Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 344 (2019) 3–23

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

On Three-Valued Acceptance Conditions of Abstract Dialectical Frameworks

João Alcântara[1]

Computer Science Department
Universidade Federal do Ceará
Fortaleza, Brazil

Samy Sá[2]

Computer Science Department
Universidade Federal do Ceará
Quixadá, Brazil

Abstract

Abstract Dialectical Frameworks (ADFs) are generalizations of Dung's abstract argumentation frameworks (AFs) where each argument has an associated acceptance condition expressed by a boolean formula. The resulting extension is robust enough not only to model the attack relation original to AFs, but also others types of dependencies and interactions between arguments. A recent development in ADFs proposed an alternative formalization involving three-valued acceptance conditions, connecting the original definitions of ADFs to the concept of three-valued argument labellings, a core concept in computational argumentation literature. In this paper, we revise some of the main semantics defined under this three-valued approach and prove our definitions hold equivalence to well-known semantics of AFs.

Keywords: Abstract Dialectical Frameworks, Argument Labellings, Argumentation, Knowledge Representation, Uncertainty.

1 Introduction

One of the main concerns in the field of Knowledge Representation and Reasoning is to answer how conclusions can be reached through reasoning. A possible path is to resort to Formal Logic to symbolically represent the knowledge required to solve a problem and ensure that the performed reasoning is sound and complete. An ongoing trend resides in the subfield of Computational Argumentation, where the connections between argumentation and Computer Science are exploited to performed both deductive and nonmonotonic or defeasible reasoning [22,23,30,31,25].

[1] Email: jnando@lia.ufc.br

[2] Email: samy@ufc.br

https://doi.org/10.1016/j.entcs.2019.07.002

In this trend, the reasoning process is characterized by constructing and evaluating arguments, which (roughly speaking) can be understood as a sets of reasons for the validity of particular claims.

A core reference in this area, Dung's Abstract Argumentation frameworks (AFs) [15] models arguments as abstract entities among which an attack relation is defined. Dung proposed important concepts widely adopted by the computational argumentation community, such as the concept of argument admissibility. This concept builds on the intuition that to accept a claim c, one needs both to have an argument A supporting c and to justify the rejection of all other arguments that conflict with A. In short, while in Logic the reasoning process is focused on the inference within an argument, in argumentation the reasoning process also involves the interactions between arguments.

A crucial innovation of Dung's approach is to separate the properties depending exclusively on argument attack from those related to the structure of the arguments [28]. Under a mathematical point of view, the thus resulting argumentation framework is a directed graph, the nodes of which are the arguments, and the edges represent the conflicts between the arguments. A semantics for an argumentation framework returns acceptable sets of arguments, the extensions of the framework. It should be clear that different criteria of acceptability will specify different semantics, including Dung's original concepts of complete, stable, preferred and grounded extension-based semantics [15] and semi-stable semantics [29,8].

Dung's argumentation frameworks achieved remarkable popularity, but they have also been subject to criticism. A contentious issue refers to their alleged limited expressive capabilities of AFs, as they lack certain features which are common in almost every form of argumentation to be found in practice [6]. Indeed, in Dung's argumentation frameworks the only interaction between atomic arguments is given by the attack relation. As overviewed in [5], many proposals extending AFs can be found in the literature; it includes among others AFs with support relations [12,13,20,21], attacks on attacks [19] and AFs with weights [18,16,14].

With such motivation, Brewka and Woltran introduced Abstract Dialectical Frameworks (ADFs) in [6,4], a generalization of Dung's argumentation frameworks that allows arbitrary relationships among arguments to be expressed. Their approach attempts to unify several of those different extensions of AFs and to generalize Dung's AFs in a principled, systematic way. In an ADF, besides the attack relation, arguments may support each other, or a group of arguments may jointly attack another while each single member of the group is not strong enough to do so [3]. This additional expressiveness is obtained by associating to each node (argument) its two-valued acceptance conditions which can be expressed as arbitrary propositional formulas. The intuition is that an argument is accepted if its associated acceptance condition is accepted. In short, we can characterize ADFs as dependency graphs + acceptance conditions.

An alternative formalization for ADFs was proposed in [1], where the authors introduced three-valued acceptance conditions for arguments. This move was motivated by the understanding that in an AF semantics, a given argument can be either accepted, rejected or undecided, but the two-valued acceptance conditions of [6,4] do not explicitly evaluate arguments as undecided. Due to this difference, the resulting ADFs of [1] yield a relatively simpler formalization which holds the same results of [6], while also capturing the semi-stable semantics of AFs [29,8], which was not contemplated in [6]. To illustrate the difference, we provide an example.

Example 1.1 Let $A = \{a, b, c\}$ be a set of arguments. To depict an ADF, we need to specify their acceptance conditions. Consider the two scenarios below where for each argument $s \in A$, its associated acceptance condition is represented as a propositional formula written within square brackets on the right of s:

I: $a \quad [b \vee c]; \quad b \quad [\neg b]; \quad c[\top]$

II: $a \quad [(b \wedge \neg c) \vee (\neg b \wedge c) \vee (b \wedge c)]; \quad b \quad [\neg b]; \quad c[\top]$

From scenario I we can observe a is accepted if b or c is accepted; b is accepted if b is not accepted, and c is always accepted. Assuming a two-valued approach for these acceptance conditions, we conclude scenario II is obviously equivalent to scenario I. However, as the arguments themselves are evaluated according to three-valued labellings, it is arguable that the acceptance conditions are evaluated according to only two truth values. By posing a three-valued approach, we are basically making a distinction between non acceptance and rejection. In that case, the two scenarios above will not be equivalent: suppose we have c is accepted and b is undefined. In scenario I, we have a will also be accepted; in scenario II only c will be accepted, for a can only be accepted if we accept or explicitly reject b.

In this paper, we take advantage of the formalization of [1] and revise their definition of complete labellings to seek fixpoints of an immediate consequences operator that are minimal concerning a truth ordering that is commonly employed in logic programming theory. As we will show, this difference will guarantee an equivalence result between ADF semantics and Dung's AF semantics for every ADF whose acceptance conditions model the attack relation of some AF. As a consequence of that result, ADF semantics will be proven to extend well-known semantics for AFs such as complete, stable, preferred and grounded semantics [15] as well as the semi-stable semantics [29,8].

The paper proceeds as follows. Firstly we recall the necessary background on Dung's abstract argumentation frameworks and present some of the well-known and well-established argumentation semantics in terms of argument labellings. Next we present ADFs with three-valued acceptance conditions, accommodating important notions such as that of complete labellings and of an immediate consequences operator Γ_D. In the subsequent section, we examine the relation between ADFs and AFs by presenting a subclass of ADFs which corresponds to AFs. Some of our main results will be introduced at this point. Finally, we round off with a discussion of the obtained results and pointers for future works.

2 Preliminaries

In this section we overview the main definitions connected with (Abstract) Argumentation Frameworks [15]. We are particularly interested in how abstract argumentation semantics can be formulated in terms of labellings. In this piece of work, we will restrict ourselves to finite argumentation frameworks.

Definition 2.1 [Argumentation Frameworks][15] An *argumentation framework* AF *is a pair* (Ar, att) *where* Ar *is a finite set of arguments and* $att \subseteq Ar \times Ar$.

Arguments are related to others by the attack relation att: an argument A attacks B iff $(A, B) \in att$. An argumentation framework can be seen as a directed graph where the arguments are nodes and each attack is an arrow.

Given an argumentation framework, we are mainly interested in computing its semantics. The traditional approach of [15] does so by identifying sets of arguments (called extensions) based on the concept of admissibility (see [15]). The approach we will consider in this work is an alternative formalization of abstract argumentation semantics originally introduced in [7] using argument labellings instead of argument extensions:

Definition 2.2 [Argument Labellings][7] Let (Ar, att) be an AF. A labelling is a total function $\mathcal{L} : Ar : \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$.

We write $\mathtt{in}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \mathtt{in}\}$, $\mathtt{out}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \mathtt{out}\}$ and $\mathtt{undec}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \mathtt{undec}\}$. Sometimes, we write a labelling $\mathcal{L}$ as a triple (Ar_1, Ar_2, Ar_3) where $Ar_1 = \mathtt{in}(\mathcal{L})$, $Ar_2 = \mathtt{out}(\mathcal{L})$ and $Ar_3 = \mathtt{undec}(\mathcal{L})$. The label `in` is intended to express the argument is explicitly accepted; the label `out` is intended to express the argument is rejected, and the label `undec` is intended to express the status of the argument is undecided, i.e., no verdict is achieved about whether the argument should be `in` or `out`. We proceed by recalling the definition of admissible labelling:

Definition 2.3 [Admissible Labellings] [2] Let (Ar, att) be an AF.

- An `in`-labelled argument is said to be *legally* `in` iff all of its attackers are labelled `out`.
- An `out`-labelled argument is said to be *legally* `out` iff it has at least one attacker that is labelled `in`.

A labelling $\mathcal{L}$ is admissible if each `in`-labelled argument is legally `in` and each `out`-labelled argument is legally out.

According to this definition, for any argumentation framework, a labelling where all arguments are `undec` is admissible. By strengthening the basic requirements employed in the characterization of admissibility, we obtain the concept of complete labelling:

Definition 2.4 [Complete Labellings] [9] Let (Ar, att) be an AF. A labelling $\mathcal{L}$ is complete iff for each argument $A \in Ar$ it holds that

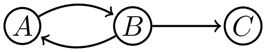
(i) $\mathcal{L}(A) = \texttt{in}$ iff every $B \in Ar$ attacking A has $\mathcal{L}(B) = \texttt{out}$.

(ii) $\mathcal{L}(A) = \texttt{out}$ iff there exists $B \in Ar$ attacking A such that $\mathcal{L}(B) = \texttt{in}$.

(iii) $\mathcal{L}(A) = \texttt{undec}$ iff not every $B \in Ar$ attacking A has $\mathcal{L}(B) = \texttt{out}$ and no $B \in Ar$ attacking A has $\mathcal{L}(B) = \texttt{in}$.

It is worthwhile noticing that every complete labelling is admissible. The following example illustrates both types of argument labellings:

Example 2.5 Consider AF $= (Ar, att)$ depicted below, where $Ar = \{A_1, A_2, A_3\}$ and $att = \{(A_1, A_2), (A_2, A_1), (A_2, A_3)\}$.



We have

- Admissible labellings: $(\{\}, \{\}, \{A, B, C\})$, $(\{A\}, \{B\}, \{C\})$, $(\{A, C\}, \{B\}, \{\})$, $(\{B\}, \{A\}, \{C\})$, $(\{B\}, \{A, C\}, \{\})$.
- Complete labellings: $(\{\}, \{\}, \{A, B, C\})$, $(\{A, C\}, \{B\}, \{\})$, $(\{B\}, \{A, C\}, \{\})$.

Now we can define the main semantics for AFs in terms of labellings:

Definition 2.6 [7,10] Let $\mathbb{L}$ be the set of all complete argument labelling of AF. Then

- $\mathcal{L}$ is the grounded argument labelling of AF if $\mathcal{L} \in \mathbb{L}$ and $\nexists \mathcal{L}' \in \mathbb{L}$ such that $\texttt{in}(\mathcal{L}') \subsetneq \texttt{in}(\mathcal{L})$.
- $\mathcal{L}$ is a preferred argument labelling of AF if $\mathcal{L} \in \mathbb{L}$ and $\nexists \mathcal{L}' \in \mathbb{L}$ such that $\texttt{in}(\mathcal{L}') \supsetneq \texttt{in}(\mathcal{L})$.
- $\mathcal{L}$ is a stable argument labelling of AF if $\mathcal{L} \in \mathbb{L}$ and $\texttt{undec}(\mathcal{L}) = \emptyset$.
- $\mathcal{L}$ is a semi-stable argument labelling of AF if $\mathcal{L} \in \mathbb{L}$ and $\nexists \mathcal{L}' \in \mathbb{L}$ such that $\texttt{undec}(\mathcal{L}') \subsetneq \texttt{undec}(\mathcal{L})$.

Note how complete argument labellings constitute the fundamental notion to characterize these semantics:

Example 2.7 Let AF $= (Ar, att)$ be an abstract argumentation framework such that $Ar = \{A, B, C, D\}$ and $att = \{(A, A), (B, A), (C, B), (C, D), (D, C)\}$. We depict AF as

Concerning semantics, AF has:

- Complete labellings: $(\{\}, \{\}, \{A, B, C, D\})$, $(\{B, D\} \{A, C\}, \{\})$ and $(\{C\}, \{B, D\}, \{A\})$.
- Grounded labelling: $(\{\}, \{\}, \{A, B, C, D\})$.
- Preferred labellings: $(\{B, D\} \{A, C\}, \{\})$ and $(\{C\}, \{B, D\}, \{A\})$.
- Stable labellings: $(\{B, D\} \{A, C\}, \{\})$.
- Semi-stable labellings: $(\{B, D\} \{A, C\}, \{\})$.

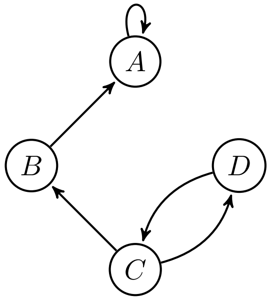


Fig. 1. An abstract argumentation framework with 6 arguments.

It is clear semi-stable labellings coincide with stable labellings whenever the framework has at least one stable labelling. Another important result showing how argument labellings relate to each other was proved in [7,11]:

Theorem 2.8 *[7,11] Let $\mathcal{L}$ be a complete argument labelling of an argumentation framework $F = (Ar, att)$. It holds*

- $\mathtt{in}(\mathcal{L})$ *is maximal (w.r.t. set-inclusion) among all complete argument labellings of F iff* $\mathtt{out}(\mathcal{L})$ *is maximal (w.r.t. set-inclusion) among all complete argument labellings of F.*
- $\mathtt{in}(\mathcal{L})$ *is minimal (w.r.t. set-inclusion) among all complete argument labellings of* AF *iff* $\mathtt{out}(\mathcal{L})$ *is minimal (w.r.t. set-inclusion) among all complete argument labellings of* AF*.*

Hence, concerning complete labellings, we can say that maximizing (resp. minimizing) the label `in` is equivalent to maximizing (resp. minimizing) the label `out`. This result will be employed in Section 4 (Theorem 4.4) to guarantee the semantics we will define for Abstract Dialectical Frameworks (Definition 3.9) are extensions of those in Definition 2.6.

3 Abstract Dialectical Frameworks

Abstract Dialectical Frameworks were introduced in [6] to treat arguments (called statements there) as abstract and atomic entities. It can be seen as a directed graph whose nodes represent statements which can be accepted or not. In addition, the links between nodes represent dependencies: the status of a node s only depends on the status of its parents (denoted $par(s)$), that is, the nodes with a direct link to s.

In this paper, we will propose a slight change to that approach: instead of resorting to subsets of $par(s)$ to assess the status of s, we will make this dependency between statements more explicit by resorting to subsets of $Lit(s)$:

$$Lit(s) = par(s) \cup \left\{\neg s' \mid s' \in par(s)\right\}$$

A set $M \subseteq Lit(s)$ is consistent iff there is no statement $s \in par(s)$ such that $\{s, \neg s\} \subseteq M$. We denote by $\mathfrak{C}_s = \{M \subseteq Lit(s) \mid M \textit{ is consistent}\}$ the set of all consistent subsets of $Lit(s)$. We will refer to $a \in S$ as a positive (resp. negative) statement in M if $a \in M$ (resp. $\neg a \in M$). Now we can accordingly alter the original definition of Abstract Dialectical Frameworks [6] as to obtain the following:

Definition 3.1 [Abstract Dialectical Frameworks] An abstract dialectical framework is a tuple $D = (S, L, C)$ where

- S is a set of statements (positions, nodes),
- $L \subseteq S \times S$ is a set of links,
- $C = \{C_s \mid s \in S\}$ is a set of total functions $C_s : \mathfrak{C}_s \to \{\mathtt{in}, \mathtt{out}\}$, one for each statement s, such that for any $M \in \mathfrak{C}_s$, if $C_s(M) = \mathtt{in}$, then $C_s(M') = \mathtt{in}$ for any $M' \supseteq M$. C_s is called the acceptance condition of s.

The function C_s for a statement s is intended to determine the acceptance status of s, which as before only depends on the status of its parent nodes $par(s)$. The novelty is that now for each node $s \in S$, its acceptance condition C_s specifies *explicitly* the exact conditions under which s is accepted, giving room to a (partial) three-valued setting. Intuitively,

- s will be accepted if there exists $M \subseteq \mathfrak{C}_s$ such that $C_s(M) = \mathtt{in}$ and every positive statement in M is accepted and every negative statement in M is rejected.
- s will be rejected if for every $M \subseteq \mathfrak{C}_s$ such that $C_s(M) = \mathtt{in}$, there exists either a rejected positive statement in M or an accepted negative statement in M.
- s will be undefined otherwise.

3.1 Representing Acceptance Conditions

The acceptance conditions in C of an ADF(S, L, C) can also be represented in two alternative ways:

- Any function $C_s \in C$ can be represented by the set of subsets of $\mathfrak{C}_s$ leading to the acceptance of s, i.e., $C_s^{\mathtt{in}} = \{M \in \mathfrak{C}_s \mid C_s(M) = \mathtt{in}\}$. We will indicate this alternative by denoting an ADF as $(S, L, C^{\mathtt{in}})$, where $C^{\mathtt{in}} = \left\{C_s^{\mathtt{in}} \mid s \in S\right\}$.
- Any function $C_s \in C$ can also be represented as a propositional formula φ_s over the vocabulary $\mathfrak{C}_s$ as follows:

$$\varphi_s \equiv \bigvee_{\{a_1,\ldots,a_m,\neg b_1,\ldots,\neg b_n\} \in C_s^{\mathtt{in}}} a_1 \wedge \cdots \wedge a_m \wedge \neg b_1 \wedge \cdots \wedge \neg b_n$$

If $m = n = 0$, the corresponding $a_1 \wedge \cdots \wedge a_m \wedge \neg b_1 \wedge \cdots \wedge \neg b_n \equiv \top$, where $\top$ is a constant always evaluated as $\mathtt{in}$. If there is no $M \in \mathfrak{C}_s$ such that $C_s(M) = \mathtt{in}$, then $\phi_s \equiv \bot$, where $\bot$ is a constant always evaluated as $\mathtt{out}$. We will indicate this alternative by denoting an ADF as (S, L, C^{φ}), where C^{φ} refers to the set $\{\varphi_s \mid s \in S\}$.

When referring to an ADF as (S, L, C^{φ}), we will assume the acceptance formulas specify implicitly the parents a node depends on. Then the set L of links between statements can be ignored, and the ADF can be represented as (S, C^{φ}), where L is recovered by $(a, b) \in L$ iff a appears in ϕ_b.

3.2 3-Valued Labellings for ADF's

In order to define different semantics for ADFs over statements s, we will resort to the notion of admissible labellings:

Definition 3.2 [Admissible Labellings] Let $D = (S, C^{\varphi})$ be an ADF. A three-valued labelling is a mapping $v : S \to \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$ that assigns the values (`in`), (`out`), or (`undec`) to each statement s. Labellings will be extended to assign values to formulas over statements according to Kleene's strong three-valued logic [17]: negation switches `in` and `out`, and leaves `undec` unchanged; a conjunction is `in` if both conjuncts are `in`, it is `out` if some conjunct is `out` and it is `undec` otherwise; disjunction is dual. Sometimes, when convenient, we will refer to a labelling v over S as a set $V = \{s \mid s \in S \textit{ and } v(s) = \mathtt{in}\} \cup \{\neg s \mid s \in S \textit{ and } v(s) = \mathtt{out}\}$. Obviously, if neither $s \in S$ nor $\neg s \in S$, then $v(s) = \mathtt{undec}$.

A three-valued labelling v is an admissible labelling of D iff for all $s \in S$, if $v(s) \neq \mathtt{undec}$, then $v(s) = v(\varphi_s)$.

Admissible labellings can also be characterized as follows:

Proposition 3.3 *Let $D = (S, L, C^{\mathtt{in}})$ be an* ADF. *Given a three-valued labelling $v : S \to \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$, let $M_v = \{s \mid v(s) = \mathtt{in}\} \cup \{\neg s \mid v(s) = \mathtt{out}\}$. Then v is an admissible labelling of D iff for any $s \in S$*

- *$s \in M_v$ iff $M_v \cap Lit(s) \in C_s^{\mathtt{in}}$;*
- *$\neg s \in M_v$ iff for all $M' \in C_s^{\mathtt{in}}$, we have $M' \not\subseteq M_v$.*

The labels `in`, `out` and `undec` are partially ordered under the information ordering $\leq_i$ or under the truth ordering $\leq_t$:

- $\mathtt{undec} \leq_i \mathtt{out}$ and $\mathtt{undec} \leq_i \mathtt{in}$, whereas `in` and `out` are incomparable. The intended meaning of $\leq_i$ is both `in` and `out` are more informative than `undec`. The pair $(\{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}, \leq_i)$ is a complete meet-semilattice, i.e., every non-empty subset of $V = \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$ has a greatest lower-bound (meet), and every directed subset $B \subseteq V$ (any two elements of B have an upper bound in B), has a least upper bound. This meet, denoted by $\sqcap_i$, is defined as follows: for every $x, y \in \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$,

$$x \sqcap_i y = \begin{cases} \mathtt{in} & \textit{if } x = y = \mathtt{in} \\ \mathtt{out} & \textit{if } x = y = \mathtt{out} \\ \mathtt{undec} & \textit{otherwise} \end{cases}$$

- $\mathtt{out} \leq_t \mathtt{undec} \leq_t \mathtt{in}$. The intended meaning of $\leq_t$ is to capture the degree of truthness. The pair $(\{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}, \leq_t)$ is a complete lattice, i.e., every non-empty subset of $V = \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$ has a greatest lower-bound (meet), denoted by $\sqcap_t$ and a least upper-bound (join), denoted by $\sqcup_t$. For every $x, y \in \{\texttt{in}, \texttt{out}, \texttt{undec}\}$, we define

$$x \sqcap_t y = \begin{cases} \texttt{in} & \textit{if } x = y = \texttt{in} \\ \texttt{out} & \textit{if } x = \texttt{out} \textit{ or } y = \texttt{out} \\ \texttt{undec} & \textit{otherwise} \end{cases} \quad x \sqcup_t y = \begin{cases} \texttt{in} & \textit{if } x = \texttt{in} \textit{ or } y = \texttt{in} \\ \texttt{out} & \textit{if } x = y = \texttt{out} \\ \texttt{undec} & \textit{otherwise} \end{cases}$$

The information ordering $\leq_i$ is extended to labellings v_1, v_2 over S: $v_1 \leq_i v_2$ iff $v_1(s) \leq_i v_2(s)$ for all $s \in S$. Let $\mathfrak{V}_S$ be the set of all three-valued labellings over S. The pair $(\mathfrak{V}_s, \leq_i)$ is a complete meet-semilattice, where the meet operation $\sqcap_i$ of this semilattice is given by $v_1 \sqcap_i v_2 = v_1(s) \sqcap_i v_2(s)$ for all $s \in S$. The valuation $v(s) = \texttt{undec}$ for every $s \in S$ is the least element of this semilattice.

Similarly, the truth ordering $\leq_t$ is extended to labellings v_1, v_2 over S: $v_1 \leq_t v_2$ iff $v_1(s) \leq_t v_2(s)$ for all $s \in S$. Let $\mathfrak{V}_S$ be the set of all three-valued labellings over S. The pair $(\mathfrak{V}_s, \leq_t)$ is a complete lattice, where the meet operation $\sqcap_t$ of this lattice is given by $v_1 \sqcap_t v_2 = v_1(s) \sqcap_t v_2(s)$ for all $s \in S$. In addition, the join operation $\sqcup_t$ of this lattice is given by $v_1 \sqcup_t v_2 = v_1(s) \sqcup_t v_2(s)$ for all $s \in S$. The valuation $v(s) = \texttt{out}$ for every $s \in S$ is the least element of this lattice whereas the valuation $v(s) = \texttt{in}$ for every $s \in S$ is its greatest element.

Besides admissible labellings, complete labellings are a crucial notion for the definition ADF semantics:

Definition 3.4 [Complete labellings] Let $D = (S, L, C^{\texttt{in}})$ be an ADF. A three-valued labelling v over S is complete iff v is a $\leq_t$-minimal labelling such that for all $s \in S$, $v(s) = v(\varphi_s)$.

This means that if v is a complete labelling of an ADF D, there is no labelling v' such that $v' <_t v$ and for all $s \in S$, $v'(s) = v'(\varphi_s)$. Such a minimization is a novelty introduced in the present work, and as we will put forward, it will guarantee an equivalence result between ADF semantics and abstract argumentation semantics.

3.3 Fixed-Point Semantics

We will employ an immediate consequences operator Γ_D over three-valued labellings to compute and define the semantics of ADFs:

Definition 3.5 [Γ_D Operator] Let $D = (S, L, C^{\varphi})$ be an ADF, and v a three-valued labelling over S. We define $\Gamma_D(v) = v'$, where $v'(s) = v(\varphi_s)$ for any $s \in S$.

We say v is a fixpoint of Γ_D if $\Gamma_D(v) = v$; it is a pre-fixpoint of Γ_D if $\Gamma_D(v) \leq_i v$; it is a post-fixpoint of Γ_D if $v \leq_i \Gamma_D(v)$. The following result suffices to guarantee each ADF has a unique $\leq_i$-least fixpoint/pre-fixpoint:

Its is important to notice that, in general, our version of Γ_D will not coincide with the corresponding operator defined in [6,4]. However, if for any acceptance condition formula ϕ (in disjunctive normal form) in an ADF, there is no occurrence of both a propositional symbol a and its negation $\neg a$, the two operators will be equivalent [1].

Proposition 3.6 *For any* ADF $D = (S, L, C^{\varphi})$*, the operator* Γ_D *is* $\leq_i$*-monotone.*

Hence, due to the fundamental result by Tarski and Knaster [27], the $\leq_i$-least pre-fixpoint and also the $\leq_i$-least fixpoint are unique (indeed, they coincide with one another) and guaranteed to exist. Besides, the set of fixpoints of Γ_D forms a complete meet-semilattice with respect to $\leq_i$:

Theorem 3.7 *Let* D *be an* ADF*. The fixpoints of* Γ_D *form a complete meet-semilattice with respect to* $\leq_i$*.*

The next results show how Γ_D operator can be employed to characterize admissible and also complete labellings:

Proposition 3.8 *Let* $D = (S, L, C^{\varphi})$ *be an* ADF*. We have*

- *A labelling* v *over* S *is an admissible labelling of* D *iff* $v \leq_i \Gamma_D(v)$*.*
- *A labelling* v *over* S *is a complete labelling of* D *iff* v *is is a* $\leq_t$*-minimal labelling such that* $v = \Gamma_D(v)$*.*

In addition, we define that a labelling v over S is a model of D iff $\Gamma_D(v) \leq_t v$. It is worthy noting our definition of model is different from that introduced in [6]. While there models and admissible labellings coincide with each other, in the present work, models in general are not admissible. Indeed, we associate admissible labellings with the post-fixpoints of Γ_D operator (with respect to information ordering $\leq_i$), whereas models are associated with the pre-fixpoints of Γ_D operator (with respect to truth ordering $\leq_t$). We emphasize anyway our defense of this characterization of models is inspired by how models are usually defined in logic programming (see [24] for more details).

When introducing some semantics for ADFs, we will resort to the set of undefined statements of a labelling v over S, denoted by $\texttt{undec}(v) = \{s \in S \mid v(s) = \texttt{undec}\}$. We can now define some of the main semantics for an ADF as follows:

Definition 3.9 Let $D = (S, L, C^{\varphi})$ be an ADF, and v a labelling over S. We have

- v is a grounded model of D iff v is a $\leq_i$-minimal complete labelling of D.
- v is a preferred model of D iff v is a $\leq_i$-maximal admissible labelling of D.
- v is a stable model of D iff v is a two-valued partial stable model of D.
- v is a semi-stable model of D iff v is a complete model of D with minimal (with respect to set inclusion) $\texttt{undec}(v)$.

We proceed by displaying an example to illustrate these semantics:

Example 3.10 Consider the ADF $D = (S, C^{\varphi})$ given by

$$a[\neg b] \qquad b[\neg a] \qquad c[\neg b \wedge e] \qquad d[\neg c] \qquad e[\neg d]$$

where $S = \{a, b, c, d, e\}$, and the acceptance formula of each statement $s \in S$ is written in square brackets on the right of s. Concerning the semantics of D, we have:

- $\{a, \neg b\}$, $\{b, d, \neg a, \neg c, \neg e\}$ and $\emptyset$ are the complete labellings of D.
- $\emptyset$ is the unique grounded model of D.
- $\{a, \neg b\}$, $\{b, d, \neg a, \neg c, \neg e\}$ are the preferred models of D.
- $\{b, d, \neg a, \neg c, \neg e\}$ is the unique stable model of D
- $\{b, d, \neg a, \neg c, \neg e\}$ is the unique semi-stable model of D.

Notice that unlike complete, grounded, preferred and semi-stable models, the stable models are not defined for all ADFs. For instance, in an ADF whose unique statement is $a[\neg a]$, there is no stable model. Furthermore, an ADF can have more than one stable model. It is the case of the ADF represented by

$$a[\neg b] \qquad b[\neg a]$$

in which $\{a, \neg b\}$ and $\{b, \neg a\}$ are the stable models of D.

We also identify the relations below involving the semantics we have defined:

Theorem 3.11 *Let D be an* ADF*.*

- *v is a preferred model of D iff v is a $\leq_i$-maximal fixpoint of Γ_D.*
- *If D has at least one stable model, then v is a stable model of D iff v is a semi-stable model of D.*
- *If v is a stable model of D, then v is a semi-stable model of D, but not vice versa.*

In the next section, we will focus on the semantic relations between ADFs and AFs.

4 From afs to adfs

In this section we will regard a subclass of ADFs which corresponds to AFs. We will start by showing how to encode a given AF into a corresponding ADF.

Definition 4.1 [4] Given an AF $F = (A, R)$, the ADF associated to F is $D_F = (A, R, C^{\varphi})$ with $C^{\varphi} = \{\varphi_a\}_{a \in A}$, where

$$\varphi_a = \bigwedge_{(b,a) \in R} \neg b$$

A distinguishing aspect of an ADF D_F associated to an AF when compared with ordinary ADFs is that the complete labellings of D_F coincide with the fixpoints of Γ_{D_F}:

Theorem 4.2 *Let $F = (A, R)$ be an* AF *and $D_F = (A, R, C^{\varphi})$ its associated* ADF*. Given a labelling v over S, if $\Gamma_{D_F}(v) = v$, then there is no labelling v' over S such that $v' <_t v$ and $\Gamma_{D_F}(v') = v'$.*

Corollary 4.3 *Let $F = (A, R)$ be an* AF *and $D_F = (A, R, C^\varphi)$ its associated* ADF*. For any labelling v over S, v is a complete labelling of D_F iff $\Gamma_{D_F}(v) = v$.*

Now we can show a fundamental result of this section: the semantics we have defined for ADFs are proper generalizations of the corresponding semantics for AFs:

Theorem 4.4 *Let F be an* AF *and D_F its associated* ADF*. A labelling is admissible, complete, preferred, grounded, stable semi-stable of F iff it is admissible, complete, preferred, grounded, stable, semi-stable of D_F.*

We thus have generalized the main semantics for AFs to ADFs by pursuing an approach based on three-valued interpretations.

5 Conclusions

In this paper, we revisited previous work on Abstract Dialectical Frameworks (ADFs) [6,4,1], offering revised definitions (Section 3) and a series of equivalence results between Abstract Argumentation semantics [15,8] and ADF semantics under AF-induced ADFs (Section 4). The main innovations of our work stem from (i) using three-valued acceptance conditions [1] for our ADFs instead of the original two-valued approach of [6,4] and (ii) requiring that the complete models of an ADF is not only a fixed-point of the consequences operator Γ_D, but also that it is $\leq_i$-minimal. Those differences allow us to characterize acceptance conditions that can evaluate arguments as undecided, an assessment that is as much of a standard in computational argumentation as the status of accepted or rejected arguments.

It is important to notice that the immediate consequences operator Γ_D used in our calculations is not the same as the one defined in [6,4], mainly because we moved from two-valued labellings of arguments to a three-valued approach. By doing so, an argument whose acceptance condition is a formula like $a \vee \neg a$ will be evaluated as `undec` whenever a is itself `undec`, while in the original formulation of [6,4], a would be accepted due to its acceptance condition being a tautology. Indeed the two operators will be equivalent if no acceptance condition ϕ (in disjunctive normal form) refers to both a propositional symbol a and its negation $\neg a$ [1]. This difference between the operators also contributes to the original results we presented in this work.

As for future work we intend to study the connections amongst ADFs semantics and logic programming semantics. In [26] the authors have tried to establish an equivalence relation between both kinds of semantics, but they only succeeded for two-valued semantics. In the same work, they showed two logic programs that are not equivalent according to a three-valued semantics would yield the same framework when translated to ADFs. Hence, their translation is not faithful with respect to three-valued semantics. We claim one of the culprits for such a negative result is the use of two-valued acceptance conditions. Considering that in [10] a variety of equivalence results between logic programming semantics and (three-valued) AF semantics were proved, we conjecture the three-valued setting for the acceptance conditions is what we need to restore the sought semantic equivalences between ADFs and logic programs.

References

[1] João Alcântara and Samy Sá. Abstract dialectical frameworks with three-valued acceptance conditions. (To appear).

[2] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[3] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes Wallner, and Stefan Woltran. Abstract dialectical frameworks. an overview. *IFCoLog Journal of Logics and Their Applications*, 4(8):2263–2317, 2017.

[4] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes Peter Wallner, and Stefan Woltran. Abstract dialectical frameworks revisited. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 803–809. AAAI Press, 2013.

[5] Gerhard Brewka, Sylwia Polberg, and Stefan Woltran. Generalizations of dung frameworks and their role in formal argumentation. *IEEE Intelligent Systems*, 29(1):30–38, 2014.

[6] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In *Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*, pages 102–111. AAAI Press, 2010.

[7] Martin Caminada. On the issue of reinstatement in argumentation. *Logics in artificial intelligence*, pages 111–123, 2006.

[8] Martin Caminada. Semi-stable semantics. *1st International Conference on Computational Models of Argument (COMMA)*, 144:121–130, 2006.

[9] Martin Caminada and Gabriella Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.

[10] Martin Caminada, Samy Sá, João Alcântara, and Wolfgang Dvořák. On the equivalence between logic programming semantics and argumentation semantics. *International Journal of Approximate Reasoning*, 58:87–111, 2015.

[11] Martin WA Caminada and Dov M Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109, 2009.

[12] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 378–389. Springer, 2005.

[13] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Bipolarity in argumentation graphs: Towards a better understanding. *International Journal of Approximate Reasoning*, 54(7):pp–876, 2013.

[14] Sylvie Coste-Marquis, Sébastien Konieczny, Pierre Marquis, and Mohand Akli Ouali. Weighted attacks in argumentation frameworks. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, page 593–597. AAAI Press,, 2012.

[15] 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.

[16] Paul E Dunne, Anthony Hunter, Peter McBurney, Simon Parsons, and Michael Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artificial Intelligence*, 175(2):457–486, 2011.

[17] Stephen Cole Kleene, NG de Bruijn, J de Groot, and Adriaan Cornelis Zaanen. *Introduction to metamathematics*, volume 483. van Nostrand New York, 1952.

[18] Diego C Martınez, Alejandro J Garcıa, and Guillermo R Simari. An abstract argumentation framework with varied-strength attacks. In *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 135–144, 2008.

[19] Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artificial intelligence*, 173(9-10):901–934, 2009.

[20] Nir Oren and Timothy J. Norman. Semantics for evidence-based argumentation. In Philippe Besnard, Sylvie Doutre, and Anthony Hunter, editors, *Proceedings of the 2nd International Conference on Computational Models of Argument (COMMA)*, volume 172, pages 276–284. IOS Press, 2008.

[21] Sylwia Polberg and Nir Oren. Revisiting support in abstract argumentation systems. In *COMMA*, pages 369–376, 2014.

[22] John L Pollock. How to reason defeasibly. *Artificial Intelligence*, 57(1):1–42, 1992.

[23] John L Pollock. *Cognitive carpentry: A blueprint for how to build a person.* Mit Press, 1995.

[24] Teodor C. Przymusinski. The well-founded semantics coincides with the three-valued stable semantics. *Fundamenta Informaticae*, 13(4):445–463, 1990.

[25] Guillermo R Simari and Ronald P Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial intelligence*, 53(2-3):125–157, 1992.

[26] Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence*, 205:39–70, 2013.

[27] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific journal of Mathematics*, 5(2):285–309, 1955.

[28] Frans H van Eemeren and Bart Verheij. Argumentation theory in formal and computational perspective. *IFColog Journal of Logics and Their Applications*, 4(8):2099–2181, 2017.

[29] Bart Verheij. Two approaches to dialectical argumentation: admissible sets and argumentation stages. *Proc. NAIC*, 96:357–368, 1996.

[30] Gerard AW Vreeswijk. *Studies in defeasible argumentation.* PhD thesis, Free University of Amsterdam, 1993.

[31] Gerard AW Vreeswijk. Abstract argumentation systems. *Artificial intelligence*, 90(1-2):225–279, 1997.

A Proofs of Theorems

A.1 Theorems and Proofs from Section 3:

Proposition 3.3 Let $D = (S, L, C^{\mathtt{in}})$ be an ADF. Given a three-valued labelling $v : S \to \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$, let $M_v = \{s \mid v(s) = \mathtt{in}\} \cup \{\neg s \mid v(s) = \mathtt{out}\}$. Then v is an admissible labelling of D iff for any $s \in S$

- $s \in M_v$ iff $M_v \cap Lit(s) \in C_s^{\mathtt{in}}$;
- $\neg s \in M_v$ iff for all $M' \in C_s^{\mathtt{in}}$, we have $M' \nsubseteq M_v$.

Proof. Let $D = (S, L, C^t)$ be an ADF and v be a model of D. Then for any $s \in S$, we have

- $s \in M_v$ iff $v(s) = \mathtt{in}$ iff $v(\varphi_s) = \mathtt{in}$ iff according to the characterization of φ_s, there exists a maximal set $M' \in C_s^{\mathtt{in}}$ such that $v(M') = \mathtt{in}$ iff there exists a maximal set $M' \in C_s^{\mathtt{in}}$ such that $M' \subseteq M_v$ and $M' \subseteq Lit(s)$ iff $M_v \cap Lit(s) \in C_s^{\mathtt{in}}$.
- $\neg s \in M_v$ iff $v(s) = \mathtt{out}$ iff $v(\varphi_s) = \mathtt{out}$ iff according to the characterization of φ_s for all $M' \in C_s^{\mathtt{in}}$, we have $v(M') = \mathtt{out}$ iff for all $M' \in C_s^{\mathtt{in}}$, we have $M' \nsubseteq M_v$ and $M' \subseteq Lit(s)$ iff $M_v \cap Lit(s) \notin C_s^{\mathtt{in}}$.

□

Proposition 3.6 For any ADF $D = (S, L, C^{\varphi})$, the operator Γ_D is $\leq_i$-monotone.

Proof. We have to prove that for any interpretations v_1 and v_2 over S if $v_1 \leq_i v_2$, then $\Gamma_D(v_1) \leq_i \Gamma_D(v_2)$.

Assume $v_1 \leq_i v_2$. This means that for each $s \in S$, $v(s) \leq_i v_s(s)$. We have three possibilities:

- $\Gamma_D(v_1)(s) = \mathtt{in}$. By definition, $v_1(\varphi_s) = \mathtt{in}$. Thus, there exists $\{a_1, \ldots, a_m, \neg b_1, \ldots, \neg b_n\} \subseteq \mathit{Lit}(s)$ such that $v_1(a_1) = \cdots = v_1(a_m) = v_1(\neg b_1) = \cdots = v_1(\neg b_n) = \mathtt{in}$. As $v_1 \leq_i v_2$, we have there exists $\{a_1, \ldots, a_m, \neg b_1, \ldots, \neg b_n\} \subseteq \mathit{Lit}(s)$ such that $v_2(a_1) = \cdots = v_2(a_m) = v_2(\neg b_1) = \cdots = v_2(\neg b_n) = \mathtt{in}$. In this case, $v_2(\varphi_s) = \Gamma_D(v_2)(s) = \mathtt{in}$.
- $\Gamma_D(v_1)(s) = \mathtt{out}$. By definition, $v_1(\varphi_s) = \mathtt{out}$. Thus, for all $\{a_1, \ldots, a_m, \neg b_1, \ldots, \neg b_n\} \subseteq \mathit{Lit}(s)$, there exists $1 \leq i \leq m$ or $1 \leq j \leq n$ such that $v_1(a_i) = \mathtt{out}$ or $v_1(\neg b_j) = \mathtt{out}$. As $v_1 \leq_i v_2$, we have for all $\{a_1, \ldots, a_m, \neg b_1, \ldots, \neg b_n\} \subseteq \mathit{Lit}(s)$, there exists $1 \leq i \leq m$ or $1 \leq j \leq n$ such that $v_2(a_i) = \mathtt{out}$ or $v_2(\neg b_j) = \mathtt{out}$. In this case, $v_2(\varphi_s) = \Gamma_D(v_2)(s) = \mathtt{out}$.
- $\Gamma_D(v_1)(s) = \mathtt{undec}$. Trivially $\Gamma_D(v_1)(s) = \mathtt{undec} \leq_i \Gamma_D(v_2)(s)$.

□

Theorem 3.7 Let D be an ADF. The fixpoints of Γ_D form a complete meet-semilattice with respect to $\leq_i$.

Proof.

Let $D = (S, L, C^\varphi)$ be an ADF and F be the set of all fixpoints of Γ_D. The set F is non-empty as the $\leq_i$-least fixpoint of Γ_D is guaranteed to exist. It is also the least element of F. Let $E \subseteq F$ be finite and non-empty. As the pair $(\mathfrak{V}_S, \leq_i)$ is a complete meet-semilattice, E has a greatest lower bound (glb) $v_E \in \mathfrak{V}_S$. We have to show that $v_E \in F$, i.e., v_E is a fixpoint of Γ_D.

Notice that the pair $(\hat{E}, \leq_i)$, where $\hat{E} = \{v \in \mathfrak{V}_S \mid v \leq_i v_E\}$ is a complete lattice. Now we will show for every $x \in \hat{E}$, $\Gamma_D(x) \in \hat{E}$:

Let w be an arbitrary element of E. We have $x \leq_i v_E$ and $v_E \leq_i w$

$\Rightarrow$ $\{$*As* Γ_D *is monotonic,* $\Gamma_D(x) \leq_i \Gamma_D(v_E) \leq_i \Gamma_D(w)\}$
$\Gamma_D(x) \leq_i \Gamma_D(w)$

$\Rightarrow$ $\{$*As* w *is a fixpoint of* $\Gamma_D\}$
$\Gamma_D(x) \leq_i w$

$\Rightarrow$ $\{w$ *is an arbitrary element of* $E\}$
$\Gamma_D(x)$ is a lower bound of E.

$\Rightarrow$ $\{v_E$ *is the greatest lower bound of* $E\}$
$\Gamma_D(x) \leq_i v_E$

$\Rightarrow$ $\left\{\textit{Definition of } \hat{E}\right\}$
$\Gamma_D(x) \in \hat{E}$

Thus, Γ_D is a mapping over the complete lattice $(\hat{E}, \leq_i)$. Therefore, it has a $\leq_i$-greatest fixpoint $\hat{v_E} \in \hat{E}$. Since v_E is the greatest element of $\hat{E}$, we have $\hat{v_E} \leq_i v_E$. Consequently, $\hat{v_E}$ is the greatest lower bound of E in F.

□

Proposition 3.8 Let $D = (S, L, C^{\varphi})$ be an ADF. We have

- A labelling v over S is an admissible labelling of D iff $v \leq_i \Gamma_D(v)$.
- A labelling v over S is a complete labelling of D iff v is is a $\leq_t$-minimal labelling such that $v = \Gamma_D(v)$.

Proof.

Firstly we will prove a valuation v over S is an admissible labelling of D iff $v \leq_i \Gamma_D(v)$:

$\Rightarrow$

Let v be an admissible labelling of D. For any $s \in S$, we have the following possibilities:

- $v(s) \in \{\mathtt{in}, \mathtt{out}\}$: as v is an admissible labelling of D, by definition, $v(s) = v(\varphi_s) = \Gamma_D(v)(s)$.
- $v(s) = \mathtt{undec}$: trivially we obtain $v(s) \leq_i \Gamma_D(v)(s)$.

$\Leftarrow$

Suppose $v \leq_i \Gamma_D(v)$. This means for every $s \in S$, we have $v(s) \leq_i \Gamma_D(v)(s)$. Then, if $v(s) \neq \mathtt{undec}$, we know $v(s) = \Gamma_D(v)(s) = v(\varphi_s)$. In other words, v is an admissible labelling of D.

Now we will prove a valuation v over S is complete iff v is a $\leq_t$-minimal labelling such that $v = \Gamma_D(v)$:

The result is immediate; according to Definitions 3.4 and 3.5, a labelling v over S is complete iff v is a $\leq_t$-minimal labelling such that for all $s \in S$, $v(s) = v(\varphi_s) = \Gamma_D(v)(s)$, i.e., $v = \Gamma_D(v)$. □

Theorem 3.11 Let D be an ADF.

- v is a preferred model of D iff v is a $\leq_i$-maximal fixpoint of Γ_D.
- If D has at least one stable model, then v is a stable model of D iff v is a semi-stable model of D.
- If v is a stable model of D, then v is a semi-stable model of D, but not vice versa.

Proof.

- *v is a preferred model of D iff v is a $\leq_i$-maximal fixpoint of* Γ_D.

  Let $A = \{v \in \mathfrak{C}_s \mid v \leq_i \Gamma_D(v)\}$ be the set of all admissible interpretations of D.

  $\Rightarrow$ If v_1 is a preferred model of D, then v_1 is a $\leq_i$-maximal element of A. Further, because $v_1 \in A$, $v_1 \leq_i \Gamma_D(v_1)$.

  $\Rightarrow$ $\{As\ \Gamma_D\ is\ \leq_i -monotonic,\}$

  $\Gamma_D(v_1) \leq_i \Gamma_D(\Gamma_D(v_1))$

  $\Rightarrow$ $\{From\ above,\ \Gamma_D(v_1) \in A;\ and\ v_1\ is\ a\ \leq_i\text{-}maximal\ \ element\ of\ A.\}$

  $v_1 \not<_i \Gamma_D(v_1)$

  $\Rightarrow$ $\{v_1\ is\ an\ admissible\ interpretation\ of\ D : v_1 \leq_i \Gamma_D(v_1)\}$

  $v_1 = \Gamma_D(v_1)$

  $\Rightarrow$ v_1 is a $\leq_i$-maximal fixpoint of Γ_D, because v_1 is a fixpoint of Γ_D, and every fixpoint of Γ_D is also a pre-fixpoint of Γ_D. As v_1 is a $\leq_i$-maximal element of

A, it is also a $\leq_i$-maximal fixpoint of Γ_D.

$\Leftarrow$

Let v_2 be a $\leq_i$-maximal fixpoint of Γ_D. Then $v_2 \in A$. By absurd, suppose v_1 is a $\leq_i$-maximal element of A and $v_2 <_i v_1$. From the result above, v_1 is a fixpoint of Γ_D. It is an absurd as we have assumed v_2 is a $\leq_i$-maximal fixpoint of Γ_D. Hence, v_2 is $\leq_i$-maximal element of A, i.e., v_2 is a preferred model of D.

- *If D has at least one stable model, then v is a stable model of D iff v is a semi-stable model of D.*

  Let D be an ADF with at least one stable model.

  $\Rightarrow$

  If v is a stable model of D, then v is a complete model of D such that $\texttt{undec}(v) = \emptyset$. Thus v is a complete model of D with minimal $\texttt{undec}(v)$, i.e., v is an semi-stable model of D.

  $\Leftarrow$

  If v is a semi-stable model of D, then v is a complete model of D with minimal $\texttt{undec}(v)$. As D has at least one stable model, there exists a complete model v' of D such that $\texttt{undec}(v') = \emptyset$. This means for any complete model v'' with minimal $\texttt{undec}(v'')$, we will have $\texttt{undec}(v'') = \emptyset$. Thus, v is a complete model of D such that $\texttt{undec}(v) = \emptyset$, i.e., v is a stable model of D.

- *If v is a stable model of D, then v is a semi-stable model of D, but not vice versa.*

  If v is a stable model of D, then v is a complete model of D such that $\texttt{undec}(v) = \emptyset$. This means for any complete model v' with minimal $\texttt{undec}(v')$, we will have $\texttt{undec}(v') = \emptyset$. Thus, v is a complete model of D with minimal $\texttt{undec}(v)$, i.e., v is a semi-stable model of D.

  As a counterexample in the opposite direction, look on the ADF D whose unique statement is $a[\neg a]$: we see $\emptyset$ is a semi-stable model of D, but it is not a stable model of D.

□

A.2 Theorems and Proofs from Section 4:

Theorem 4.2 Let $F = (A, R)$ be an AF and $D_F = (A, R, C^\varphi)$ its associated ADF. Given a labelling v over S, if $\Gamma_{D_F}(v) = v$, then there is no labelling v' over S such that $v' <_t v$ and $\Gamma_{D_F}(v') = v'$.

Proof. Let $F = (A, R)$ be an AF and $D_F = (A, R, C^\varphi)$ its associated ADF. Consider the labelling v over S and assume $\Gamma_{D_F}(v) = v$. We will prove there is no labelling v' over S such that $v' <_t v$ and $\Gamma_{D_F}(v') = v'$. By absurd, suppose there exists a labelling v' over S such that $v' <_t v$ and $\Gamma_{D_F}(v') = v'$. For each $s \in S$, we have the following possibilities:

- $v'(s) = \texttt{in}$. As $v' <_t v$, we have $v(s) = \texttt{in}$.
- $v'(s) = \texttt{out}$. As $\Gamma_{D_F}(v') = v'$, we know $v'(\varphi_s) = \texttt{out}$. This means there exists $(b, s) \in R$ such that $v'(b) = \texttt{in}$. As $v' <_t v$, we have there exists $(b, s) \in R$ such

that $v(b) = \texttt{in}$. In this case, $v(s) = \texttt{out}$.

- $v(s) = \texttt{in}$. Then for all $(b, s) \in R$, we have $v(b) = \texttt{out}$. As $v' <_t v$, we have for all $(b, s) \in R$, $v'(b) = \texttt{out}$. In this case, $v'(s) = \texttt{in}$.
- $v(s) = \texttt{out}$. As $v' <_t v$, we have $v'(s) = \texttt{out}$.

Hence $v(s) = \texttt{in}$ iff $v'(s) = \texttt{in}$ and $v(s) = \texttt{out}$ iff $v'(s) = \texttt{out}$. This implies $v(s) = \texttt{undec}$ iff $v'(s) = \texttt{undec}$. But then $v = v'$. It is an absurd as $v <_t v'$. Consequently, there is no labelling v' over S such that $v' <_t v$ and $\Gamma_{D_F}(v') = v'$. □

Corollary 4.3 Let $F = (A, R)$ be an AF and $D_F = (A, R, C^\varphi)$ its associated ADF. For any labelling v over S, v is a complete labelling of D_F iff $\Gamma_{D_F}(v) = v$.

Proof.

Let $F = (A, R)$ be an AF and $D_F = (A, R, C^\varphi)$ its associated ADF. Consider the labelling v over S.

⇒

Assume v is a complete labelling of D_F. We will prove $\Gamma_{D_F}(v) = v$:

It is straightforward as any complete labelling of D_F is in particular a fixpoint of Γ_{D_F}.

⇐

Assume $\Gamma_{D_F}(v) = v$. We will prove v is a complete labelling of D_F:

As $\Gamma_{D_F}(v) = v$, we obtain from Theorem 4.2 there is no labelling v' over S such that $v' <_t v$ and $\Gamma_{D_F}(v') = v'$. This implies v is a $\leq_t$-minimal labelling such that $v = \Gamma_{D_F}(v)$. From Proposition 3.8 we conclude v is a complete labelling of D_F.

□

Lemma A.1 *[7,11] Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be complete argument labellings of an argumentation framework* AF $= (Ar, att)$. *It holds*

(i) $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ *iff* $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$

(ii) $\texttt{in}(\mathcal{L}_1) \subsetneq \texttt{in}(\mathcal{L}_2)$ *iff* $\texttt{out}(\mathcal{L}_1) \subsetneq \texttt{out}(\mathcal{L}_2)$

(iii) $\texttt{in}(\mathcal{L}_1) = \texttt{in}(\mathcal{L}_2)$ *iff* $\texttt{out}(\mathcal{L}_1) = \texttt{out}(\mathcal{L}_2)$

Lemma A.2 *Let $F = (Ar, att)$ be an argumentation framework, and v be a complete labelling of F. it holds*

- $\texttt{in}(v)$ *is minimal (w.r.t. set-inclusion) iff v is a $\leq_i$-minimal complete labelling of F.*
- $\texttt{in}(v)$ *is maximal (w.r.t. set-inclusion) iff v is a $\leq_i$-maximal complete labelling of F.*

Proof.

- $\texttt{in}(v)$ is minimal (w.r.t. set-inclusion) iff v is a $\leq_i$-minimal complete labelling of F.

  ⇒ Let us assume $\texttt{in}(v)$ is minimal (w.r.t. set-inclusion) among all complete labellings of F. From Theorem 2.8, we know $\texttt{out}(v)$ is also minimal (w.r.t. set-inclusion) among all complete labellings of F. We will prove v is a $\leq_i$-minimal complete labelling of F:

By absurd, suppose v is not a $\leq_i$-minimal complete labelling of F. This means there exists a complete labelling v' of F such that $v' <_i v$. Thus $\texttt{in}(v') \subset \texttt{in}(v)$ or $\texttt{out}(v') \subset \texttt{out}(v)$.

If $\texttt{in}(v') \subset \texttt{in}(v)$ we have an absurd as we assumed $\texttt{in}(v)$ is minimal; otherwise, if $\texttt{out}(v') \subset \texttt{out}(v)$ we have also an absurd as $\texttt{out}(v)$ is minimal.

$\Leftarrow$

Let us assume v is a $\leq_i$-minimal complete labelling of F. We will prove $\texttt{in}(v)$ is minimal (w.r.t. set-inclusion) among all complete labelling of F:

By absurd, suppose $\texttt{in}(v)$ is not minimal (w.r.t. set-inclusion) among all complete labellings of F. This means there exists a complete labelling v' of F such that $\texttt{in}(v') \subset \texttt{in}(v)$. From Lemma A.1, we also know $\texttt{out}(v') \subset \texttt{out}(v)$. Thus $v' <_i v$. It is an absurd as we assumed v is a $\leq_i$-minimal complete labelling of F.

- $\texttt{in}(v)$ is maximal (w.r.t. set-inclusion) iff v is a $\leq_i$-maximal complete labelling of F.

$\Rightarrow$ Let us assume $\texttt{in}(v)$ is maximal (w.r.t. set-inclusion) among all complete labellings of F. From Theorem 2.8, we know $\texttt{out}(v)$ is also maximal (w.r.t. set-inclusion) among all complete labellings of F. We will prove v is a $\leq_i$-maximal complete labelling of F:

By absurd, suppose v is not a $\leq_i$-maximal complete labelling of F. This means there exists a complete labelling v' of F such that $v <_i v'$. Thus $\texttt{in}(v) \subset \texttt{in}(v')$ or $\texttt{out}(v) \subset \texttt{out}(v')$.

If $\texttt{in}(v) \subset \texttt{in}(v')$ we have an absurd as we assumed $\texttt{in}(v)$ is maximal; otherwise, if $\texttt{out}(v) \subset \texttt{out}(v')$ we have also an absurd as $\texttt{out}(v)$ is maximal.

$\Leftarrow$

Let us assume v is a $\leq_i$-maximal complete labelling of F. We will prove $\texttt{in}(v)$ is maximal (w.r.t. set-inclusion) among all complete labellings of F:

By absurd, suppose $\texttt{in}(v)$ is not maximal (w.r.t. set-inclusion) among all complete labellings of F. This means there exists a complete labelling v' of F such that $\texttt{in}(v) \subset \texttt{in}(v')$. From Lemma A.1, we also know $\texttt{out}(v) \subset \texttt{out}(v')$. Thus $v <_i v'$. It is an absurd as we assumed v is a $\leq_i$-maximal complete labelling of F.

□

Theorem 4.4 Let F be an AF and D_F its associated ADF. A labelling is admissible, complete, preferred, grounded, stable semi-stable of F iff it is admissible, complete, preferred, grounded, stable, semi-stable of D_F.

Proof.

Let $F = (A, R)$ be an AF and $D_F = (A, R, C^\varphi)$ be its associated ADF, in which

$C^{\varphi} = \{\varphi_a\}_{a \in A}$ and for every $a \in A$,

$$\varphi_a = \bigwedge_{(b,a) \in R} \neg b$$

Let v be a labelling over A. We will prove the following equivalences:

- v is an admissible labelling of F iff v is an admissible labelling of D_F.
  $\Rightarrow$

  Let v be an admissible labelling of F. Then for every argument $a \in A$,

  · If $v(a) = \mathtt{in}$, then for every $(b,a) \in R$, $v(b) = \mathtt{out}$. This means $v(\varphi_a) = \mathtt{in}$.

  · If $v(a) = \mathtt{out}$, then there exists $(b,a) \in R$, such that $v(b) = \mathtt{in}$. This means $v(\varphi_a) = \mathtt{out}$.

  Hence, $v(a) \leq_i v(\varphi_a)$, i.e., v is an admissible labelling of D_F (Proposition 3.8).

  $\Leftarrow$

  Let v be an admissible labelling of D_F. Then for every argument $a \in A$,

  · If $v(a) = \mathtt{in}$, then $v(\varphi_a) = \mathtt{in}$. This implies for every $(b,a) \in R$, $v(b) = \mathtt{out}$.

  · If $v(a) = \mathtt{out}$, then $v(\varphi_a) = \mathtt{out}$. This implies there exists $(b,a) \in R$ such that $v(b) = \mathtt{in}$.

  Hence, v is an admissible labelling of D.

- v is a complete labelling of F iff v is a complete model of D_F.

  We know v is a complete labelling of F iff for every argument $a \in A$,

  · $v(a) = \mathtt{in}$ iff for every $(b,a) \in R$, $v(b) = \mathtt{out}$ iff $v(\varphi_a) = \mathtt{in}$.

  · $v(a) = \mathtt{out}$ iff there exists $(b,a) \in R$, such that $v(b) = \mathtt{in}$ iff $v(\varphi_a) = \mathtt{out}$.

  · $v(a) = \mathtt{undec}$ iff there exists $(b,a) \in R$ such that $v(b) \neq \mathtt{out}$ and for every $(b,a) \in R$, $v(b) \neq \mathtt{in}$ iff $v(\varphi_a) = \mathtt{undec}$

  This is equivalent to say $v = \Gamma_{D_F}(v)$. From Corollary 4.3 we obtain v is a complete model of D_F.

- v is a preferred labelling of F iff v is a preferred model of D_F.

  v is a preferred labelling of F iff $\mathtt{in}(v)$ is maximal (w.r.t. set-inclusion) among all complete labellings of F iff (Lemma A.2) iff v is a $\leq_i$-maximal complete labelling of F iff v is a $\leq_i$-maximal complete labelling of D_F iff v is a $\leq_i$-maximal fixpoint of Γ_{D_F} (Corollary 4.3) iff v is a preferred model of D_F (Theorem 3.11).

- v is a grounded labelling of F iff v is a grounded model of D_F.

  v is a grounded labelling of F iff $\mathtt{in}(v)$ is the least (w.r.t. set-inclusion) among all complete labellings of F iff (Lemma A.2) v is the $\leq_i$-least complete labelling of F iff v is $\leq_i$-least complete labelling of D_F iff v is the grounded labelling of D_F.

- v is a stable labelling of F iff v is a stable model of D_F.

  v is a stable labelling of F iff v is a complete labelling of F with $\mathtt{undec}(v) = \emptyset$ iff v is a complete labelling of D_F with $\mathtt{undec}(v) = \emptyset$ iff v is a stable model of D_F.

- v is a semi-stable labelling of F iff v is a semi-stable model of D_F.

v is a semi-stable labelling of F iff v is a complete labelling of F with minimal (w.r.t. set-inclusion) $\texttt{undec}(v)$ iff v is a complete labelling of D_F with minimal (w.r.t. set-inclusion) $\texttt{undec}(v)$ iff v is a semi-stable labelling of D_F.

□