal{L}$ is a complete labelling where $\texttt{out}(\mathcal{L})$ is minimal (w.r.t. set inclusion) among all complete labellings of AF*
3. *$\mathcal{L}$ is a complete labelling where $\texttt{undec}(\mathcal{L})$ is maximal (w.r.t. set inclusion) among all complete labellings of AF*
4. *$\mathcal{L}$ is the grounded labelling*

PROOF.

from 1 to 2: Let $\mathcal{L}$ be a complete labelling where $\texttt{out}(\mathcal{L})$ is not minimal. Then there exists a complete labelling $\mathcal{L}'$ with $\texttt{out}(\mathcal{L}') \subsetneq \texttt{out}(\mathcal{L})$. From Lemma 1 it then follows that $\texttt{in}(\mathcal{L}') \subsetneq \texttt{in}(\mathcal{L})$, so $\mathcal{L}$ is a labelling where $\texttt{in}(\mathcal{L})$ is not minimal.

from 2 to 1: Let $\mathcal{L}$ be a complete labelling where $\texttt{in}(\mathcal{L})$ is not minimal. Then there exists a complete labelling $\mathcal{L}'$ with $\texttt{in}(\mathcal{L}') \subsetneq \texttt{in}(\mathcal{L})$. From Lemma 1 it then follows that $\texttt{out}(\mathcal{L}') \subsetneq \texttt{out}(\mathcal{L})$, so $\mathcal{L}$ is a labelling where $\texttt{out}(\mathcal{L})$ is not minimal.

from 1 to 4: Let $\mathcal{L}$ be a complete labelling where $\texttt{in}(\mathcal{L})$ is minimal. Then $\texttt{Lab2Ext}_{AF}(\mathcal{L})$ is a minimal complete extension, which implies it is the grounded extension, so $\mathcal{L}$ is the grounded labelling.

from 4 to 1: Let $\mathcal{L}$ be the grounded labelling. Then $\texttt{Lab2Ext}_{AF}(\mathcal{L})$ is the grounded extension, which means it is the minimal complete extension. It then follows that $\texttt{in}(\mathcal{L})$ is minimal.

from 1 to 3: Let $\mathcal{L}$ be a complete labelling where $\texttt{in}(\mathcal{L})$ is minimal. Then $\texttt{Lab2Ext}_{AF}(\mathcal{L})$ is the grounded extension. Now suppose that $\texttt{undec}(\mathcal{L})$ is not maximal. Then there exists a complete labelling $\mathcal{L}'$ with $\texttt{undec}(\mathcal{L}) \subsetneq \texttt{undec}(\mathcal{L}')$. It holds that $\texttt{Lab2Ext}_{AF}(\mathcal{L}')$ is a complete extension, and from the fact that the grounded extension is a subset of each complete extension, it follows that $\texttt{Lab2Ext}_{AF}(\mathcal{L}) \subseteq \texttt{Lab2Ext}_{AF}(\mathcal{L}')$, so $\texttt{in}(\mathcal{L}) \subseteq \texttt{in}(\mathcal{L}')$. From Lemma 1 it then follows that $\texttt{out}(\mathcal{L}) \subseteq \texttt{out}(\mathcal{L}')$. From the fact that $\texttt{in}(\mathcal{L}) \subseteq \texttt{in}(\mathcal{L}')$ and $\texttt{out}(\mathcal{L}) \subseteq \texttt{out}(\mathcal{L}')$ it then follows that $\texttt{undec}(\mathcal{L}') \subseteq \texttt{undec}(\mathcal{L})$. Contradiction.

from 3 to 1: Let $\mathcal{L}$ be a complete labelling where $\texttt{in}(\mathcal{L})$ is not minimal. Then there exists a complete labelling $\mathcal{L}'$ with $\texttt{in}(\mathcal{L}') \subsetneq \texttt{in}(\mathcal{L})$. It then also follows (Lemma 1) that $\texttt{out}(\mathcal{L}') \subsetneq \texttt{out}(\mathcal{L})$, so $\texttt{undec}(\mathcal{L}) \subsetneq \texttt{undec}(\mathcal{L}')$. Contradiction. ■

3.4. Preferred Labellings

We now examine the properties of the labellings associated with preferred extensions.

DEFINITION 16. Let $AF = (Ar, att)$ be an argumentation framework. A preferred labelling is a complete labelling $\mathcal{L}$ such that $\texttt{Lab2Ext}_{AF}(\mathcal{L})$ is a preferred extension.

The fact that for complete labellings and complete extensions, $\texttt{Lab2Ext}$ and $\texttt{Ext2Lab}$ are inverse functions implies that if $\mathcal{A}rgs$ is a preferred extension, then $\texttt{Ext2Lab}(\mathcal{A}rgs)$ is a preferred labelling.

Preferred labellings can be characterized as complete labellings with maximal $\texttt{in}$, or as complete labellings with maximal $\texttt{out}$.

THEOREM 17. *Let $AF = (Ar, att)$ be an argumentation framework. The following statements are equivalent:*

1. *$\mathcal{L}$ is a complete labelling where $\texttt{in}(\mathcal{L})$ is maximal (w.r.t. set inclusion) among all complete labellings of AF*
2. *$\mathcal{L}$ is a complete labelling where $\texttt{out}(\mathcal{L})$ is maximal (w.r.t. set inclusion) among all complete labellings of AF*
3. *$\mathcal{L}$ is a preferred labelling*

PROOF.

from 1 to 2: Let $\mathcal{L}$ be a complete labelling where $\texttt{out}(\mathcal{L})$ is not maximal. Then there exists a complete labelling $\mathcal{L}'$ with $\texttt{out}(\mathcal{L}) \subsetneq \texttt{out}(\mathcal{L}')$. From Lemma 1 it then follows that $\texttt{in}(\mathcal{L}) \subsetneq \texttt{in}(\mathcal{L}')$, so $\mathcal{L}$ is a labelling where $\texttt{in}(\mathcal{L})$ is not maximal.

from 2 to 1: Let $\mathcal{L}$ be a complete labelling where $\texttt{in}(\mathcal{L})$ is not maximal. Then there exists a complete labelling $\mathcal{L}'$ with $\texttt{in}(\mathcal{L}) \subsetneq \texttt{in}(\mathcal{L}')$. From Lemma 1 it then follows that $\texttt{out}(\mathcal{L}) \subsetneq \texttt{out}(\mathcal{L}')$, so $\mathcal{L}$ is a labelling where $\texttt{out}(\mathcal{L})$ is not maximal.

from 1 to 3: Let $\mathcal{L}$ be a complete labelling where $\texttt{in}(\mathcal{L})$ is maximal. Then $\texttt{Lab2Ext}_{AF}(\mathcal{L})$ is a maximal complete extension, which implies it is a preferred extension, so $\mathcal{L}$ is a preferred labelling.

from 3 to 1: Let $\mathcal{L}$ be a preferred labelling. Then $\texttt{Lab2Ext}_{AF}(\mathcal{L})$ is a preferred extension, which means it is a maximal complete extension. It then follows that $\texttt{in}(\mathcal{L})$ is maximal. ■

3.5. Semi-Stable Labellings

We now examine the properties of the labellings associated with semi-stable extensions.

DEFINITION 18. Let $AF = (Ar, att)$ be an argumentation framework. A semi-stable labelling is a complete labelling $\mathcal{L}$ such that $\texttt{Lab2Ext}_{AF}(\mathcal{L})$ is a semi-stable extension.

The fact that for complete labellings and complete extensions, `Lab2Ext` and `Ext2Lab` are inverse functions implies that if $\mathcal{A}rgs$ is a semi-stable extension, then $\texttt{Ext2Lab}(\mathcal{A}rgs)$ is a semi-stable labelling.

Semi-stable labellings can be characterized as complete labellings with minimal `undec`.

THEOREM 19. *Let* $AF = (Ar, att)$ *be an argumentation framework. The following statements are equivalent:*

1. $\mathcal{L}$ *is a complete labelling where* $\texttt{undec}(\mathcal{L})$ *is minimal (w.r.t. set inclusion) among all complete labellings of* AF
2. $\mathcal{L}$ *is a semi-stable labelling*

PROOF.

from 1 to 2: Suppose $\mathcal{L}$ is a complete labelling, but not a semi-stable labelling. Then $\texttt{Lab2Ext}_{AF}(\mathcal{L})$ is a complete extension but not a semi-stable extension. This implies that $\texttt{Lab2Ext}_{AF}(\mathcal{L}) \cup \texttt{Lab2Ext}_{AF}(\mathcal{L})^+$ is not maximal. So there exists a complete extension $\mathcal{A}rgs'$ such that $\texttt{Lab2Ext}_{AF}(\mathcal{L}) \cup \texttt{Lab2Ext}_{AF}(\mathcal{L})^+ \subsetneq \mathcal{A}rgs' \cup \mathcal{A}rgs'^+$. Let $\mathcal{L}'$ be the labelling associated with $\mathcal{A}rgs'$ (that is: $\mathcal{L}' = \texttt{Ext2Lab}_{AF}(\mathcal{A}rgs')$ and $\mathcal{A}rgs'$ $= \texttt{Lab2Ext}_{AF}(\mathcal{L}')$). It then follows that $\texttt{Lab2Ext}_{AF}(\mathcal{L}) \cup \texttt{Lab2Ext}_{AF}(\mathcal{L})^+$ $\subsetneq \texttt{Lab2Ext}_{AF}(\mathcal{L}') \cup \texttt{Lab2Ext}_{AF}(\mathcal{L}')^+$, so $\texttt{in}(\mathcal{L}) \cup \texttt{out}(\mathcal{L}) \subsetneq \texttt{in}(\mathcal{L}') \cup \texttt{out}(\mathcal{L}')$, so $\texttt{undec}(\mathcal{L}') \subsetneq \texttt{undec}(\mathcal{L})$. This means that $\mathcal{L}$ is not a labelling where $\texttt{undec}(\mathcal{L})$ is minimal.

from 2 to 1: Suppose that $\mathcal{L}$ is a complete labelling where $\texttt{undec}(\mathcal{L})$ is not minimal. This implies that there exists a complete labelling $\mathcal{L}'$ such that $\texttt{undec}(\mathcal{L}') \subsetneq \texttt{undec}(\mathcal{L})$. It then follows that $\texttt{in}(\mathcal{L}) \cup \texttt{out}(\mathcal{L}) \subsetneq \texttt{in}(\mathcal{L}') \cup \texttt{out}(\mathcal{L}')$. Let $\mathcal{A}rgs = \texttt{Lab2Ext}_{AF}(\mathcal{L})$ and $\mathcal{A}rgs' = \texttt{Lab2Ext}_{AF}(\mathcal{L}')$. It then follows that $\mathcal{A}rgs \cup \mathcal{A}rgs^+ \subsetneq \mathcal{A}rgs' \cup \mathcal{A}rgs'^+$, which means that $\mathcal{A}rgs$ is not a semi-stable extension, which implies that $\mathcal{L}$ is not a semi-stable labelling. ■

3.6. Roundup

The main results of the discussion until so far are summarized in Table 1. Notice that we have covered all combinations of minimal or maximal `in`, `out` or `undec`. Almost all combinations turn out to correspond to the traditional Dung-style semantics. The only exception are the labellings with minimal `undec`, which corresponds with a semantics not introduced in [12] but in [9].

restriction complete labellings	Dung-style semantics	linked by def. and th.
no restrictions	complete semantics	Def. 5 and Th. 7
empty `undec`	stable semantics	Def. 12 and Th. 13
maximal `in`	preferred semantics	Def. 16 and Th. 17
maximal `out`	preferred semantics	Def. 16 and Th. 17
maximal `undec`	grounded semantics	Def. 14 and Th. 15
minimal `in`	grounded semantics	Def. 14 and Th. 15
minimal `out`	grounded semantics	Def. 14 and Th. 15
minimal `undec`	semi-stable semantics	Def. 18 and Th. 19

Table 1. Argument labellings and Dung-style semantics.

Overall, one can see labellings as an alternative way to specify argumentation semantics. 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 abstract argumentation is all about.

3.7. Admissible Labellings

Until so far, we have modelled the concepts of complete, stable, grounded, preferred and semi-stable semantics in terms of labellings. This was done by *strengthening* the concept of complete labellings. Another route would be to *weaken* the concept of a complete labelling. An obvious way to do this would be to take only a subset of the four properties of Definition 5. However, these four properties are not completely independent. If one requires property 1 ("if A is labelled `in` then all attackers of A are labelled `out`") then it makes sense also to require property 3 ("if A is labelled `out` then A has an attacker that is labelled `in`"), and vice versa. If one requires each `in` label to make sense, and defines this in terms of `out` labels, then one should also require each `out` label to make sense (and vice versa). Together, properties 1 and 3 stand for admissibility.

Definition 20. 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, but the relationship is no longer one-to-one.

THEOREM 21. *Let $AF = (Ar, att)$ be an argumentation framework. It holds that:*

1. *if $\mathcal{A}rgs$ is an admissible set of AF then $\texttt{Ext2Lab}_{AF}(\mathcal{A}rgs)$ is an admissible labelling of AF, and*
2. *if $\mathcal{L}$ is an admissible labelling of AF then $\texttt{Lab2Ext}_{AF}(\mathcal{L})$ is an admissible set of AF.*

PROOF.

1. Let $\mathcal{A}rgs$ be an admisssible set of AF, and let $\mathcal{L} = \texttt{Ext2Lab}_{AF}(\mathcal{A}rgs)$. We now prove that $\mathcal{L}$ is an admissible labelling.
   (a) Let A be an argument that is labelled `in` by $\mathcal{L}$. Let B be an arbitrary attacker of A. The fact that A is labelled `in` by $\mathcal{L}$ implies that $A \in \mathcal{A}rgs$. The fact that $\mathcal{A}rgs$ is an admissible set implies that $\mathcal{A}rgs$ attacks B. That is, $B \in \mathcal{A}rgs^+$, so B is labelled `out` by $\mathcal{L}$. Since this holds for any attacker B of A it follows that every attacker of A is labelled `out` by $\mathcal{L}$
   (b) Let A be an argument that is labelled `out` by $\mathcal{L}$. It then follows that $A \in \mathcal{A}rgs^+$, so there exists a $B \in \mathcal{A}rgs$ that attacks A. This B is labelled `in` by $\mathcal{L}$. This means that A has a attacker that is labelled `in` by $\mathcal{L}$.
2. Let $\mathcal{L}$ be an admissible labelling of AF, and let $\mathcal{A}rgs = \texttt{Lab2Ext}_{AF}(\mathcal{L})$. We now prove that $\mathcal{A}rgs$ is an admissible set.
   (a) We first prove that $\mathcal{A}rgs$ is conflict-free. Suppose this is not the case. Then there exist $A, B \in \mathcal{A}rgs$ such that A attacks B. It then follows that both A and B are labelled `in` by $\mathcal{L}$. But this cannot be the case because the fact that $\mathcal{L}$ is an admissible labelling implies that all attackers of A (including B) are labelled `out` by $\mathcal{L}$.
   (b) We now prove that $\mathcal{A}rgs$ defends all its elements. Let $A \in \mathcal{A}rgs$. Then A is labelled `in` by $\mathcal{L}$. Let B be an arbitrary argument that attacks A. From the fact that $\mathcal{L}$ is an admissible labelling, it follows that B is labelled `out` by $\mathcal{L}$. From the fact that $\mathcal{L}$ is an admissible labelling it then also follows that B has an attacker C that is labelled `in` by $\mathcal{L}$, so $C \in \mathcal{A}rgs$. Therefore $\mathcal{A}rgs$ is self-defending. ■

Admissible labellings and admissible sets have a many-to-one relationship. That is, each admissible labelling is associated with exactly one admissible set, but each admissible set is associated with one or more admissible labellings. As an example, consider again the argumentation framework of Figure 1. Here, $(\{B\}, \{A, C\}, \{D, E\})$ and $(\{B\}, \{A\}, \{C, D, E\})$ are two admissible labellings associated with the same admissible set $\{B\}$.

For admissible labellings (say $\mathcal{L}_1$ and $\mathcal{L}_2$) it does *not* hold that if $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ then $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$. As a counter example, consider again the argumentation framework of Figure 1, with $\mathcal{L}_1 = (\{B\}, \{A, C\}, \{D, E\})$ and $\mathcal{L}_2 = (\{B\}, \{A\}, \{C, D, E\})$. Similarly, it also does *not* hold that if $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$ then $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$. As a counter example, consider the argumentation framework $AF = (\{A, B, C, D\}, \{(A, C), (B, C), (C, D)\})$ with $\mathcal{L}_1 = (\{A, B, D\}, \{C\}, \emptyset)$ and $\mathcal{L}_2 = (\{A, D\}, \{C\}, \{B\})$.

Between the concept of an admissible labelling and a complete labelling, one can distinguish two intermediate forms.

Definition 22. Let $AF = (Ar, att)$ be an argumentation framework.
A *JV-labelling* is a labelling that satisfies:

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 has an attacker that is labelled `in` then A is labelled `out`

A *VJ-labelling* is a labelling that satisfies:

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 all attackers of A are labelled `out` then A is labelled `in`

In essence, an admissible labelling satisfies point 1 and 3 of a complete labelling (Definition 5), a JV-labelling satisfies point 1, 3 and 4, and a VJ-labelling satisfies point 1, 3 and 2. It immediately follows that every complete labelling is also a JV-labelling and a VJ-labelling, and that every JV-labelling or VJ-labelling is also an admissible labelling. We use the term JV-labelling, because these are quite close to a proposal of Jakobovits and Vermeir [20]. An important difference, however, is that in our approach each argument gets exactly one out of three possible labels (`in`, `out` or `undec`) whereas in Jakobovits and Vermeir's original proposal, there are four possibilities (either no label, single `in`, single `out`, or both `in` and `out`).[1]

---

[1]Another difference is that Jakobovits and Vermeir use the symbol "+" instead of `in` and the symbol "−" instead of `out`.

It turns out that JV-labellings are uniquely identified by their `in` labelled part, whereas VJ-labellings are uniquely identified by their `out` labelled part.

LEMMA 3. *Let $AF = (Ar, att)$ be an argumentation framework. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be JV-labellings of AF and let $\mathcal{L}_3$ and $\mathcal{L}_4$ be VJ-labellings of AF. It holds that:*

1. *if* $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$ *then* $\mathtt{out}(\mathcal{L}_1) \subseteq \mathtt{out}(\mathcal{L}_2)$
2. *if* $\mathtt{in}(\mathcal{L}_1) \subsetneq \mathtt{in}(\mathcal{L}_2)$ *then* $\mathtt{out}(\mathcal{L}_1) \subsetneq \mathtt{out}(\mathcal{L}_2)$
3. *if* $\mathtt{out}(\mathcal{L}_3) \subseteq \mathtt{out}(\mathcal{L}_4)$ *then* $\mathtt{in}(\mathcal{L}_3) \subseteq \mathtt{in}(\mathcal{L}_4)$
4. *if* $\mathtt{out}(\mathcal{L}_3) \subsetneq \mathtt{out}(\mathcal{L}_4)$ *then* $\mathtt{in}(\mathcal{L}_3) \subsetneq \mathtt{in}(\mathcal{L}_4)$

PROOF. Similar to the proof of Lemma 1. ■

LEMMA 4. *Let $AF = (Ar, att)$ be an argumentation framework. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be JV-labellings of AF and let $\mathcal{L}_3$ and $\mathcal{L}_4$ be VJ-labellings of AF. It holds that:*

1. *if* $\mathtt{in}(\mathcal{L}_1) = \mathtt{in}(\mathcal{L}_2)$ *then* $\mathcal{L}_1 = \mathcal{L}_2$
2. *if* $\mathtt{out}(\mathcal{L}_3) = \mathtt{out}(\mathcal{L}_4)$ *then* $\mathcal{L}_3 = \mathcal{L}_4$

PROOF. Similar to the proof of Lemma 2. ■

Since JV-labellings are uniquely identified by their `in` labelled part, and the fact that an admissible set essentially specifies a set of `in` labelled arguments, it does not come as a surprise that there exists a one-to-one relation between admissible sets and JV-labellings.

THEOREM 23. *Let $AF = (Ar, att)$ be an argumentation framework, asets be its set of admissible sets and JV-labellings be its set of JV-labellings. Let $\mathtt{Ext2Lab}_{AF}^{JV}$: asets $\rightarrow$ JV-labellings be a function such that $\mathtt{Ext2Lab}_{AF}^{JV}(\mathcal{A}rgs) = \mathtt{Ext2Lab}_{AF}(\mathcal{A}rgs)$ and $\mathtt{Lab2Ext}_{AF}^{JV}$: JV-labellings $\rightarrow$ asets be a function such that $\mathtt{Lab2Ext}_{AF}^{JV}(\mathcal{L}) = \mathtt{Lab2Ext}_{AF}(\mathcal{L})$. The functions $\mathtt{Ext2Lab}_{AF}^{JV}$ and $\mathtt{Lab2Ext}_{AF}^{JV}$ are bijective and each other's inverse.*

PROOF. Similar to the proof of Theorem 11. ■

A global overview of the relations between the various forms of labellings is provide in Figure 3.

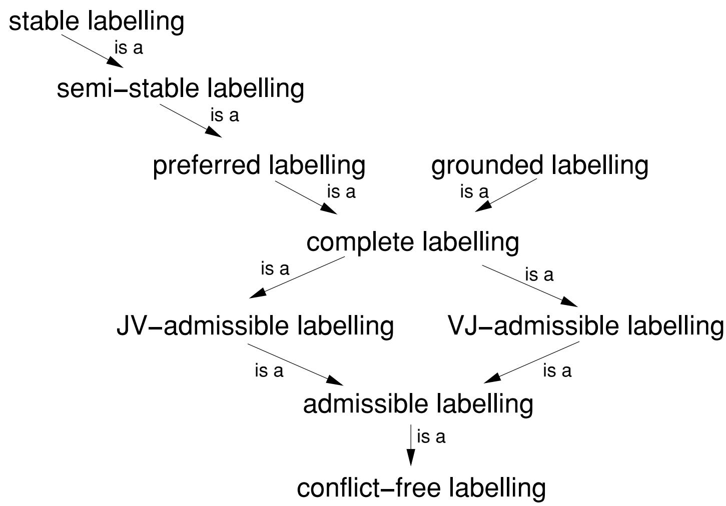

Figure 3. An overview of argumentation semantics (labelling based).

4. Argumentation and Modal Logic

There are three main methods of introducing modal logic into argumentation theory: the metalevel approach, the object level approach and the mixed approach. We view an argumentation system as a combination of an argumentation framework $AF = (Ar, att)$ (where Ar is a set of atomic arguments and $att \subseteq Ar \times Ar$ is the attack relation) and a complete labelling $\mathcal{L}$ of AF. To us, the thus described argumentation system serves as the object level.

The metalevel approach talks about the argumentation framework from "above", using another language and logic. The metalevel language can be classical logic or modal logic. These are traditional metalevel languages and are used in this way in many areas. Classical logic can be full classical logic or the computational Horn clause logic programming language. When classical logic is used, one can think of the process as translation. There is a traditional translation of modal logic into classical logic and through this translation the metalevel approaches are related.

For the sake of clarity, we shall present all three metalevel versions in the appropriate sections: the classical logic one, the Horn clause logic programming one, and the modal logic one. They are related but are not the same; each one has its advantages and limitations.

4.1. Modal Logic Preliminaries

This subsection introduces some modal logic background needed for introducing our approaches.

The modal logic $\mathcal{K}$ is a propositional system with the modal operator $\Box$ and the atomic propositions $Q = \{q_1, q_2, q_3 \dots\}$ and $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\top$ and $\bot$. Models for $\mathcal{K}$ have the form $\mathcal{M} = (S, R, h)$ where S is the set of possible worlds, $R \subseteq S \times S$, $S \neq \emptyset$ and h is the assignment function, giving to each atomic letter q a subset $h(q) \subseteq S$. Satisfaction is defined as follows for $t \in S$:

- $t \vDash q$ iff $t \in h(q)$, for atomic q
- $t \vDash A \wedge B,\ A \vee B,\ \neg A,\ A \rightarrow B$ are defined as usual
- $t \vDash \Box A$ iff for all s such that tRs we have $s \vDash A$
- A holds in (S, R, h) iff for all $t \in S$ we have $t \vDash A$.

$\mathcal{K}$ can be axiomatised as follows:

1. all substitution instances of truth functional tautologies
2. $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$
3. $\dfrac{\vdash A}{\vdash \Box A}$

It is complete for the class of all finite frames of the form (S, R) where $S \neq \emptyset$, S is finite, and $R \subseteq S \times S$.

4.2. Products of Modal Logics

We need the concept of an H-product of a fixed frame modal logic. To define this, we need to introduce the concept of a fixed frame modal logic as well as the concept of a product.

Definition 24. A modal logic $\mathbb{L}$ is said to be a fixed frame modal logic (FF modal logic) if it can be characterised by a family of models of the form (S_0, R_0, h), $h \in H$, where (S_0, R_0) is a fixed frame and H is a set of assignments to this frame. We have $\mathbb{L} \vdash A$ iff A holds in all FF-models (S_0, R_0, h), $h \in H$.

We can now define the cross product of the two modal logics. Let $\Box_1$ be a modality of modal logic $\mathbb{E}_1$ that is completely chacterised by a class of models of the form $\mathcal{M}_1 = \{(S^1_i, R^1_i, h^1_i)\}$, $i \in I_1$ and similarly for $\Box_2$, $\mathbb{E}_2$, $\mathcal{M}_2 = \{(S^2_j, R^2_j, h^2_j)\}$, $j \in I_2$. We define the flat-cross product of these two logics semantically, through their models. The language contains

the two modalities $\{\Box_1, \Box_2\}$. The models have the form $(S^1 \times S^2, R_1 \cup R_2, h_{1,2})$ where (S^1, R^1, h^1) and (S^2, R^2, h^2) are any models of $\Box_1$ and $\Box_2$ respectively and $S^1 \times S^2$ is the product of the sets S^1 and S^2. We define R_1 and R_2 as follows.
$(x_1, x_2)R_1(y_1, x_2)$ iff $x_1 R^1 y_1$
$(x_1, x_2)R_2(x_1, y_2)$ iff $x_2 R^2 y_2$
We define $h_{1,2}$ by some boolean function of h_1 and h_2, for example
$(x_1, x_2) \in h_{1,2}(q)$ iff $x_1 \in h_1(q)$ or $x_2 \in h_2(q)$
or another definition
$(x_1, x_2) \in h_{1,2}(q)$ iff $x_1 \in h_1(q)$.
We have
$(x_1, x_2) \vDash \Box_1 A$ iff for all y_1, $x_1 R^1 y_1$ we have $(y_1, x_2) \vDash A$.
$(x_1, x_2) \vDash \Box_2 A$ iff for all y_2, $x_2 R^2 y_2$ we have $(x_1, y_2) \vDash A$.
The above definition defines a general flat product.

When we have a single FF-logic, with $\Box$ characterised by a fixed frame (S_0, R_0) and a family of assignments H, we can form the universal product of the logic with $\Box$ along the axis of H. We need a new modality for the H axis, which we denote by $\boxdot$.

Definition 25. Let (S_0, R_0) be a frame for a fixed frame modal logic $\mathbb{L}$ with $\Box$. Let H be a family of assignments $h \in H$ such that (S_0, R_0, h) is a model of $\mathbb{L}$. We form the universal H product of the frame as follows. We form the set $\mathcal{M} = \{(S_0, R_0, h) \mid h \in H\}$. We use a modality $\boxdot$ to move around $\mathcal{M}$. We then have two modalities: $\Box$ and $\boxdot$. Satisfaction in $\mathcal{M}$ is defined as follows.
$(t, h) \vDash q$ iff $t \in h(q)$, for $t \in S$, $h \in H$
$(t, h) \vDash \Box A$ iff for all s, tRs implies $(s, h) \vDash A$
$(t, h) \vDash \boxdot A$ iff for all $h' \neq h$, $(t, h') \vDash A$
A holds in the model iff for all t, h, $(t, h) \vDash A$.

Figure 4 shows this is indeed a product.

See the book [17] for a wealth of material products.

4.3. Modal Provability Logic

Löb's logic for one modality $\Box$ has the following axioms and rules.

1. axioms and rules of modal logic K
2. $\Diamond A \to \Diamond(A \wedge \Box\neg A)$
3. $\Box A \to \Box\Box A$

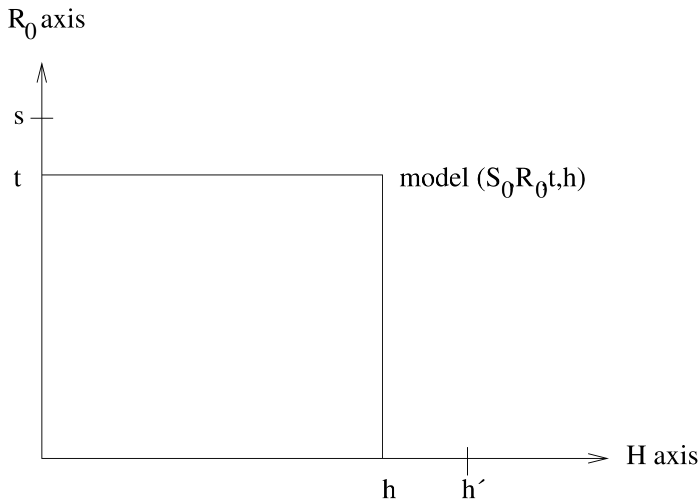

Figure 4. $\Box$ moves from t to s on the h vertical.$\boxdot$ moves from h to h' on the t horizontal.

Axiom 2 says that if A is possible then it is possible for the last time. Axiom 3 says R is transitive.

It is complete for the class of all frames which are finite and acyclic (we can take all finite tree frames). The following holds.

THEOREM 26 (fixed point theorem). *Let $\Psi(x)$ be a modal formula with the propositional variable x such that x is in the scope of a modality $\Box$. Then there is a formula ϕ which is a solution to the fixed point equation $x \leftrightarrow \Psi(x)$. Namely, we have $\vdash \phi \leftrightarrow \Psi(x/\phi)$.*

PROOF. See Theorem 3.4 (page 464) of [25]. ■

We shall use an extension of this logic in our object level modal approach. The logic we use we call $LN1$; it is characterised by linear chains. The axiom we add for it is:

4. $\Diamond A \wedge \Diamond B \to \Diamond(A \wedge B) \vee \Diamond(A \wedge \Diamond B) \vee \Diamond(B \wedge \Diamond A)$

If we then also add the following axiom

5. $\Box\Box\Box\bot$

we get the logic of chains of length 3 (3-chains) of the form of Figure 5.

The next step will be to connect these chains to labellings.

Figure 5. A 3-chain.

4.4. The Metalevel Approach

We use the modal logic $\mathcal{K}$ to describe the system of Definition 4. We can view (Ar, att) as a modal frame and view $\mathcal{L}$ as an assignment to three propositional atomic letters: q_1, q_0 and $q_?$, which we regard as constants. We have
$a \vDash q_1$ iff $\mathcal{L}(a) = \texttt{in}$,
$a \vDash q_0$ iff $\mathcal{L}(a) = \texttt{out}$, and
$a \vDash q_?$ iff $\mathcal{L}(a) = \texttt{undec}$.
The modality goes by the direction of "being attacked from", namely
$a \vDash \Box A$ iff for all y such that y *att* a (that is, $(y, a) \in att$), we have $y \vDash A$.
We therefore must adopt the following axioms, written as $\Delta(q_1, q_0, q_?)$.

1. $\Box\bot \vee \Box q_0 \rightarrow q_1$ (if a is not attacked by any argument, i.e. $\Box\bot$ holds, or all attackets of a are `out` then a is `in`)
2. $\Diamond q_1 \rightarrow q_0$ (if a is attacked by an argument which is `in` then a is `out`)
3. $\Box(q_0 \vee q_?) \wedge \Diamond q_? \rightarrow q_?$ (if all the attackers of a are either `out` or `undec` and at least one attacker of a is `undec` then a is `undec`)
4. $\vdash^m (q_0 \vee q_1 \vee q_?)$, $m \geq 0$ (each argument has at least one label `out`, `in` or `undec`)
5. $\vdash^m (\neg(q_i \wedge q_j))$, $i \neq j$, $i, j \in \{0, 1, ?\}$, $m \geq 0$ (no argument has more than one label)

Let Δ be the above theory and let $AF = (Ar, att)$ be an argumentation framework and $\mathcal{L}$ be a labelling of AF. Then for any $a \in Ar$ it holds that $a \vDash \Delta(q_1, q_0, q_?)$, provided that
$a \vDash q_1$ iff $\mathcal{L}(a) = \texttt{in}$

$a \vDash q_0$ iff $\mathcal{L}(a) = \texttt{out}$
$a \vDash q_?$ iff $\mathcal{L}(a) = \texttt{undec}$

Conversely if (S, R, h) is a modal model of Δ (i.e. for all $t \in S$, $t \vDash \Delta$) then it is an argumentation system with
$\mathcal{L}(a) = \texttt{in}$ iff $a \vDash q_1$
$\mathcal{L}(a) = \texttt{out}$ iff $a \vDash q_0$
$\mathcal{L}(a) = \texttt{undec}$ iff $a \vDash q_?$
This turns g into a complete labelling (see Theorem 7 and Proposition 1).

The above is may be a nice model but there is still not much we can do with it. Since the extensions are assignments to $q_0, q_1, q_?$ satisfying Δ we cannot directly talk about them, except for the grounded labelling. We shall see later how to deal with the other labellings, using circumscription. We shall see that to be able to fully understand our metalevel modal model and its options we should also introduce and compare with the classical logic metamodel. This we shall do in section 5.

Using the modality $\Box$ as above and results from [5], we can define extensions. Let E be a propositional letter. Then E defines a set of points in Ar. So according to our notation, $t \vDash \Box E$ iff for all s such that sRt we have $s \vDash E$.

So if E denotes a set of arguments, the $\Diamond E$ is the set of arguments attacked by E and $\Box\Diamond E$ is the set of arguments defended by E. We have:

1. E is stable iff $E = \neg\Diamond E = \Box\neg E$
2. (a) E is conflict-free iff $E \to \neg\Diamond E$
   (b) E is admissible iff $E \to (\Box\Diamond E) \wedge (\Box\neg E)$ iff $E \to \Box(\neg E \wedge \Diamond E)$
3. E is a complete extension iff $E = \Box(\neg E \wedge \Diamond E)$

Using fixed points methods of Section 4.5 and reference [15], we can find the fixed point solutions for (1) (stable extensions) and (3) (complete extensions) for finite frames.

Altogether, the metalevel approach can be summarized as follows. Given $AF = (Ar, att)$

1. We view elements $a \in Ar$ as possible worlds and so AF becomes a model for modal logic $\mathcal{K}$ (with the modality $\Box$).
2. Labellings become assignments in the modal logic.
3. The frame of our modal logic is fixed, it is (Ar, att). What changes are the assignments defined by the labellings. This means we have what we call a fixed frame modal logic FF modality.
4. Properties of labellings are studied in a circumscription logic based on modal logic $\mathcal{K}$ (to be introduced in Section 5.

4.5. The object level approach

The previous view used modal logic to talk about argumentation. The now to be introduced object level approach will model argumentation from within. To introduce our point of view, let us ask what does an argumentation framework of the form $AF = (Ar, att)$ say to us? Here we view the arguments $a \in \mathcal{A}rgs$ as atoms in some logic. So let X be the logical content of the argumentation framework AF and labelling $\mathcal{L}$. Hence if a is `in` then $X \vdash a$ in some logic. If a is `out` then $X \vdash \neg a$ and if a is `undec` then we have neither. We might take the simple minded view and let $X_{\mathcal{L}} = \{a \mid \mathcal{L}(a) = \texttt{in}\} \cup \{\neg a \mid \mathcal{L}(a) = \texttt{out}\}$ and propose this as a solution. The problem, however, is that this is too simplistic because

1. It does not take into account the internal structure of AF (i.e. the attack relation)
2. It has an explicit dependence on $\mathcal{L}$

Our aim is therefore to introduce a more sophisticated approach.

We borrow the Löb modal provability logic and use a suitable extension of it, which we call $LN1$, and view the logical content of $(AF, \mathcal{L})$ as a formula $\mathcal{M}(AF, \mathcal{L})$ of $LN1$. $LN1$ is a fixed frame modal logic, the frame being a chain of 3 elements. We should therefore have (using $\boxminus$ as the modal provability operator)

1. $\mathcal{M}(AF, \mathcal{L}) \vdash \boxminus a$ if a is `in`
2. $\mathcal{M}(AF, \mathcal{L}) \vdash \boxminus \neg a$ if a is `out`
3. neither, if a is `undec`

We read $\boxminus$ as provability. Figure 6 shows how we find $\mathcal{M}(AF, \mathcal{L})$.

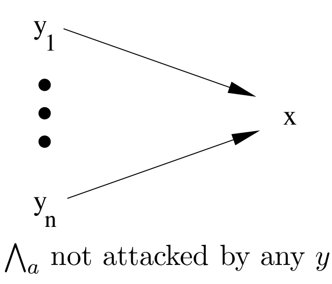

Figure 6. Finding $\mathcal{M}(AF, \mathcal{L})$.

Let $x \in Ar$ and let $y_1, \ldots, y_n$ denote all arguments attacking it (i.e. $(y_i, x) \in att$). The labelling conditions basically say that $\mathcal{L}(x) = \texttt{in}$ iff

for each attacker y_i it holds that $\mathcal{L}(y_i)$ is `out`. So if `in` means provable, we get $x \leftrightarrow \bigwedge_{y_i} \neg \boxminus \neg(\mathcal{M}(AF, \mathcal{L}) \wedge \neg y_i)$
so we have something like
$\mathcal{M}(AF, \mathcal{L}) \leftrightarrow \wedge_a$ not atttacked by any
$y \boxminus a \wedge [\bigwedge_{a \in Ar, x}$ attacked by $_{y_1, \ldots, y_n}$ $(x \leftrightarrow \bigwedge_{y_i} \neg \boxminus \neg(\mathcal{M}(AF, \mathcal{L}) \wedge \neg y_i))]$.

The exact solution formula for $\mathcal{M}(AF, \mathcal{L})$ is given in the beginning of Section 6.

Thus, $\mathcal{M}(AF, \mathcal{L})$ is obtained as a fixed point solution in a suitable modal provability logic. The fixpoint equation is generated by AF. All models of $\mathcal{M}(AF)$ should give us all labellings of AF. A model of $\mathcal{M}(AF)$ gives the atoms, the elements of Ar assignments and from the assignments we can tell which a is `in`, `out` or `undec`. This we will see in later sections.

Altogether, the object level approach can be summarised as follows. Let $AF = (Ar, att)$.

- We regard elements of Ar as atomic propositions in some modal provability logic $LN1$. The graph (Ar, att) generates a fixed point equation in the modal logic and the unique solution $\mathcal{M}(AF)$ of this equation is a formula of modal logic representing the logical content of AF.
- The possible world models of $\mathcal{M}(AF)$ are in one to one correspondence with all labellings $\mathcal{L}$ of AF.
- The logic we use, $LN1$ is a fixed frame modal logic.

4.6. The Mixed Approach

We saw that the metalevel approach regards arguments as possible worlds, while the object level approach regards them as propositions in a logic. The problem in both approaches is how to characterise extensions. We will now introduce a mixed approach. Consider the previous two approaches. In each of them, the fixed argumentation framework gives rise to a fixed frame modal logic. On the metalevel approach, the fixed frame is (Ar, att), and the set of assignments H is what all the possible labellings give us. On the object level approach the fixed frame is a chain of three elements (see Figure 5) and the assignments are all those assignments $\mathcal{L}$ that give us models of $\mathcal{M}(AF)$. We shall see later that there is a one-to-one correspondance between the assignments which are models of $\mathcal{M}(AF)$ and the labellings of AF.

In either case we get a fixed frame modal logic, one with $\Box$ and the other with $\boxminus$. The mixed approach adds the modality $\boxdot$ to the existing modality and forms the universal products with $\{\Box, \boxdot\}$ and with $\{\boxminus, \boxdot\}$ respectively.

To show how the mixed approach works, consider the semi-stable semantics of Table 1. For this we need to say that our labelling has a minimal `undec`. Let us explain the approach for the metalevel mixed case. Let g_{s-s} be the assignment arising from the semi-stable labelling. Then there is no other h for a model such that $h(q_2) \subsetneqq g_{s-s}(q_?)$. We can express this in the universal model.

The grounded extension for example is characterised by minimal `in`, so we obtain $\vDash q_1 \wedge \boxdot q_1$.

5. The metalevel approach

We saw that the metalevel approach uses modal logic $\mathcal{K}$ to talk about argumentation frameworks of the form $AF = (Ar, att)$. The arguments become the possible worlds of a Kripke model, the attack relation becomes the accessibility relation and an associated labelling $\mathcal{L}$ gives rise to and assignment. By varying $\mathcal{L}$ we get a fixed frame modal logic based on $\mathcal{K}$. To fully appreciate the advantages and limitations of this approach we need to compare it to two other approaches, the Logic Programming and the classical logic ones.

5.1. The Classical Logic Metalevel Approach

We start with classical logic with equality ("=") and three modal predicates Q_1, Q_0 and $Q_?$ and a binary relation R. Given an argumentation framework $AF = (Ar, att)$ and a labelling $\mathcal{L}$ we construct the associated model of classical logic. We take Ar as the domain, att as the relation R and use $\mathcal{L}$ to get the extensions of tree predicates Q_0, Q_1 and $Q_?$ as follows.

1. $a \in Q_1$ iff $\mathcal{L}$ labels a as `in`
2. $a \in Q_0$ iff $\mathcal{L}$ labels a as `out`
3. $a \in Q_?$ iff $\mathcal{L}$ labels a as `undec`

Consider the following classical theory $\Delta(R, Q_0, Q_1, Q_?)$.

1. $\forall x(Q_0(x) \vee Q_1(x) \vee Q_?(x))$
2. $\neg\exists x(Q_i(x) \wedge Q_j(x))$ for $i \neq j$, $i, j \in \{0, 1, ?\}$
3. $\forall y(\forall x(xRy \to Q_0(x)) \to Q_1(y)$
4. $\forall y(\exists x(xRy \wedge Q_1(x)) \to Q_0(y))$
5. $\forall y(\forall x(xRy \to (Q_0(x) \vee Q_?(x))) \wedge \exists x(xRy \wedge Q_?(x)) \to Q_?(y))$

Any model of Δ with domain D defines an argumentation framework with $Ar = D$, $att = R$ and $\mathcal{L}$ is what we obtain from the elements satisfying the respective predicates Q_0, Q_1 and $Q_?$. Notice that we are not using "=".

If we want to characterise any specific argumentation framework $AF = (Ar, att)$ we need equality and we need constant names for every element of Ar. We write the following additional axioms $\theta(AF)$.

6. $\forall x(\bigvee_{a \in Ar} x = \underline{a})$
7. $\bigwedge_{a,b \in Ar, a \neq b} \underline{a} \neq \underline{b}$
8. $\bigwedge_{a,b \in att} \underline{a} R \underline{b}$

We want to see how to characterise the different extensions of the argumentation frameworks obtained in classical logic as models of Δ. This means for example that we want to say that the predicate Q_1 is minimal in the model (in order to obtain the grounded extension) or for example that the predicate $Q_?$ is minimal to get the semi-stable extension, or that the predicate Q_1 is maximal to get the preferred extension. The concept "Q is maximal" or "Q is minimal" is not first order. We therefore need second order formula to express this, an approach that is know as predicate circumscription of John McCarthy; the subject has a very well developed theory.

So the theory we want is for example

$$\Delta_{semi-stable}(R, Q_0, Q_1, Q_?) = \Delta(R, Q_0, Q_1, Q_?) \wedge \text{ "}Q_? \text{ is minimal".}$$

Any model of $\Delta_{semi-stable}$ yields an argumentation framework based on the domain of the model, with $\mathcal{L}$ (derived from Q_1, Q_1, $Q_?$) which is semi-stable.

Using circumscription we write

$$\begin{array}{rcl} \Delta_{semi-stable} & = & \Delta(AF, Q_0, Q_1, Q_?) \wedge \forall Q_0 \forall Q_1 \forall Q_?^*((Q_?^* \subsetneq Q_?) \\ & & \rightarrow \neg\Delta(AF, Q_0, Q_1, Q_?^*)) \end{array}$$

where $X \subsetneq Y$ is defined as $\forall y(X(y) \rightarrow Y(y)) \wedge \exists z(Y(z) \wedge \neg X(z))$.

Similarly to maximise we use $\supsetneq$ in the circumscription formulas for Q_1 and Q_1^*. According to our book [18] it is possible to eliminate such second order quantifiers under certain circumstance. We need to check whether this can be done in our case.

When we deal with a specific finite argumentation framework AF, our set of axioms is $\Delta(AF) = \Delta \cup \theta(AF)$. In this case characterising some of the extensions is easier. For one thing we can use provability to characterise some extensions and for others circumscription becomes first order.

The above axioms $\Delta(AF) = \Delta \cup \theta(AF)$ can immediately characterise the grounded extension of AF as the set of all x such that $\Delta(AF) \vdash Q_1(x)$.

The preferred semantics is characterised by maximal `in`, which implies that Q_1 is maximal, so $\neg Q_1$ is minimal. So let $\overline{Q}_1^{min} = \{x \mid \Delta(\mathcal{L}) \vdash \neg Q_1\}$ and add the axiom $\forall(Q_1(x) \leftrightarrow \overline{Q}_1^{min}(x))$. The problem with the above definition is that $Q_?^{min}$ and $\overline{Q}_1^{min}$ are defined using provability, which is outside the logic itself. To bring this in, we need to use circumscription as we did before. However, since for a given AF, Ar is finite, we can turn the second order qualtifier $\forall X$ into a big conjunction by enumerating all possible subsets $B \subseteq Ar$. We get $\forall X \Psi[y \in X]$ is replaced by $\bigwedge_{B \subseteq Ar} \Psi[\bigvee_{a \in B} y = \underline{a}]$. This makes circumscription first order for AF fixed.

5.2. The Logic Programming Metalevel Approach

The logic programming metalevel approach has been worked out in [29]. The approach is similar to the classical logic approach except that we use logic programming to represent the argumentation framework. The extensions correspond to the models of the corresponding logic program. For each argument x we write the clause $x \leftarrow \neg y_1, \ldots, \neg y_n$ $(n \geq 0)$, where $\{y_1, \ldots, y_n\}$ is the set of all arguments attacking x. We get a resulting logic program satisfying

1. each atom is the head of exactly one clause
2. the bodies of clauses consist of weakly negated atoms only

For more information, we refer to [29] and [16].

6. The Modal Provability Object Level Approach

Let $AF = (Ar, att)$ be an argumentation framework. Let x be an argument whose set of attackers is $y_1, \ldots, y_n$, as was illustrated in Figure 6.

We construct the following modal formula in the modal logic $LN3$ of provability.

$$\begin{aligned}\mathcal{M}(AF) \quad = \quad & G(\Box\bot \vee \textstyle\bigwedge_{y \in Ar \text{ and } y \text{ has attackers } y_i} y \leftrightarrow \bigwedge_i \Diamond\neg y_i) \wedge \\ & \textstyle\bigwedge_{y \in Ar \text{ and } y \text{ is not attacked}} Gx\end{aligned}$$

Here, GA stands for $A \wedge \Box A$. The logic $LN3$ has the following axioms.

1. all substitution instances of classical tautologies
2. all $K4$ axioms

- $\Box(A \to B) \to (\Box A \to \Box B)$
- $\Box A \to \Box\Box A$
- $\dfrac{\vdash A}{\vdash \Box A}$

3. all substitution instances of Löb's axioms
   $\Diamond A \to \Diamond(A \wedge \Box\neg A)$
4. linearity axiom
   $\Diamond A \wedge \Diamond B \to \Diamond(A \wedge B) \vee \Diamond(A \wedge \Diamond B) \vee \Diamond(B \wedge \Diamond A)$
5. 3-chain axiom
   $\Box\Box\Box\bot$
6. axioms for atoms q only
   - $q \to \Box(\neg q \to \Box q)$
   - $\Box(\Box\bot \vee q) \leftrightarrow \Box q$
   - $\Box(\Box\bot \vee \neg q) \leftrightarrow \Box\neg q$

Lemma 5.

1. *The logic $LN3$ is complete for all 3-chain models whose assignment to atoms has one of the types of Table 2.*
2. *Any 3-chain model of $LN1$ allows for the atoms to have one of types 1, 0 or 2 assignments.*

Proof. The axioms of **LN1** force the following for atomic q in the chain $1 < 2 < 3$:

- $1 \vDash q$ and $2 \vDash q \Rightarrow 3 \vDash q$
- $1 \vDash\sim q$ and $2 \vDash\sim q \Rightarrow 3 \vDash\sim q$
- $1 \vDash q$ and $2 \vDash\sim q \Rightarrow 3 \vDash q$.

We need to prove that the option "$1 \vDash q$ and $2 \vDash\sim q$ and $3 \vDash\sim q$" cannot arise. Consider a point y such as y is being attacked. (If not then Gy is in $M(AF)$ and so y is of type 1.) We have $G(\Box\bot \vee (y \leftrightarrow \bigwedge \Diamond \sim y_i))$ holds in the model. Hence $y \leftrightarrow \bigwedge_i \Diamond \sim y_i$ holds at points 1 and 2. We distinguish several cases:

1. If for any j, if $y_j = \bot$ at 3 then $\Diamond \sim y_j$ is true at 1 and 2 and it plays no role in the conjunction. If for all $y_j, 3 \vDash\sim y_j$ then $y = \top$ at 1 and 2 and y is of type 1.
2. Some y_j are $\top$ at 3. Then $\Diamond \sim y_j$ is $\bot$ at 2 and $y = \bot$ at 2.

3. We now check whether y_j is true or false at 2. If for some j, $y_j = \top$ at 2 then $\Diamond \sim y_j = \bot$ at 1 and $y = \bot$ at 1 and hence $y = \bot$ everywhere, and hence y is of type 0.

If for all $j, y_j = \bot$ at 2 then $y = \top$ at 1 and we have that $y = \top$ at 1 and $y = \bot$ at 2 and we must have by the axiom 4.1 that $3 \vDash y$ and y is of type 3. ■

world	type 1	type 0	type 2
3	$\top$	$\bot$	$\top$
2	$\top$	$\bot$	$\bot$
1	$\top$	$\bot$	$\top$

Table 2. 3-chain models.

THEOREM 27. *Let $AF = (Ar, att)$ be an argumentation framework and let $\mathcal{M}(AF)$ be the modal sentence as defined. Then:*

1. *Any complete labelling $\mathcal{L}$ for AF gives rise to a model of $\mathcal{M}(AF)$, where for any q, the correspondence is as in Table 2.*
   - *q is assigned type 1 assignment iff $\mathcal{L}(q) = 1$*
   - *q is assigned type 0 assignment iff $\mathcal{L}(q) = 0$*
   - *q is assigned type 2 assignment iff $\mathcal{L}(q) = ?$*
2. *Let $M_1, M_2, M_3, \ldots$ be all the 3-chain models of $\mathcal{M}(AF)$. Then $\{M_i\}$ are all the complete labellings for AF, where these complete labellings are obtained as in Table 2.*

PROOF.

Direction 1. Let h be an assignment model satisfying $M(AF)$. By Lemma 5 every atom node x in P gets assigned values $h(x)$ of types 0, 1, or ?.

Define a Caminada candidate function L according to the types, namely $\mathcal{L}(x) = i$ iff $h(x)$ is of type $i, i \in \{0, 1, ?\}$. We now show that L satisfies the conditions of a Caminada function of Definition 5.

1 If x is an initial point then Gx is a conjunct of $M(AF)$, therefore $h(x)$ is of type 1 and hence $\mathcal{L}(x) = 1$.

2 If $y_1, \ldots, y_n$ are all nodes with arrows to y, then we have the conjunct $y \leftrightarrow \bigwedge_i \Diamond \sim y_i$ in $M(AF)$ in the clause

$$G(\Box\bot \vee \bigwedge_y (y \leftrightarrow \bigwedge_i \Diamond \sim y_i))$$

This means that at nodes 1 and 2 of the chain $y \leftrightarrow \bigwedge_i \Diamond \sim y_i$ must hold. Assume all y_i are of type 0, then $y = \top$ at 2 and 1. Hence by axiom 6.2, $y = \top$ also at node 3 and y is of type 1. This means that if all y_i are 'out' then y is 'in'.

Assume one of y_i is of type 1. Then for this $y_i, \Diamond \sim y_i$ is $\bot$ at nodes 1 and 2. Hence y is false at 1 and 2 and by axiom 4.3, $y = \bot$ at node 3. Hence y is of type 0.

Now assume all of y_i are either of type 0 or of type ?, and at least one of y_i say y_1 is of type?. The y_i of type 0 have no influence on y because $\Diamond \sim y_i$ is $\top$ at nodes 1 and 2. The crucial nodes are the nodes like y_1 which are of type ?. This means that $1 \vDash y_1, 2 \vDash \sim y_1, 3 \vDash y_1$. Thus $\Diamond \sim y_1 = \bot$ at 2 and $\Diamond \sim y_1 = \top$ at 1.

This holds for any type y_i of type ?. Thus $y = \top$ at 1 and $y = \bot$ at 2. Hence by axiom 6.1, $y = \top$ at 3 and hence y is of type ?.

Direction 2. Let L be a Caminada function. Define a model by assigning values to the propositions according to the code of Table 2. We claim $M(AF)$ holds in this model.

First all nodes y without arrows into them are assigned type 1 and hence Gy holds. For the other nodes we must check the formula

$$G(\Box\bot \vee \bigwedge_{\substack{y \\ y \text{ has arrows leading to it}}} (y \leftrightarrow \bigwedge_i \Diamond \sim y_i))$$

and show it holds in the model.

We distinguish several cases. Assume all y_i are of type 0. This means $\mathcal{L}(y_i) = 0$ for all y_i. So $\Diamond \sim y_i$ is $\top$ at nodes 1 and 2. But since all $L(y_i) = 0$ we get that $\mathcal{L}(y) = 1$ and hence $y \leftrightarrow \bigwedge_i \Diamond \sim y_i$ holds at 1 and 2 as required.

If one of y_i is of type 1, this means $\Diamond \sim y_i$ is $\bot$ at 1 and 2. Hence $\bigwedge_i \Diamond \sim y_i$ is $\bot$ at 1 and 2. But also since $\mathcal{L}(y_i) = 1$, we get $\mathcal{L}(y) = 0$ and so y is of type 0 and $y = \bot$ at 1 and 2.

Hence $y \leftrightarrow \bigwedge_i \Diamond \sim y_i$ holds at 1 and 2.
Now assume that all y_i are either of type 0 or type ?. This means $\mathcal{L}(y_i)$ is either 0 or ?. Assume that at least one y_i is of type ? (i.e. $\mathcal{L}(y_i) =?$). Then $\mathcal{L}(y) =?$ and we have $y = \top$ at 1 and $y = \bot$ at 2. Let us check whether $\bigwedge_i \Diamond \sim y_i$ is $\bot$ at and $\top$ at 2. Since all y_i are of type ? or $\bot$ with at least one y_i of type ?, the type ? y_i will be $\top$ at node 2 and hence $\Diamond \sim y_i = \bot$ at node 1 and hence $y = \bot$ at node 1.
On the other hand all y_i are $\bot$ at node 3 and hence $\bigwedge_i \Diamond \sim y_i$ is $\top$ at node 2. This shows that $y \leftrightarrow \bigwedge_i \Diamond \sim y_i$ is $\top$ at nodes 1 and 2.
Thus the above argument shows that

$$G(\Box\bot \vee \bigwedge_u (y \leftrightarrow \bigwedge_i \Diamond \sim y_i))$$

holds at all nodes 1, 2, and 3.

This completes the proof of the theorem. ■

COROLLARY 1. *$\mathcal{M}(AF)$ characterises all the extensions of AF through the modal logic LN3.*

In particular, we obtain the following theorem.

THEOREM 28. *x is in the grounded extension of AF iff $\mathcal{M}(AF) \vdash_{LN3} Gx$.*

PROOF. This holds because the grounded extension is the minimal complete extension [12]. ■

Since Gx is in every model of $\mathcal{M}(AF)$ we cannot characterise other extensions, e.g. preferred extensions, in a similar way. We need the mixed approach with additional modalities.

7. Discussion

Grossi in [19] uses modal logic to represent argumentation frameworks. His approach is metalevel. We use (in Section 5.1) classical logic to talk about argumentation frameworks and use circumscription to define the various extensions. Grossi uses modal logic to describe the argumentation frameworks. He needs two modalities, one to go with the attack relation and one to go with the converse of the attack relation (like two temporal logic modalities). He also uses a universal modality and to get the extensions he needs

μ-calculus on top. Our approach, on the other hand, is to use classical logic with circumscription to do the job. We would not be surprised if the approach of Grossi could simulate the relevant classical logic with circumscription, since all the ingredients are present. Note that our use of modal logic in section 6 is object level and is completely different from its use as a metalevel tool.

As was mentioned in Section 3 the approach of argument labellings can be traced back to Pollock, who in his 1995 book [21] described his OSCAR system in terms of labellings. As explained in [20], Pollock's approach essentially boils down to preferred semantics. The labelling approach of Jakobovits and Vermeir [20] is aimed not so much at describing Dung's original semantics but rather to defining additional semantics driven by what they perceive to be problems in Dung's original semantics. Caminada first described complete semantics in terms of labellings in [8] and showed how this can be used to provide labelling based descriptions of other semantics as well. This approach was then applied in [10] and [28] to provide labelling-based algorithms for computing particular argumentation sets and extensions.

Besnard and Doutre [5] examine how complete and preferred semantics can be expressed in terms of set theoretical equations, but do not provide a logical account of these semantics.

A connection between the current work and the topic of linear programming equations can be found in [14]. We did use some of the equations of [14] in Section 4.4.

References

[1] Amgoud, L., N. Maudet, and S. Parsons, 'Modelling dialogues using argumentation', *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-00)*, Boston, MA, 2000, pp. 31–38.

[2] ASPIC-consortium, Deliverable D2.5: Draft formal semantics for ASPIC system, June 2005.

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

[4] Baroni, P., and M. Giacomin, 'On principle-based evaluation of extension-based argumentation semantics', *Artificial Intelligence* 171 (10-15): 675–700, 2007.

[5] Besnard, Ph., and S. Doutre, 'Characterization of semantics for argument systems', *Proceedings of the Ninth International Conference on the principles of Knowledge Representation and reasoning*, AAAI Press, 2004, pp. 183–193.

[6] Boella, G., J. Hulstijn, and L. van der Torre, 'A logic of abstract argumentation', *Proceedings of the Workshop on Argumentation in Multi-Agent Systems (ArgMAS)*, 2005.

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

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

[9] Caminada, M. W. A., 'Semi-stable semantics', in P. E. Dunne and T. J. M. Bench-Capon (eds.), *Computational Models of Argument; Proceedings of COMMA 2006*, IOS Press, 2006, pp. 121–130.

[10] Caminada, M. W. A., 'An algorithm for computing semi-stable semantics', *Proceedings of the 9th European Conference on Symbolic and Quantitalive Approaches to Reasoning with Uncertainty (ECSQARU 2007)*, number 4724 Springer Lecture Notes in AI, Springer Verlag, Berlin, 2007, pp. 222–234.

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

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

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

[14] Gabbay, D. M., 'Fibring argumentation frames', *Studia Logica* 93 (2-3): 231–295, 2009, this issue.

[15] Gabbay, D. M., 'Modal provability foundations for argumentation networks', *Studia Logica* 93 (2-3): 181–198, 2009, this issue.

[16] Gabbay, D. M., and A. S. d'Avila Garcez, 'Logical modes of attack in argumentation networks', *Studia Logica* 93 (2-3): 199–230, 2009, this issue.

[17] Gabbay, D. M., A. Kurucz, F. Wolter, and M. Zakharyaschev, *Many-dimensional Modal Logics: Theory and Applications*, Number 148 *Studies in Logic and the Foundations of Mathematics*, Elsevier, 2003.

[18] Gabbay, D. M., R. Schmidt, and A. Szałas, *Order Quantifier Elmination*, College Publications, 2008.

[19] Grossi, D., *Doing argumentation theory in modal logic*, Technical Report PP-2009-24, ILLC, 2009.

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

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

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

[23] Rahwan, I., F. Zablith, and C. Reed, 'Laying the foundations for a world wide argument web', *Artificial Intelligence* 171 (10-15): 897–921, 2007.

[24] Simari, G. R., and R. P. Loui, 'A mathematical treatment of defeasible reasoning and its implementation', *Artificial Intelligence* 53: 125–157, 1992.

[25] Smorynski, C., 'Modal logic and self reference', in D. M. Gabbay and F. Guenter (eds.), *Handbook of Philosophical Logic, 1st edition*, volume 2. Kluwer, 1984.

[26] Verheij, B., 'A labeling approach to the computation of credulous acceptance in argumentation', in M. M. Veloso (ed.), *Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India*, 2007, pp. 623–628.

[27] Vreeswijk, G. A. W., *Studies in defeasible argumentation*, PhD thesis at Free University of Amsterdam, 1993.

[28] Vreeswijk, G. A. W., 'An algorithm to compute minimally grounded and admissible defence sets in argument systems', in P. E. Dunne and T. J. M. Bench-Capon (eds.), *Computational Models of Argument; Proceedings of COMMA 2006*, IOS, 2006, pp. 109–120.

[29] Wu, Y., M. Caminada, and D. M. Gabbay, 'Complete extensions in argumentation coincide with 3-valued stable models in logic programming', *Studia Logica* 93 (2-3): 383–403, 2009, this issue.

Martin W. A. Caminada
Interdisciplinary Lab
for Intelligent and Adaptive Systems
University of Luxembourg
martin.caminada@uni.lu

Dov M. Gabbay
King's College, London
Bar-Alan University, Israel
University of Luxembourg
dov.gabbay@kcl.ac.uk