Argumentation Frameworks with Higher-Order Attacks: Labelling Semantics

Sylvie Doutre
Mickaël Lafages
Marie-Christine Lagasquie-Schiex

Université de Toulouse, IRIT
{sylvie.doutre,mickael.lafages,lagasq}@irit.fr

Tech. Report
IRIT/RR – 2020 – 01 – FR

Abstract

Recursive argumentation frameworks (RAF) take into account the notion of higher-order attacks. Their semantics are defined in terms of structures. A labelling version of these semantics is defined in this report, and their correspondence with structures is shown. The case of RAF with no recursive attacks is considered, confirming that RAF are a generalization of Dung's argumentation frameworks. A semantics which had not been considered so far for RAF, the semistable semantics, is in addition introduced.

Contents

1 Introduction **1**

2 Background **3**
2.1 Dung argumentation framework and semantics 3
2.2 Recursive argumentation framework and semantics 5

3 New semantics for RAF **9**
3.1 The Semi-stable semantics 9
3.1.1 Definition and some properties 9
3.1.2 The case of RAF with no recursive attacks 11
3.2 Reinstatement RAF labellings 13

4 RAF labellings and structure semantics **18**
4.1 Complete semantics 19
4.2 Preferred semantics 22
4.2.1 Reinstatement RAF labellings with maximal in 22
4.2.2 Reinstatement RAF labellings with maximal out 23
4.3 Stable semantics: reinstatement RAF labellings with empty und 26
4.4 Grounded semantics 27
4.4.1 Reinstatement RAF labellings with maximal und 27
4.4.2 Reinstatement RAF labellings with minimal in 28
4.4.3 Reinstatement RAF labellings with minimal out 29
4.5 Semi-stable semantics 29
4.6 A one-to-one mapping 31
4.6.1 Structures and labellings in RAF 31
4.6.2 AF labellings and RAF labellings when no recursive attack exists 31

5 Conclusion and perspectives **33**

Chapter 1

Introduction

Argumentation frameworks with higher-order attacks are a rich extension of classical Argumentation Framework (AF) (proposed by Dung in [9]): not only they consider arguments and attacks between arguments, but also attacks on attacks. Among the frameworks that consider higher-order attacks (e.g. [4, 11, 12, 2, 3]), a recent one is the Recursive Argumentation Framework (RAF) [6].

Contrarily to other higher-order attacks systems whose semantics produce sets of arguments, RAF semantics produce sets of arguments and/or attacks. Moreover, this is done "directly", *i.e.* without introducing any additional elements in the framework and without transforming the framework into another one.[1] These characteristics make RAF particularly interesting to consider as higher-order systems.

Acceptability semantics for RAF have so far been defined respectively to the notion of structure (a set of arguments along with a set of attacks). A correspondence between Dung's extension-based semantics for AF and structure-based semantics of RAF without any attack on attacks has been shown in [6], proving that RAF are a conservative generalisation of AF.

Dung's extension-based semantics for AF have also been defined in terms of labellings [5, 1]. Whereas an extension assigns to its elements an accepted or rejected status, a labelling considers a third status, undecided, which applies to arguments which are not accepted. This enrichment has proven useful for the computation of acceptance statuses in AF (see [8] for a survey).

The computation of semantics of RAF has not been addressed so far. Having

[1] See, for instance, the flattening process used in [2, 3] in order to transform higher-order frameworks into AF.

the concern of computing such semantics in the future, we adapt in this report the notion of labellings to structures. This adaptation leads us to considering an additional semantics for RAF, the semistable semantics, originally defined for AF [5].

The report is organised as follows: the basics of Dung's argumentation framework, the definition of Recursive Argumentation Frameworks and of their structure-based semantics is recalled in Chapter 2. Chapter 3 extends the semantics associated to RAFs, by introducing the semistable semantics and RAF labelling semantics. Chapter 4 shows the correspondence between structure semantics and RAF labelling semantics. Chapter 5 concludes and opens future perspectives to be studied.

Chapter 2

Background

In this section is given the necessary background on Dung's Argumentation Frameworks (AF) and then on Recursive Argumentation Frameworks (RAF).

2.1 Dung argumentation framework and semantics

In [9], Dung introduced a framework to represent argumentation in an abstract way.

Definition 1 (Dung's abstract argumentation framework [9]) *A Dung's abstract Argumentation Framework (AF for short) is a pair $\Gamma = \langle A, K \rangle$ where A is a set of arguments and $K \subseteq A \times A$ is a relation representing attacks over arguments.*

Definition 2 (Defeat and acceptability in Dung's framework) *Let $\Gamma = \langle A, K \rangle$ be an AF and $S \subseteq A$ be a set of arguments. An argument $a \in A$ is said to be:*

- *defeated w.r.t. S iff $\exists b \in S$ s.t. $(b, a) \in K$.*
- *accepted w.r.t. S iff $\forall (b, a) \in K, \exists c \in S$ s.t. $(c, b) \in K$.*

We define the sets of defeated and accepted arguments w.r.t. S as follows:

$$Def(S) = \{a \in A | \exists b \in S \text{ s.t. } (b, a) \in K\}$$

$$Acc(S) = \{a \in A | \forall (b, a) \in K, \exists c \in S \text{ s.t. } (c, b) \in K\}$$

Several "semantics" defining sets of arguments (so called "extensions") solving the argumentation have been defined on Dung's framework. Here are some of them.

Definition 3 (Semantics of Dung's AF) *Let $\Gamma = \langle A, K \rangle$ be an AF and $S \subseteq A$ be a set of arguments. S is said to be an extension:*

1. Conflict-free *iff* $S \cap Def(S) = \varnothing$.
2. Naive *iff it is a* $\subseteq$*-maximal* conflict-free *extension.*
3. Admissible *iff it is* conflict-free *and* $S \subseteq Acc(S)$.
4. Complete *iff it is* conflict-free *and* $S = Acc(S)$.
5. Preferred *iff it is a* $\subseteq$*-maximal* admissible *extension.*
6. Grounded *iff it is a* $\subseteq$*-minimal* complete *extension.*
7. Semi-stable *iff it is a* complete *extension such that* $S \cup Def(S)$ *is maximal w.r.t.* $\subseteq$.
8. Stable *iff it is* conflict-free *and* $S \cup Def(S) = A$.

Acceptability semantics can be defined in terms of labellings [5, 1].

Definition 4 (Labelling) *Let $\Gamma = \langle A, K \rangle$ be an AF, and $S \subseteq A$. A* labelling *of S is a total function $\ell : S \rightarrow \{\mathtt{in}, \mathtt{out}, \mathtt{und}\}$.*

The set of all labellings of S is denoted as $\mathscr{L}(S)$. A labelling of Γ is a labelling of A.

The set of all labellings of Γ is denoted as $\mathscr{L}(\Gamma)$.

We write $\mathtt{in}(\ell)$ for $\{a | \ell(a) = \mathtt{in}\}$, $\mathtt{out}(\ell)$ for $\{a | \ell(a) = \mathtt{out}\}$ and $\mathtt{und}(\ell)$ for $\{a | \ell(a) = \mathtt{und}\}$.

Definition 5 (Legally labelled arguments, valid labelling)

- *An* `in`*-labelled argument is said to be* legally `in` *iff all its attackers are labelled* `out`*.*
- *An* `out`*-labelled argument is said to be* legally `out` *iff at least one of its attackers is labelled* `in`*.*
- *An* `und`*-labelled argument is said to be* legally `und` *iff it does not have any attacker that is labelled* `in`*, and one of its attackers is not labelled* `out`*.*

A valid labelling *is a labelling in which all arguments are legally labelled.*

Let $\Gamma = \langle A, K \rangle$ be an AF, and $\ell \in \mathscr{L}(\Gamma)$ be a labelling. Different kinds of labelling can be defined [5, 1]:

Definition 6

- ℓ *is an* admissible *labelling of* Γ *iff for any argument* $a \in A$ *such that* $\ell(a) = \mathtt{in}$ *or* $\ell(a) = \mathtt{out}$*, a is legally labelled.*

- ℓ *is a* complete *labelling of* Γ *iff for any argument* $a \in A$*, a is legally labelled.*

- ℓ *is the* grounded *labelling of* Γ *iff it is the complete labelling of* Γ *that minimizes (w.r.t* $\subseteq$*) the set of* $\mathtt{in}$*-labelled arguments.*

- ℓ *is a* preferred *labelling of* Γ *iff it is a complete labelling of* Γ *that maximizes (w.r.t* $\subseteq$*) the set of* $\mathtt{in}$*-labelled arguments.*

- ℓ *is a* stable *labelling of* Γ *iff it is a complete labelling of* Γ *which has no* $\mathtt{und}$*-labelled argument.*

- ℓ *is an* semi-stable *labelling of* Γ *iff it is a complete labelling of* Γ *that minimizes (w.r.t* $\subseteq$*) the set of* $\mathtt{und}$*-labelled arguments.*

It has been shown in [5], that there exists a one-to-one mapping between extensions and labellings for those semantics.

2.2 Recursive argumentation framework and semantics

To the best of our knowledge, the first work where the idea of higher-order interactions appears is [4]. Then many different works followed. For instance, in [11, 12], second-order attacks are used in order to explicitly represent the impact of the preferences between arguments in the argumentation framework. Then [2, 3] introduce Argumentation Frameworks with Recursive Attacks (AFRA) that take into account the attacks on attacks and propose some semantics. A more recent variant of AFRA is given in [6] with other semantics and called Recursive Argumentation Framework (RAF). One keypoint of the RAF approach is the fact that semantics proposed in [6] produce sets of arguments and/or attacks and not only sets of arguments (as it is done for instance in [2, 3] and more recently in [10]). RAF semantics need neither the introduction of additional elements in the framework, nor its transformation into another framework.[1]

Definition 7 (Recursive argumentation framework [6]) *A recursive argumentation framework $\Gamma = \langle A, K, s, t \rangle$ is a quadruple where A and K are (possibly infinite) disjoint sets respectively representing arguments and attack names, and where $s : K \to A$ and $t : K \to A \cup K$ are functions respectively mapping each attack to its source and its target.*

As in Dung's framework, a RAF can be graphically represented as a directed graph.

Example 1 *Figure 2.1 shows an example of RAF. In all this document, arguments will be represented by a round box, while attacks will be represented by a square box.*

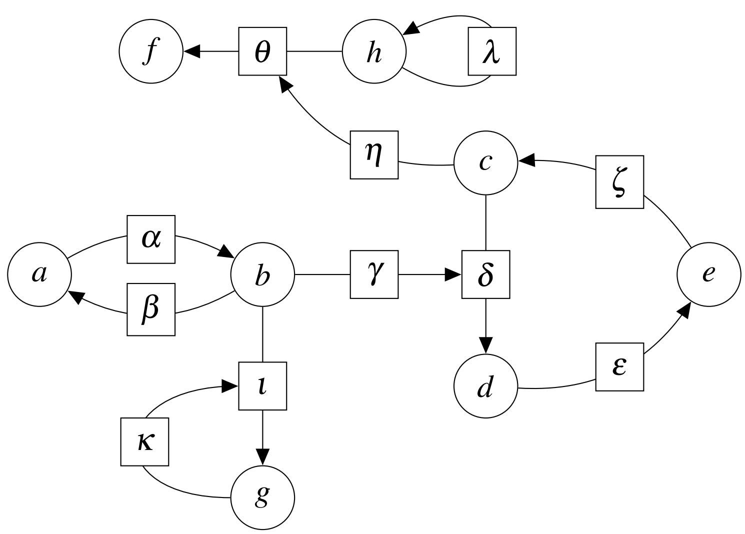

Figure 2.1: Example of a RAF

[1]For instance, one approach developed in [2, 3] uses a "flattening process" that transforms an AFRA into a Dung AF and then uses AF semantics.

Definition 8 (Structure [6]) *A pair* $\mathcal{U} = \langle S, Q \rangle$ *is said to be a structure of some* $\Gamma = \langle A, K, s, t \rangle$ *if it satisfies:* $S \subseteq A$ *and* $Q \subseteq K$*.*

Notice that by $x \in \mathcal{U}$ *we mean* $x \in S \cup Q$*.*

Definition 9 (Defeat and Inhibition [6]) *Let* $\mathcal{U} = \langle S, Q \rangle$ *be a structure.*

We denote by $Def(\mathcal{U})$ *the set of all arguments defeated by* $\mathcal{U}$*, defined by:*

$$Def(\mathcal{U}) = \{a \in A | \exists \alpha \in Q \text{ s.t. } s(\alpha) \in S \text{ and } t(\alpha) = a\}$$

We denote by $Inh(\mathcal{U})$ *the set of all attacks inhibited by* $\mathcal{U}$*, defined by:*

$$Inh(\mathcal{U}) = \{\alpha \in K | \exists \beta \in Q \text{ s.t. } s(\beta) \in S \text{ and } t(\beta) = \alpha\}$$

Definition 10 (Acceptability [6]) *An element* $x \in (A \cup K)$ *is said to be acceptable w.r.t. some structure* $\mathcal{U}$ *iff every attack* $\alpha \in K$ *with* $t(\alpha) = x$ *satisfies one of the following conditions: (i)* $s(\alpha) \in Def(\mathcal{U})$ *or (ii)* $\alpha \in Inh(\mathcal{U})$*.*

By $Acc(\mathcal{U})$ *we denote the set containing all acceptable arguments and attacks with respect to* $\mathcal{U}$*.*

For any pair of structures $\mathcal{U} = \langle S, Q \rangle$ and $\mathcal{U}' = \langle S', Q' \rangle$, we write $\mathcal{U}' \sqsubseteq \mathcal{U}'$ *iff* $(S \cup Q) \subseteq (S' \cup Q')$ and we write $\mathcal{U} \sqsubseteq_{ar} \mathcal{U}'$ *iff* $S \subseteq S'$. As usual, we say that a structure $\mathcal{U}$ is $\sqsubseteq$-maximal (resp. $\sqsubseteq_{ar}$-maximal) *iff* every $\mathcal{U}'$ that satisfies $\mathcal{U} \sqsubseteq \mathcal{U}'$ (resp. $\mathcal{U} \sqsubseteq_{ar} \mathcal{U}'$) also satisfies $\mathcal{U}' \sqsubseteq \mathcal{U}$ (resp. $\mathcal{U}' \sqsubseteq_{ar} \mathcal{U}$).

Definition 11 (Structure semantics [6, 7]) *Let* $\mathcal{U} = \langle S, Q \rangle$ *be a structure over some RAF* $\Gamma = \langle A, K, s, t \rangle$*.* $\mathcal{U}$ *is said to be:*

1. Conflict-free *iff* $S \cap Def(\mathcal{U}) = \varnothing$ *and* $Q \cap Inh(\mathcal{U}) = \varnothing$*.*
2. Admissible *iff it is conflict free and* $(S \cup Q) \subseteq Acc(\mathcal{U})$*.*
3. Complete *iff it is conflict free and* $(S \cup Q) = Acc(\mathcal{U})$*.*
4. Preferred *iff it is a* $\sqsubseteq$*-maximal* admissible *structure.*
5. Grounded *iff it is a* $\sqsubseteq$*-minimal* complete *structure.*
6. Arg-preferred *iff it is a* $\sqsubseteq_{ar}$*-maximal* admissible *structure.*
7. Stable *iff* $S = A \setminus Def(\mathcal{U})$ *and* $Q = K \setminus Inh(\mathcal{U})$*.*

In [6], Proposition 1 and 2 and Theorem 1 have been proven.

Proposition 1 ([6]) *There is always a unique grounded structure.*

Proposition 2 ([6]) *The set of all* admissible *structures forms a complete partial order with respect to* $\sqsubseteq$. *Furthermore, for every* admissible *structure* $\mathfrak{U}$, *there exists an (arg-) preferred extension* $\mathfrak{U}'$ *such that* $\mathfrak{U} \sqsubseteq \mathfrak{U}'$.

Theorem 1 ([6]) *The following assertions hold:*

- *every* complete *structure is also* admissible *,*
- *every* preferred *structure is also* complete *,*
- *every* stable *structure is also* preferred *.*

Moreover, in [6], it is proven that RAF are a conservative generalization of AF since there is a one-to-one correspondence between the structures of a RAF without recursive attacks and their corresponding Dung's extensions (the proof is given for the *complete*, *grounded*, *preferred* and *stable* semantics).

Chapter 3

New semantics for RAF

In this section is introduced the *semi-stable* semantics for RAF, and a focus on RAFs with no recursive attacks is done. The notion of AF reinstatement labelling introduced in [5] is also generalized for RAF (so called "reinstatement RAF labelling"). The same work can be done in the case of RAF.

3.1 The Semi-stable semantics

While introducing labellings for AF and studying labellings over some constraints, Caminada et al. ([5]) highlighted a non yet discovered semantics: the *semi-stable* semantics. From the *semi-stable* labelling semantics has been defined the *semi-stable* extension semantics.

3.1.1 Definition and some properties

As for AF, the *semi-stable* structures are the ones that decide the most on the acceptance or the rejection of arguments and attacks.

Definition 12 *(Semi-stable structure). Let* $\Gamma = \langle A, K, s, t \rangle$ *be a RAF and* $\mathfrak{U} = \langle S, Q \rangle$ *be some structure over it.* $\mathfrak{U}$ *is said to be an* semi-stable *structure iff* $\mathfrak{U}$ *is a* complete *structure such that:*

$$S \cup Q \cup Def(\mathfrak{U}) \cup Inh(\mathfrak{U}) \text{ is maximal w.r.t. to inclusion.}$$

Theorem 2 *The following assertions hold:*

1. *Every* stable *structure is an* semi-stable *structure*

2. *Every* semi-stable *structure is a* preferred *structure*

PROOF.

1. *(*Stable *structures are* semi-stable *ones). Let* $\Gamma = \langle A, K, s, t \rangle$ *be a RAF and* $\mathcal{U} = \langle S, Q \rangle$ *be a* stable *structure. According to the definition of a* stable *structure (Definition 11), we have:*

   $$S = A \setminus Def(\mathcal{U}) \text{ and } Q = K \setminus Inh(\mathcal{U})$$

   For any $x \in (A \cup K)$*, x is whether in* $\mathcal{U}$ *or is defeated or inhibited by* $\mathcal{U}$*. As a consequence,* $(S \cup Q \cup Def(\mathcal{U}) \cup Inh(\mathcal{U}))$ *is maximal w.r.t. to inclusion.*

   We prove so that every stable *structure is an semistable structure.*

2. *(*Semi-stable *structures are* preferred *ones). Let* $\Gamma = \langle A, K, s, t \rangle$ *be a RAF and* $\mathcal{U} = \langle S, Q \rangle$ *be an* semi-stable *structure. Let suppose that* $\mathcal{U}$ *is not a* preferred *structure.* $\mathcal{U}$ *being by definition a* complete *structure (Definition 12), there exists thus a* preferred *structure* $\mathcal{U}' = \langle S', Q' \rangle$ *such that* $\mathcal{U} \sqsubset \mathcal{U}'$*. We have thus by definition of* $\sqsubseteq$*-inclusion:*

   $$S \subseteq S' \text{ and } Q \subseteq Q' \tag{3.1}$$

   From the strict inclusion, we also have:

   $$(S \cup Q) \subset (S' \cup Q') \tag{3.2}$$

   It follows, from Equations 3.1, 3.2 and from Definition 9 that:

   $$(Def(\mathcal{U}) \cup Inh(\mathcal{U})) \subset (Def(\mathcal{U}') \cup Inh(\mathcal{U}')) \tag{3.3}$$

   Combining Equations 3.2 and 3.3, we have:

   $$(S \cup Q \cup Def(\mathcal{U}) \cup Inh(\mathcal{U})) \subset (S' \cup Q' \cup Def(\mathcal{U}') \cup Inh(\mathcal{U}')) \tag{3.4}$$

   Given that $\mathcal{U}'$ *is also a* complete *structure, the consequence of Equation 3.4 is that* $\mathcal{U}$ *is not an* semi-stable *structure, as* $(S \cup Q \cup Def(\mathcal{U}) \cup Inh(\mathcal{U}))$ *is not maximal. There is thus a contradiction.*

   We prove so that every semi-stable *structure is also a* preferred *structure.*

■

Theorem 3 *Let* $\Gamma = \langle A, K, s, t \rangle$ *be a RAF. If there exists a* stable *structure, then the* semi-stable *structures coincide with the* stable *structures.*

Proof. *Let suppose that there exists a* stable *structure* $\mathcal{U} = \langle S, Q \rangle$. *Following the definition of* stable *structures (Definition 11), we have:* $S = A \setminus Def(\mathcal{U})$ *and* $Q = K \setminus Inh(\mathcal{U})$. *As a consequence, we have* $(S \cup Def(\mathcal{U}) \cup Q \cup Inh(\mathcal{U}))$ *including all the arguments and attacks of* Γ.

According to Theorem 2, $\mathcal{U}$ *is also an* semi-stable *structure. As any* semi-stable *structure* $\mathcal{U}' = \langle S', Q' \rangle$ *maximizes the set* $(S' \cup Def(\mathcal{U}') \cup Q' \cup Inh(\mathcal{U}'))$ *and as there exists* $\mathcal{U}$, *a structure such that* $(S \cup Def(\mathcal{U}) \cup Q \cup Inh(\mathcal{U}))$ *is maximized to point that it includes all the arguments and attacks of* Γ, *then for* $\mathcal{U}'$ *to be maximal we necessarily have* $(S' \cup Def(\mathcal{U}') \cup Q' \cup Inh(\mathcal{U}'))$ *also including all the arguments and attacks of* Γ. $\mathcal{U}'$ *is then a* stable *structure.*

We prove thus that if there exists a stable *structure, then the* semi-stable *structures coincide with the* stable *structures.* ■

The *complete*, *grounded*, *preferred*, *semistable*, *argpref* and *stable* semantics corresponding to Example 1 are given in Table 3.1.

We can observe that the *stable* semantics produces no structure for that RAF. This example shows that an *semi-stable* structure is not always a *stable* one (see $\mathcal{U}_3$ and $\mathcal{U}_4$) and that a *preferred* structure is not always an *semi-stable* one (see $\mathcal{U}_2$).

3.1.2 The case of RAF with no recursive attacks

As stated in Section 2.2, it has been proven in [6] that in RAFs without recursive attacks there is a one-to-one correspondence between structures and Dung's extensions for the *complete*, *grounded*, *preferred* and *stable* semantics. Let now consider the case of the *semi-stable* semantics.

The set of *semi-stable* extensions coincides with the set of *semi-stable* structures on RAF with no recursive attacks (*i.e.* RAF that happened to be simple AF). Notice that all the structures of such a RAF contain all the attacks.

Proposition 3 (*Semi-stable* extensions and structures) *Let $\Gamma = \langle A, K, s, t\rangle$ be a RAF such that $\forall \alpha \in K$, $t(\alpha) \in A$. Γ can thus be considered as a simple AF. Let $\Gamma' = \langle A, K\rangle$ be the AF version of Γ.*

$\mathcal{U} = \langle S, K\rangle$ *is an* semi-stable *structure of* Γ *iff* S *is an* semi-stable *extension of* Γ'

Proof. *Let $\Gamma = \langle A, K, s, t\rangle$ be a RAF such that $\forall \alpha \in K$, $t(\alpha) \in A$. Γ can thus be considered as a simple AF. Let $\Gamma' = \langle A, K\rangle$ be the AF version of Γ.*

Step 1: *Let prove in a first place that if $\mathcal{U} = \langle S, K\rangle$ is an* semi-stable *structure of Γ then S is an* semi-stable *extension of Γ'.*

Let $\mathcal{U} = \langle S, K\rangle$ be an semi-stable *structure over Γ. Notice that the set of attacks of $\mathcal{U}$ is K as attacks are always valid in Γ, and so that $Inh(\mathcal{U}) = \varnothing$. Let suppose that S is not an* semi-stable *extension of Γ'. There exists thus an extension S' of Γ' such that:*

$$(S \cup Def(S)) \subset (S' \cup Def(S')) \tag{3.5}$$

We have thus :

$$(S \cup Def(S) \cup K) \subset (S' \cup Def(S') \cup K) \tag{3.6}$$

Let $\mathcal{U}' = \langle S', K\rangle$ be the structure over Γ whose set of arguments is the extension S'. For the same reason as $\mathcal{U}$, the set of attacks of $\mathcal{U}'$ is K and $Inh(\mathcal{U}') = \varnothing$.

As $Inh(\mathcal{U}) = \varnothing$ and $Inh(\mathcal{U}') = \varnothing$, we can thus say from Equation 3.6 that:

$$(S \cup Def(S) \cup K \cup Inh(\mathcal{U})) \subset (S' \cup Def(S') \cup K \cup Inh(\mathcal{U}')) \tag{3.7}$$

Given that all attacks are valid in Γ, we have: $Def(S) = Def(\mathcal{U})$ and $Def(S') = Def(\mathcal{U}')$. We have thus from Equation 3.7:

$$(S \cup Def(\mathcal{U}) \cup K \cup Inh(\mathcal{U})) \subset (S' \cup Def(\mathcal{U}') \cup K \cup Inh(\mathcal{U}')) \tag{3.8}$$

As stated by Equation 3.8, $(S \cup Def(S) \cup K \cup Inh(\mathcal{U}))$ is not maximal. It follows that $\mathcal{U}$ is not an semi-stable *structure, which is a contradiction.*

We prove so that if $\mathcal{U} = \langle S, K\rangle$ is an semi-stable *structure of Γ then S is an* semi-stable *extension of Γ'.*

Step 2: *Let now prove that if S is an* semi-stable *extension of Γ' then $\mathcal{U} = \langle S, K\rangle$ is an* semi-stable *structure of Γ.*

Let S be an semi-stable *extension of* Γ' *and let* $\mathcal{U} = \langle S, K\rangle$ *be a structure over* Γ *whose set of arguments is S. Notice that the set of attacks of* $\mathcal{U}$ *is K as attacks are always valid in* Γ.

Let suppose that $\mathcal{U}$ *is not an* semi-stable *structure. There exists thus an* semi-stable *structure* $\mathcal{U}' = \langle S', K\rangle$ *such that:*

$$(S \cup Def(\mathcal{U}) \cup K \cup Inh(\mathcal{U})) \subset (S' \cup Def(\mathcal{U}') \cup K \cup Inh(\mathcal{U}')) \tag{3.9}$$

Given that all attacks are valid in Γ*, we have:* $Inh(\mathcal{U}) = \varnothing$ *and* $Inh(\mathcal{U}') = \varnothing$*. We have thus from Equation 3.9:*

$$(S \cup Def(\mathcal{U})) \subset (S' \cup Def(\mathcal{U}')) \tag{3.10}$$

Furthermore, as all attacks are valid in Γ*, we have:* $Def(S) = Def(\mathcal{U})$ *and* $Def(S') = Def(\mathcal{U}')$*. We have thus from Equation 3.10:*

$$(S \cup Def(S)) \subset (S' \cup Def(S')) \tag{3.11}$$

As stated by Equation 3.11, $(S \cup Def(S))$ *is not maximal. It follows that S is not an* semi-stable *extension, which is a contradiction.*

We prove so that if S is an semi-stable *extension of* Γ' *then* $\mathcal{U} = \langle S, K\rangle$ *is an* semi-stable *structure of* Γ.

With steps 1 and 2, we have thus proven that:

$\mathcal{U} = \langle S, K\rangle$ *is an* semi-stable *structure of* Γ *iff S is an* semi-stable *extension of* Γ'

■

3.2 Reinstatement RAF labellings

Now that relations between structure semantics and between structure and extensions semantics has been stated, we introduce the notion of labelling on RAF.

The reason why we are interested in the labelling approach to compute semantics is that labellings are more precise than structures (as there are three statuses to describe the acceptance of elements) and especially because it seems to be more practical for finding algorithms.

Definition 13 *(RAF labelling). Let* $\Gamma = \langle A, K, s, t \rangle$ *be a recursive argumentation framework. A RAF labelling is a tuple* $\mathcal{L} = \langle \ell_A, \ell_K \rangle$ *such that* ℓ_A *is a total function* $\ell_A : A \to \{\mathtt{in}, \mathtt{out}, \mathtt{und}\}$ *and* ℓ_K*, a total function* $\ell_K : K \to \{\mathtt{in}, \mathtt{out}, \mathtt{und}\}$*.*

We define:

- $\mathtt{in}(\mathcal{L})$ *as the tuple* $\langle \{a \in A | \ell_A(a) = \mathtt{in}\}, \{\alpha \in K | \ell_K(\alpha) = \mathtt{in}\} \rangle$*,*
- $\mathtt{und}(\mathcal{L})$ *as the tuple* $\langle \{a \in A | \ell_A(a) = \mathtt{und}\}, \{\alpha \in K | \ell_K(\alpha) = \mathtt{und}\} \rangle$ *and*
- $\mathtt{out}(\mathcal{L})$ *as the tuple* $\langle \{a \in A | \ell_A(a) = \mathtt{out}\}, \{\alpha \in K | \ell_K(\alpha) = \mathtt{out}\} \rangle$*.*

Let $x \in (A \cup K)$*. Given a certain* $\mathcal{L}$*, we use the notation* $\mathcal{L}(x)$ *to indicate the labelling of x in* $\mathcal{L}$*. It could mean* $\ell_A(x)$ *or* $\ell_K(x)$*, following the nature of x. We also use the notation* $\mathtt{in}(\mathcal{L})$ *(respectively* $\mathtt{out}(\mathcal{L})$*,* $\mathtt{und}(\mathcal{L})$*) to represent the set of all* $\mathtt{in}$*-labelled (respectively* $\mathtt{out}$*-labelled,* $\mathtt{und}$*-labelled) attacks or argument in* $\mathcal{L}$*.*

Definition 14 *(Reinstatement RAF labelling). Let* $\Gamma = \langle A, K, s, t \rangle$ *be a recursive argumentation framework and* $\mathcal{L} = \langle \ell_A, \ell_K \rangle$ *be a RAF labelling.* $\mathcal{L}$ *is a reinstatement RAF labelling iff it satisfies the following conditions:* $\forall x \in (A \cup K)$*,*

- $(\mathcal{L}(x) = \mathtt{out}) \iff (\exists \alpha \in K \text{ s.t. } t(\alpha) = x, \ell_K(\alpha) = \mathtt{in} \text{ and } \ell_A(s(\alpha)) = \mathtt{in})$
- $(\mathcal{L}(x) = \mathtt{in}) \iff (\forall \alpha \in K \text{ s.t. } t(\alpha) = x, \ell_K(\alpha) = \mathtt{out} \text{ or } \ell_A(s(\alpha)) = \mathtt{out})$

Moreover, as for AF, we also introduce a notion of valid labelling. An $\mathtt{in}$-labelled element is said to be *legally* $\mathtt{in}$ *iff* all its attackers or their involved attacks are labelled $\mathtt{out}$. An $\mathtt{out}$-labelled element is said to be *legally* $\mathtt{out}$ *iff* at least one of its attackers and the involved attack are labelled $\mathtt{in}$. An $\mathtt{und}$-labelled element is said to be *legally* $\mathtt{und}$ *iff* it does not have any attacker and its involved attack that are labelled $\mathtt{in}$ and one of its attackers and the involved attack are not labelled $\mathtt{out}$. Formally, valid labellings are defined as follows.

Definition 15 (Legally labelled elements, valid RAF labelling)
Let $\Gamma = \langle A, K, s, t \rangle$ *be a recursive argumentation framework and* $\mathcal{L} = \langle \ell_A, \ell_K \rangle$ *be a RAF labelling over* Γ*. Let x be an argument or an attack of* Γ*. x is said to be* legally labelled *in* $\mathcal{L}$ *if and only if the 3 following conditions hold:*

- $x \in \mathtt{in}(\mathcal{L})$ *iff* $(\forall \alpha \in K \text{ s.t. } t(\alpha) = x, \ell_K(\alpha) = \mathtt{out} \text{ or } \ell_A(s(\alpha)) = \mathtt{out})$
- $x \in \mathtt{out}(\mathcal{L})$ *iff* $(\exists \alpha \in K \text{ s.t. } t(\alpha) = x, \ell_K(\alpha) = \mathtt{in} \text{ and } \ell_A(s(\alpha)) = \mathtt{in})$

- $x \in \mathtt{und}(\mathcal{L})$ *iff* $((\nexists \alpha \in K$ *s.t.* $t(\alpha) = x$, $\ell_K(\alpha) = \mathtt{in}$ *and* $\ell_A(s(\alpha)) = \mathtt{in})$ *and* $(\exists \alpha \in K$ *s.t.* $t(\alpha) = x$, $\ell_K(\alpha) \neq \mathtt{out}$ *and* $\ell_A(s(\alpha)) \neq \mathtt{out}))$

$\mathcal{L}$ *is said to be a* valid RAF labelling *if all its elements are* legally labelled.

Notice that by definition reinstatement RAF labellings are valid ones.
The labelling version of Table 3.1 about Example 1 is shown in Table 3.2.

		$\mathcal{U}_1$	$\mathcal{U}_2$	$\mathcal{U}_3$	$\mathcal{U}_4$
Arguments or attacks	a		✓		
	b			✓	✓
	c			✓	✓
	d			✓	✓
	e				
	f			✓	✓
	g		✓	✓	
	h				
	α	✓	✓	✓	✓
	β	✓	✓	✓	✓
	γ	✓	✓	✓	✓
	δ		✓		
	ε	✓	✓	✓	✓
	ζ	✓	✓	✓	✓
	η	✓	✓	✓	✓
	θ				
	ι				✓
	κ	✓	✓	✓	✓
	λ	✓	✓	✓	✓
Semantics with structures	*grounded*	✓			
	complete	✓	✓	✓	✓
	preferred		✓	✓	✓
	arg-preferred		✓	✓	
	semi-stable			✓	✓
	stable				

In the first part of the table, $i \checkmark j$ means that the element i belongs to the structure j.
In the second part of the table, $i \checkmark j$ means that j is a structure of the semantics i.

Table 3.1: Semantics structures

		$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$	$\mathcal{L}_4$
Arguments or attacks	a	und	in	out	out
	b	und	out	in	in
	c	und	und	in	in
	d	und	und	in	in
	e	und	und	out	out
	f	und	und	in	in
	g	und	in	in	out
	h	und	und	und	und
	α	in	in	in	in
	β	in	in	in	in
	γ	in	in	in	in
	δ	und	in	out	out
	ε	in	in	in	in
	ζ	in	in	in	in
	η	in	in	in	in
	θ	und	und	out	out
	ι	und	out	out	in
	κ	in	in	in	in
	λ	in	in	in	in
Semantics with RAF labellings	*grounded*	✓			
	complete	✓	✓	✓	✓
	preferred		✓	✓	✓
	arg-preferred		✓	✓	
	semi-stable			✓	✓
	stable				

i ✓ *j* means that *j* is a labelling of semantics *i*.

Table 3.2: Semantics RAF labellings

Chapter 4

RAF labellings and structure semantics

In this section we show that there exists a one-to-one mapping between RAF labellings and structures. Specific semantics structures happen to be coinciding with RAF labellings under some constraints.

Definition 16 *(*`Struct2Lab` *and* `Lab2Struct`*). Let $\Gamma = \langle A, K, s, t \rangle$ be a RAF, $\mathcal{U} = \langle S, Q \rangle$ be a structure and $\mathcal{L} = \langle \ell_A, \ell_K \rangle$ be a RAF labelling. The functions* $\texttt{Struct2Lab}_\Gamma$ *and* $\texttt{Lab2Struct}_\Gamma$ *are defined as following:*

- $\texttt{Struct2Lab}_\Gamma(\mathcal{U}) = \langle \ell_A, \ell_K \rangle$*, a RAF labelling with:*
  - $\ell_A = \{(a, \mathit{in}) | a \in S\} \cup \{(a, \mathit{out}) | a \in (A \setminus S) \text{ and } a \in \mathit{Def}(\mathcal{U})\} \cup \{(a, \mathit{und}) | a \in (A \setminus S) \text{ and } a \notin \mathit{Def}(\mathcal{U})\}$
  - $\ell_K = \{(\alpha, \mathit{in}) | \alpha \in Q\} \cup \{(\alpha, \mathit{out}) | \alpha \in (K \setminus Q) \text{ and } \alpha \in \mathit{Inh}(\mathcal{U})\} \cup \{(\alpha, \mathit{und}) |$ $\alpha \in (K \setminus Q) \text{ and } \alpha \notin \mathit{Inh}(\mathcal{U})\}$
- $\texttt{Lab2Struct}_\Gamma(\mathcal{L}) = \langle S, Q \rangle$*, a structure with:*
  - $S = \{a | \ell_A(a) = \mathit{in}\}$
  - $Q = \{\alpha | \ell_K(\alpha) = \mathit{in}\}$

We write `Struct2Lab` *and* `Lab2Struct` *instead of* $\texttt{Struct2Lab}_\Gamma$ *and* $\texttt{Lab2Struct}_\Gamma$ *when there is no ambiguity about the RAF Γ we refer to.*

4.1 Complete semantics

Reinstatement RAF labellings coincide with *complete* structures as stated by Theorems 4 and 5.

Theorem 4 *Let $\Gamma = \langle A, K, s, t\rangle$ be a RAF and let $\mathcal{L} = \langle \ell_A, \ell_K\rangle$ be a reinstatement RAF labelling. Then* `Lab2Struct`$(\mathcal{L})$ *is a* complete *structure.*

PROOF. *Let $\mathcal{U} =$* `Lab2Struct`$(\mathcal{L})$*. According to Definition 11, $\mathcal{U}$ being a* complete *structure (with $\mathcal{U} = \langle S, Q\rangle$) means that $(S \cup Q) = Acc(\mathcal{U})$. In a first step, let us prove that $(S \cup Q) \subseteq Acc(\mathcal{U})$ and then that $(S \cup Q) \supseteq Acc(\mathcal{U})$.*

Step 1: $(S \cup Q) \subseteq Acc(\mathcal{U})$

Let $x \in (S \cup Q)$. By definition of `Lab2Struct`$(\mathcal{L})$*, we have $\mathcal{L}(x) = \mathtt{in}$. Given that $\mathcal{L}$ is a reinstatement RAF labelling, we have:*

$$\forall \alpha \in K \text{ s.t. } t(\alpha) = x,\ \ell_K(\alpha) = \mathtt{out} \text{ or } \ell_A(s(\alpha)) = \mathtt{out} \tag{4.1}$$

So two cases must be considered: $\ell_K(\alpha) = \mathtt{out}$ or $\ell_A(s(\alpha)) = \mathtt{out}$.

1. $\ell_K(\alpha) = \mathtt{out}$.

   Given $\mathcal{L}$ is a reinstatement RAF labelling there exists an attack β such that $t(\beta) = \alpha$, $\ell_K(\beta) = \mathtt{in}$ and $\ell_A(s(\beta)) = \mathtt{in}$. As a consequence, $\beta \in Q$ and $s(\beta) \in S$. According to Definition 9, we have so: $\alpha \in Inh(\mathcal{U})$.

2. $\ell_A(s(\alpha)) = \mathtt{out}$.

   Given $\mathcal{L}$ is a reinstatement RAF labelling there exists an attack γ such that $t(\gamma) = s(\alpha)$, $\ell_K(\gamma) = \mathtt{in}$ and $\ell_A(s(\gamma)) = \mathtt{in}$. As a consequence, $\gamma \in Q$ and $s(\gamma) \in S$. According to Definition 9, we have so: $s(\alpha) \in Def(\mathcal{U})$.

As a consequence and following Definition 10, we have: $x \in Acc(\mathcal{U})$.
We prove so that:

$$(S \cup Q) \subseteq Acc(\mathcal{U}) \tag{4.2}$$

Step 2: $(S \cup Q) \supseteq Acc(\mathcal{U})$

Let $x \in Acc(\mathcal{U})$, x being an argument or an attack. According to Definition 10, for all $\alpha \in K$ such that $t(\alpha) = x$, we have: $s(\alpha) \in Def(\mathcal{U})$ or $\alpha \in Inh(\mathcal{U})$.

Let y be $s(\alpha)$ or α. Given that $y \in (Def(\mathcal{U}) \cup Inh(\mathcal{U}))$, there exists an attack β such that $s(\beta) \in S$, $\beta \in Q$ and $t(\beta) = y$. By definition of $\texttt{Lab2Struct}(\mathcal{L})$*, we have $\ell_A(s(\beta)) = \texttt{in}$ and $\ell_K(\beta) = \texttt{in}$. Given $\mathcal{L}$ is a reinstatement RAF labelling, we have $\mathcal{L}(y) = \texttt{out}$.*

As a consequence, we have:

$$\forall \alpha \in K \text{ s.t. } t(\alpha) = x,\ \ell_A(s(\alpha)) = \texttt{out} \text{ or } \ell_K(\alpha) = \texttt{out} \tag{4.3}$$

Then, given that $\mathcal{L}$ is a reinstatement RAF labelling, we have : $\mathcal{L}(x) = \texttt{in}$. By definition of $\texttt{Lab2Struct}(\mathcal{L})$ *we have so $x \in (S \cup Q)$.*

We prove so that :

$$(S \cup Q) \supseteq Acc(\mathcal{U}) \tag{4.4}$$

Finally, because of Equations 4.2 and 4.4 we have:

$$(S \cup Q) = Acc(\mathcal{U}) \tag{4.5}$$

We prove thus prove that $\texttt{Lab2Struct}(\mathcal{L})$ *is a* complete *structure.* ■

Theorem 5 *Let $\Gamma = \langle A, K, s, t\rangle$ be a RAF and let $\mathcal{U} = \langle S, Q\rangle$ be a complete structure. Then* $\texttt{Struct2Lab}(\mathcal{U})$ *is a reinstatement RAF labelling.*

PROOF. *Let $\mathcal{L} = \texttt{Struct2Lab}(\mathcal{U})$. In order to prove that $\mathcal{L}$ is a reinstatement RAF labelling (with $\mathcal{L} = \langle \ell_A, \ell_K\rangle$) we have to prove that, for all $x \in (A \cup K)$:*

1. $(\mathcal{L}(x) = \texttt{out}) \Longrightarrow (\exists \alpha \in K \text{ s.t. } t(\alpha) = x,\ \ell_K(\alpha) = \texttt{in} \text{ and } \ell_A(s(\alpha)) = \texttt{in})$
2. $(\mathcal{L}(x) = \texttt{out}) \Longleftarrow (\exists \alpha \in K \text{ s.t. } t(\alpha) = x,\ \ell_K(\alpha) = \texttt{in} \text{ and } \ell_A(s(\alpha)) = \texttt{in})$
3. $(\mathcal{L}(x) = \texttt{in}) \Longrightarrow (\forall \alpha \in K \text{ s.t. } t(\alpha) = x,\ \ell_K(\alpha) = \texttt{out} \text{ or } \ell_A(s(\alpha)) = \texttt{out})$
4. $(\mathcal{L}(x) = \texttt{in}) \Longleftarrow (\forall \alpha \in K \text{ s.t. } t(\alpha) = x,\ \ell_K(\alpha) = \texttt{out} \text{ or } \ell_A(s(\alpha)) = \texttt{out})$

Step 1: $(\mathcal{L}(x) = \texttt{out}) \Longrightarrow (\exists \alpha \in K \text{ s.t. } t(\alpha) = x,\ \ell_K(\alpha) = \texttt{in} \text{ and } \ell_A(s(\alpha)) = \texttt{in})$

Let $x \in (A \cup K)$ be an argument or an attack such that $\mathcal{L}(x) = \texttt{out}$. According to the definition of $\texttt{Struct2Lab}(\mathcal{U})$*, we have $x \in (Def(\mathcal{U}) \cup Inh(\mathcal{U}))$. Following the definitions of $Def(\mathcal{U})$ and $Inh(\mathcal{U})$, we can state that there exists an attack α such that $\alpha \in Q$, $s(\alpha) \in S$ and $t(\alpha) = x$. According to the definition of* $\texttt{Struct2Lab}(\mathcal{U})$*, we have so $\ell_K(\alpha) = \texttt{in}$ and $\ell_A(s(\alpha)) = \texttt{in}$.*

We prove so that for all $x \in (A \cup K)$:

$$(\mathcal{L}(x) = \mathtt{out}) \Longrightarrow (\exists \alpha \in K \text{ s.t. } t(\alpha) = x, \ell_K(\alpha) = \mathtt{in} \text{ and } \ell_A(s(\alpha)) = \mathtt{in}) \tag{4.6}$$

Step 2: $(\mathcal{L}(x) = \mathtt{out}) \Longleftarrow (\exists \alpha \in K$ *s.t.* $t(\alpha) = x$, $\ell_K(\alpha) = \mathtt{in}$ *and* $\ell_A(s(\alpha)) = \mathtt{in})$

Let $x \in (A \cup K)$ be an argument or an attack. If there exists an attack $\alpha \in K$ such that $t(\alpha) = x$, $\ell_K(\alpha) = \mathtt{in}$ and $\ell_A(s(\alpha)) = \mathtt{in}$, then according to the definition of `Struct2Lab`$(\mathcal{U})$*, we have $\alpha \in Q$ and $s(\alpha) \in S$. As a consequence, we have $x \in (Def(\mathcal{U}) \cup Inh(\mathcal{U}))$. We have thus, according to the definition of* `Struct2Lab`$(\mathcal{U})$*: $\mathcal{L}(x) = \mathtt{out}$.*

We prove so that for all $x \in (A \cup K)$:

$$(\mathcal{L}(x) = \mathtt{out}) \Longleftarrow (\exists \alpha \in K \text{ s.t. } t(\alpha) = x, \ell_K(\alpha) = \mathtt{in} \text{ and } \ell_A(s(\alpha)) = \mathtt{in}) \tag{4.7}$$

Step 3: $(\mathcal{L}(x) = \mathtt{in}) \Longrightarrow (\forall \alpha \in K$ *s.t.* $t(\alpha) = x$, $\ell_K(\alpha) = \mathtt{out}$ *or* $\ell_A(s(\alpha)) = \mathtt{out})$

Let $x \in (A \cup K)$ be an argument or an attack such that $\mathcal{L}(x) = \mathtt{in}$. According to the definition of `Struct2Lab`$(\mathcal{U})$*, we have then $x \in \mathcal{U}$ and as $\mathcal{U}$ is a complete structure we have $x \in Acc(\mathcal{U})$. As a consequence, for all $\alpha \in K$ such that $t(\alpha) = x$, we have: $s(\alpha) \in Def(\mathcal{U})$ or $\alpha \in Inh(\mathcal{U})$. According to the definition of* `Struct2Lab`$(\mathcal{U})$*, we have then: $\ell_A(s(\alpha)) = \mathtt{out}$ or $\ell_K(\alpha) = \mathtt{out}$.*

We prove so that for all $x \in (A \cup K)$:

$$(\mathcal{L}(x) = \mathtt{in}) \Longrightarrow (\forall \alpha \in K \text{ s.t. } t(\alpha) = x, \ell_K(\alpha) = \mathtt{out} \text{ or } \ell_A(s(\alpha)) = \mathtt{out}) \tag{4.8}$$

Step 4: $(\mathcal{L}(x) = \mathtt{in}) \Longleftarrow (\forall \alpha \in K$ *s.t.* $t(\alpha) = x$, $\ell_K(\alpha) = \mathtt{out}$ *or* $\ell_A(s(\alpha)) = \mathtt{out})$

Let $x \in (A \cup K)$ be an argument or an attack such that for all attacks $\alpha \in K$ s.t. $t(\alpha) = x$, $\ell_K(\alpha) = \mathtt{out}$ or $\ell_A(s(\alpha)) = \mathtt{out}$. For all such attack α, we have then, according to the definition of `Struct2Lab`$(\mathcal{U})$*: $\alpha \in Inh(\mathcal{U})$ or $s(\alpha) \in Def(\mathcal{U})$. As a consequence, we have $x \in Acc(\mathcal{U})$ and so $x \in (S \cup Q)$, $\mathcal{U}$ being complete structure. According to the definition of* `Struct2Lab`$(\mathcal{U})$*, we have then: $\mathcal{L}(x) = \mathtt{in}$.*

We prove so that for all $x \in (A \cup K)$:

$$(\mathcal{L}(x) = \mathit{in}) \Longleftarrow (\forall \alpha \in K \text{ s.t. } t(\alpha) = x, \ell_K(\alpha) = \mathtt{out} \text{ or } \ell_A(s(\alpha)) = \mathtt{out}) \quad (4.9)$$

Equations 4.6, 4.7, 4.8 and 4.9 being stated, we prove thus that $\mathcal{L}$ is a reinstatement RAF labelling. ■

4.2 Preferred semantics

In this section we show that several constraints on reinstatement RAF labellings lead to the *preferred* semantics.

4.2.1 Reinstatement RAF labellings with maximal *in*

Reinstatement RAF labellings such that $\mathit{in}(\mathcal{L})$ is maximal coincide with *preferred* structures as stated by Theorems 6 and 7.

Theorem 6 *Let $\Gamma = \langle A, K, s, t\rangle$ be a RAF and let $\mathcal{L} = \langle \ell_A, \ell_K\rangle$ be a reinstatement RAF labelling such that $\mathit{in}(\mathcal{L})$ is maximal. Then* $\mathtt{Lab2Struct}(\mathcal{L})$ *is a* preferred *structure.*

Proof. *Let $\mathcal{L}$ be a reinstatement RAF labelling such that $\mathit{in}(\mathcal{L})$ is maximal. Let suppose that $\mathcal{U} = \mathtt{Lab2Struct}(\mathcal{L})$ is not a* preferred *structure. According to Definition 11, Proposition 2 and Theorem 1, there exists then a* complete *structure $\mathcal{U}'$ such that $\mathcal{U} \sqsubset \mathcal{U}'$ (strict inclusion). Let $\mathcal{L}' = \mathtt{Struct2Lab}(\mathcal{U}')$. Then $\mathit{in}(\mathcal{L}') \subset \mathit{in}(\mathcal{L})$. As a consequence $\mathcal{L}$ is not a reinstatement RAF labelling such that $\mathit{in}(\mathcal{L})$ is maximal, which is a contradiction.* ■

Theorem 7 *Let $\Gamma = \langle A, K, s, t\rangle$ be a RAF and let $\mathcal{U} = \langle S, Q\rangle$ be a* preferred *structure. Then $\mathcal{L} = \mathtt{Struct2Lab}(\mathcal{U})$ is a reinstatement RAF labelling such that $\mathit{in}(\mathcal{L})$ is maximal.*

Proof. *Let $\mathcal{U}$ be a* preferred *structure and $\mathcal{L} = \mathtt{Struct2Lab}(\mathcal{U})$. Let us suppose that $\mathcal{L}$ is not a reinstatement RAF labelling such that $\mathit{in}(\mathcal{L})$ is maximal.*

Then there exists a reinstatement RAF labelling $\mathcal{L}'$ such that $\mathtt{in}(\mathcal{L}) \subset \mathtt{in}(\mathcal{L}')$. Let $\mathcal{U}' = \mathtt{Lab2Struct}(\mathcal{L}')$. Then $\mathcal{U}'$ is a complete *structure such that $\mathcal{U} \sqsubset \mathcal{U}'$ (strict inclusion). As a consequence, $\mathcal{U}$ is not a* preferred *structure, which is a contradiction.*

■

4.2.2 Reinstatement RAF labellings with maximal $\mathtt{out}$

Reinstatement RAF labellings such that $\mathtt{out}\,(\mathcal{L})$ is maximal also coincide with *preferred* structures. In order to prove it, let first prove the two following propositions.

Proposition 4 *Let $\mathcal{L}$ and $\mathcal{L}'$ be two reinstatement RAF labellings. If $\mathtt{in}(\mathcal{L}) \subset \mathtt{in}(\mathcal{L}')$ then $\mathtt{out}(\mathcal{L}) \subset \mathtt{out}(\mathcal{L}')$.*

PROOF. *Let $\mathcal{L}$ and $\mathcal{L}'$ be two reinstatement RAF labellings such that $\mathtt{in}(\mathcal{L}) \subset \mathtt{in}(\mathcal{L}')$, meaning that:*

$$\forall w \in \mathtt{in}(\mathcal{L}),\, w \in \mathtt{in}(\mathcal{L}') \tag{4.10}$$

and

$$\exists x \in \mathtt{in}(\mathcal{L}'),\, x \notin \mathtt{in}(\mathcal{L}) \tag{4.11}$$

Let prove that $\mathtt{out}(\mathcal{L}) \subset \mathtt{out}(\mathcal{L}')$, and so that :

1. $\forall y \in \mathtt{out}(\mathcal{L}),\, y \in \mathtt{out}(\mathcal{L}')$
2. $\exists z \in \mathtt{out}(\mathcal{L}'),\, z \notin \mathtt{out}(\mathcal{L})$

Step 1: $\forall y \in \mathtt{out}(\mathcal{L}),\, y \in \mathtt{out}(\mathcal{L}')$

Let y be an attack or an argument such that $y \in \mathtt{out}(\mathcal{L})$. Given $\mathcal{L}$ is a reinstatement RAF labelling, we have by definition:

$$(\mathcal{L}(y) = \mathtt{out}) \Longrightarrow (\exists \alpha \in K \text{ s.t. } t(\alpha) = y, \ell_K(\alpha) = \mathtt{in} \text{ and } \ell_A(s(\alpha)) = \mathtt{in})$$

Then according to Equation 4.10, $\alpha \in \mathtt{in}(\mathcal{L}')$ and $s(\alpha) \in \mathtt{in}(\mathcal{L}')$. As $\mathcal{L}'$ is also a reinstatement RAF labelling, we have so $y \in \mathtt{out}(\mathcal{L}')$.

Step 2: $\exists z \in \mathtt{out}(\mathcal{L}'), z \notin \mathtt{out}(\mathcal{L})$

Let x be an attack or an argument such that $x \in \mathtt{in}(\mathcal{L}')$ and $x \notin \mathtt{in}(\mathcal{L})$. Given $\mathcal{L}$ and $\mathcal{L}'$ are reinstatement RAF labellings, we have by definition:

$$(\mathcal{L}'(x) = \mathtt{in}) \iff (\forall \alpha \in K \text{ s.t. } t(\alpha) = x, \ell'_K(\alpha) = \mathtt{out} \text{ or } \ell'_A(s(\alpha)) = \mathtt{out}) \tag{4.12}$$

$$(\mathcal{L}(x) \neq \mathtt{in}) \iff (\exists \alpha \in K \text{ s.t. } t(\alpha) = x, \ell_K(\alpha) \neq \mathtt{out} \text{ and } \ell_A(s(\alpha)) \neq \mathtt{out}) \tag{4.13}$$

Let α be such an attack with $t(\alpha) = x$, $\ell_K(\alpha) \neq \mathtt{out} \cap \ell_A(s(\alpha)) \neq \mathtt{out}$.

By definition of α we have, $\alpha \notin \mathtt{out}(\mathcal{L})$ and $s(\alpha) \notin \mathtt{out}(\mathcal{L})$. Furthermore, given that $\mathcal{L}'(x) = \mathtt{in}$, we have following Equation 4.12, $\alpha \in \mathtt{out}(\mathcal{L}')$ or $s(\alpha) \in \mathtt{out}(\mathcal{L}')$.

We prove thus that there exists z such that, $z \in \mathtt{out}(\mathcal{L}')$ and $z \notin \mathtt{out}(\mathcal{L})$.

■

Proposition 5 *Let $\mathcal{L}$ and $\mathcal{L}'$ be two reinstatement RAF labellings. If $\mathtt{out}(\mathcal{L}) \subset \mathtt{out}(\mathcal{L}')$ then $\mathtt{in}(\mathcal{L}) \subset \mathtt{in}(\mathcal{L}')$.*

Proof. *Let $\mathcal{L}$ and $\mathcal{L}'$ be two reinstatement RAF labellings such that $\mathtt{out}(\mathcal{L}) \subset \mathtt{out}(\mathcal{L}')$, meaning that:*

$$\forall w \in \mathtt{out}(\mathcal{L}), w \in \mathtt{out}(\mathcal{L}') \tag{4.14}$$

and

$$\exists x \in \mathtt{out}(\mathcal{L}'), x \notin \mathtt{out}(\mathcal{L}) \tag{4.15}$$

Let prove that $\mathtt{in}(\mathcal{L}) \subset \mathtt{in}(\mathcal{L}')$, and so that :

1. $\forall y \in \mathtt{in}(\mathcal{L}), y \in \mathtt{in}(\mathcal{L}')$
2. $\exists z \in \mathtt{in}(\mathcal{L}'), z \notin \mathtt{in}(\mathcal{L})$

Step 1: $\forall y \in \mathtt{in}(\mathcal{L}), y \in \mathtt{in}(\mathcal{L}')$

Let y be an attack or an argument such that $y \in \mathtt{in}(\mathcal{L})$. Given $\mathcal{L}$ is a reinstatement RAF labelling, we have by definition:

$$(\mathcal{L}(y) = \mathtt{in}) \Longrightarrow (\forall \alpha \in K \text{ s.t. } t(\alpha) = y, \ell_K(\alpha) = \mathtt{out} \text{ or } \ell_A(s(\alpha)) = \mathtt{out})$$

Then according to Equation 4.14, we have:

$$(\mathcal{L}(y) = \mathtt{in}) \Longrightarrow (\forall \alpha \in K \text{ s.t. } t(\alpha) = y, \ell'_K(\alpha) = \mathtt{out} \text{ or } \ell'_A(s(\alpha)) = \mathtt{out})$$

As $\mathcal{L}'$ is also a reinstatement RAF labelling, we have then $y \in \mathtt{in}(\mathcal{L}')$.

Step 2: $\exists z \in \mathtt{in}(\mathcal{L}'), z \notin \mathtt{in}(\mathcal{L})$

Let x be an attack or an argument such that $x \in \mathtt{out}(\mathcal{L}')$ and $x \notin \mathtt{out}(\mathcal{L})$. Given $\mathcal{L}$ and $\mathcal{L}'$ are reinstatement RAF labellings, we have by definition:

$$(\mathcal{L}'(x) = \mathtt{out}) \iff (\exists \alpha \in K \text{ s.t. } t(\alpha) = x, \ell'_K(\alpha) = \mathtt{in} \text{ and } \ell'_A(s(\alpha)) = \mathtt{in}) \tag{4.16}$$

$$(\mathcal{L}(x) \neq \mathtt{out}) \iff (\forall \alpha \in K \text{ s.t. } t(\alpha) = x, \ell_K(\alpha) \neq \mathtt{in} \text{ or } \ell_A(s(\alpha)) \neq \mathtt{in}) \tag{4.17}$$

According to Equation 4.17, for all attack α such that $t(\alpha) = x$, we have: $\alpha \notin \mathtt{in}(\mathcal{L})$ or $s(\alpha) \notin \mathtt{in}(\mathcal{L})$. However, we have following Equation 4.16, there exists at least one attack $\alpha \in K$ s.t. $\alpha \in \mathtt{in}(\mathcal{L}')$ and $s(\alpha) \in \mathtt{in}(\mathcal{L}')$.

We prove thus that there exists z such that, $z \in \mathtt{in}(\mathcal{L}')$ and $z \notin \mathtt{in}(\mathcal{L})$. ■

Now that Propositions 4 and 5 have been proven, let now prove that reinstatement RAF labellings such that $\mathtt{out}(\mathcal{L})$ is maximal coincide with *preferred* structures.

Theorem 8 *Let $\Gamma = \langle A, K, s, t \rangle$ be a RAF and let $\mathcal{L} = \langle \ell_A, \ell_K \rangle$ be a reinstatement RAF labelling such that $\mathtt{out}(\mathcal{L})$ is maximal. Then* $\mathtt{Lab2Struct}(\mathcal{L})$ *is a* preferred *structure.*

PROOF. *Let $\mathcal{L}$ be a reinstatement RAF labelling such that $\mathtt{out}(\mathcal{L})$ is maximal. Let suppose that* $\mathtt{Lab2Struct}(\mathcal{L})$ *is not a* preferred *structure. Then according to Theorem 6, $\mathtt{in}(\mathcal{L})$ is not maximal. There exists thus a reinstatement RAF labelling $\mathcal{L}'$ such that $\mathtt{in}(\mathcal{L}) \subset \mathtt{in}(\mathcal{L}')$. We have then, following Proposition 4, $\mathtt{out}(\mathcal{L}) \subset \mathtt{out}(\mathcal{L}')$, which is a contradiction.* ■

Theorem 9 *Let* $\Gamma = \langle A, K, s, t\rangle$ *be a RAF and let* $\mathcal{U} = \langle S, Q\rangle$ *be a* preferred *structure. Then* $\mathcal{L} = \texttt{Struct2Lab}(\mathcal{U})$ *is a reinstatement RAF labelling such that* $\mathtt{out}(\mathcal{L})$ *is maximal.*

PROOF. *Let* $\mathcal{U}$ *be a* preferred *structure. According to Theorem 7,* $\mathcal{L} = \texttt{Struct2Lab}(\mathcal{U})$ *is a reinstatement RAF labelling such that* $\mathtt{in}(\mathcal{L})$ *is maximal. Let suppose that* $\mathtt{out}(\mathcal{L})$ *is not maximal. There exist thus a reinstatement RAF labelling* $\mathcal{L}'$ *such that* $\mathtt{out}(\mathcal{L}) \subset \mathtt{out}(\mathcal{L}')$*. We have then, following Proposition 5,* $\mathtt{in}(\mathcal{L}) \subset \mathtt{in}(\mathcal{L}')$*, which is a contradiction.* ■

4.3 Stable semantics: reinstatement RAF labellings with empty $\mathtt{und}$

Reinstatement RAF labellings such that $\mathtt{und}(\mathcal{L})$ is empty coincide with *stable* structures as stated by Theorems 10 and 11.

Theorem 10 *Let* $\Gamma = \langle A, K, s, t\rangle$ *be a RAF and let* $\mathcal{L} = \langle \ell_A, \ell_K\rangle$ *be a reinstatement RAF labelling such that* $\mathtt{und}(\mathcal{L}) = \varnothing$*. Then* $\texttt{Lab2Struct}(\mathcal{L})$ *is a* stable *structure.*

PROOF. *Let* $\mathcal{L} = \langle \ell_A, \ell_K\rangle$ *be a reinstatement RAF labelling such that* $\mathtt{und}(\mathcal{L}) = \varnothing$*. Let* $\mathcal{U} = \texttt{Lab2Struct}(\mathcal{L})$*. Let* x *be any attack or argument such that* $x \notin \mathcal{U}$*.*

Given that $\mathtt{und}(\mathcal{L}) = \varnothing$*, we have according to Definition 16:* $\mathcal{L}(x) = \mathtt{out}$*. There exists then an attack* α *such that:* $\ell_A(s(\alpha)) = \mathtt{in} \cap \ell_K(\alpha) = \mathtt{in}$*. We have then* $s(\alpha) \in \mathcal{U}$ *and* $\alpha \in \mathcal{U}$*.*

Therefore, according to Definition 9, we have: $x \in Def(\mathcal{U})$ *or* $x \in Inh(\mathcal{U})$*, following the nature of* x*. This means that* $\mathcal{U}$ *defeats or inhibits any argument and attack which is not in it.*

We prove so that $\mathcal{U}$ *is, thus, a* stable *structure.* ■

Theorem 11 *Let* $\Gamma = \langle A, K, s, t\rangle$ *be a RAF and let* $\mathcal{U} = \langle S, Q\rangle$ *be a* stable *structure. Then* $\mathcal{L} = \texttt{Struct2Lab}(\mathcal{U})$ *is a reinstatement RAF labelling such that* $\mathtt{und}(\mathcal{L})$ *is empty.*

PROOF. *Let* $\mathcal{U}$ *be a* stable *structure and let x be an argument or an attack.*

If $x \in \mathcal{U}$ *then* $\mathcal{L}(x) = \mathtt{in}$.

Let consider the case when $x \notin \mathcal{U}$. *Given* $\mathcal{U}$ *is a* stable *structure then there exists an attack* α *in* $\mathcal{U}$ *that defeats or inhibits x. We have then, according to Definition 16:* $\mathcal{L}(x) = \mathtt{out}$.

In both cases $\mathcal{L}(x) \neq \mathtt{und}$. *We prove so that* $\mathtt{und}(\mathcal{L}) = \varnothing$. ■

4.4 Grounded semantics

In this section we show that several constraints on reinstatement RAF labellings lead to the *grounded* semantics.

4.4.1 Reinstatement RAF labellings with maximal $\mathtt{und}$

Reinstatement RAF labellings such that $\mathtt{und}(\mathcal{L})$ is maximal coincide with the *grounded* structure as stated by Theorems 12 and 13.

Theorem 12 *Let* $\Gamma = \langle A, K, s, t \rangle$ *be a RAF and let* $\mathcal{L} = \langle \ell_A, \ell_K \rangle$ *be a reinstatement RAF labelling such that* $\mathtt{und}(\mathcal{L})$ *is maximal. Then* $\mathtt{Lab2Struct}(\mathcal{L})$ *is the* grounded *structure.*

PROOF. *Let* $\mathcal{L} = \langle \ell_A, \ell_K \rangle$ *be a reinstatement RAF labelling such that* $\mathtt{und}(\mathcal{L})$ *is maximal. Let suppose that* $\mathcal{U} = \mathtt{Lab2Struct}(\mathcal{L})$ *is not the* grounded *structure. According to Theorem 4,* $\mathcal{U}$ *is a* complete *structure. By definition of the* grounded *structure (Definition 11), we can thus say that there exists a structure* $\mathcal{U}'$ *that is the grounded structure and such that* $\mathcal{U}' \sqsubset \mathcal{U}$ *(strict inclusion). Let* $\mathcal{L}' = \mathtt{Struct2Lab}(\mathcal{U}')$ *be the reinstatement RAF labelling corresponding with the* grounded *structure.*

As $\mathcal{U}' \sqsubset \mathcal{U}$ *we have, by definition of* $\mathtt{Struct2Lab}$*:* $\mathtt{in}(\mathcal{L}') \subset \mathtt{in}(\mathcal{L})$

Following Proposition 4, we thus also have: $\mathtt{out}(\mathcal{L}') \subset \mathtt{out}(\mathcal{L})$

As a consequence, we can say that: $\mathtt{und}(\mathcal{L}) \subset \mathtt{und}(\mathcal{L}')$

There is a contradiction.

We prove so that $\mathcal{U}$ *is, thus, the* grounded *structure.*

■

Theorem 13 *Let* $\Gamma = \langle A, K, s, t \rangle$ *be a RAF and let* $\mathcal{U} = \langle S, Q \rangle$ *be the* grounded *structure. Then* $\mathcal{L} = \mathtt{Struct2Lab}(\mathcal{U})$ *is a reinstatement RAF labelling such that* $\mathtt{und}(\mathcal{L})$ *is maximal.*

Proof. *Let* $\Gamma = \langle A, K, s, t \rangle$ *be a RAF, let* $\mathcal{U} = \langle S, Q \rangle$ *be the* grounded *structure and* $\mathcal{U}'$ *be any* complete *structure that is not* grounded. *Let* $\mathcal{L} = \mathtt{Struct2Lab}(\mathcal{U})$ *and* $\mathcal{L}' = \mathtt{Struct2Lab}(\mathcal{U}')$.

According to Definition 11, we have: $\mathcal{U} \sqsubset \mathcal{U}'$.

By definition of Struct2Lab, *we thus have:* $\mathtt{in}(\mathcal{L}) \subset \mathtt{in}(\mathcal{L}')$.

Following Proposition 4, we thus also have: $\mathtt{out}(\mathcal{L}) \subset \mathtt{out}(\mathcal{L}')$.

As a consequence, we have: $\mathtt{und}(\mathcal{L}') \subset \mathtt{und}(\mathcal{L})$.

We prove so that $\mathcal{L} = \mathtt{Struct2Lab}(\mathcal{U})$ *is a reinstatement RAF labelling such that* $\mathtt{und}(\mathcal{L})$ *is maximal.*

■

4.4.2 Reinstatement RAF labellings with minimal $\mathtt{in}$

Reinstatement RAF labellings such that $\mathtt{in}(\mathcal{L})$ is minimal coincide with the *grounded* structure as stated by Theorems 14 and 15.

Theorem 14 *Let* $\Gamma = \langle A, K, s, t \rangle$ *be a RAF and let* $\mathcal{L} = \langle \ell_A, \ell_K \rangle$ *be a reinstatement RAF labelling such that* $\mathtt{in}(\mathcal{L})$ *is minimal. Then* $\mathtt{Lab2Struct}(\mathcal{L})$ *is the* grounded *structure.*

Proof. $\mathcal{L}$ *be a reinstatement RAF labelling such that* $\mathtt{in}(\mathcal{L})$ *is minimal. Let suppose that* $\mathcal{U} = \mathtt{Lab2Struct}(\mathcal{L})$ *is not the* grounded *structure. By definition of the* grounded *structure (Definition 11), we can thus say that there exists a structure* $\mathcal{U}'$ *that is the grounded structure and such that* $\mathcal{U}' \sqsubset \mathcal{U}$ *(strict inclusion). Let* $\mathcal{L}' = \mathtt{Struct2Lab}(\mathcal{U}')$ *be the reinstatement RAF labelling corresponding with the* grounded *structure. As* $\mathcal{U}' \sqsubset \mathcal{U}$ *we have, by definition of* Struct2Lab*:* $\mathtt{in}(\mathcal{L}') \subset \mathtt{in}(\mathcal{L})$. *We have then a contradiction.*

We prove so that $\mathcal{U} = \mathtt{Lab2Struct}(\mathcal{L})$ *is the* grounded *structure.* ■

Theorem 15 *Let* $\Gamma = \langle A, K, s, t \rangle$ *be a RAF and let* $\mathcal{U} = \langle S, Q \rangle$ *be the* grounded *structure. Then* $\mathcal{L} = \mathtt{Struct2Lab}(\mathcal{U})$ *is a reinstatement RAF labelling such that* $\mathtt{in}(\mathcal{L})$ *is minimal.*

PROOF. *Let* $\mathcal{U}$ *be the* grounded *structure and* $\mathcal{L} = \texttt{Struct2Lab}(\mathcal{U})$. *Let suppose that* $\mathtt{in}(\mathcal{L})$ *is not minimal. There exists then a reinstatement RAF labelling* $\mathcal{L}'$ *such that* $\mathtt{in}(\mathcal{L}') \subset \mathtt{in}(\mathcal{L})$. *Let* $\mathcal{U}' = \texttt{Lab2Struct}(\mathcal{L}')$. *From the definition of* $\texttt{Lab2Struct}$, *we can say that:* $\mathcal{U}' \sqsubset \mathcal{U}$. *This contradicts the definition of the grounded structure (Definition 11).*

We prove so that $\mathcal{L} = \texttt{Struct2Lab}(\mathcal{U})$ *is a reinstatement RAF labelling such that* $\mathtt{in}(\mathcal{L})$ *is minimal.* ■

4.4.3 Reinstatement RAF labellings with minimal $\mathtt{out}$

Note that reinstatement RAF labellings such that $\mathtt{out}(\mathcal{L})$ is minimal coincide with the *grounded* structure as stated by Theorems 16 and 17.

Theorem 16 *Let* $\Gamma = \langle A, K, s, t\rangle$ *be a RAF and let* $\mathcal{L} = \langle \ell_A, \ell_K\rangle$ *be a reinstatement RAF labelling such that* $\mathtt{out}(\mathcal{L})$ *is minimal. Then* $\texttt{Lab2Struct}(\mathcal{L})$ *is the* grounded *structure.*

PROOF. $\mathcal{L}$ *be a reinstatement RAF labelling such that* $\mathtt{out}(\mathcal{L})$ *is minimal. Following Proposition 5,* $\mathtt{in}(\mathcal{L})$ *is also minimal. Therefore, according to Theorem 14,* $\texttt{Lab2Struct}(\mathcal{L})$ *is the* grounded *structure.* ■

Theorem 17 *Let* $\Gamma = \langle A, K, s, t\rangle$ *be a RAF and let* $\mathcal{U} = \langle S, Q\rangle$ *be the* grounded *structure. Then* $\mathcal{L} = \texttt{Struct2Lab}(\mathcal{U})$ *is a reinstatement RAF labelling such that* $\mathtt{out}(\mathcal{L})$ *is minimal.*

PROOF. *Let* $\mathcal{U}$ *be the* grounded *structure and* $\mathcal{L} = \texttt{Struct2Lab}(\mathcal{U})$. *According to Theorem 15,* $\mathtt{in}(\mathcal{L})$ *is minimal. Following Proposition 4,* $\mathtt{out}(\mathcal{L})$ *is also minimal.* $\mathcal{L}$ *is thus a reinstatement RAF labelling such that* $\mathtt{out}(\mathcal{L})$ *is minimal.* ■

4.5 Semi-stable semantics

Reinstatement RAF labellings such that $\mathtt{und}(\mathcal{L})$ is minimal coincide with *semi-stable* structures as stated by Theorems 18 and 19.

Theorem 18 *Let $\Gamma = \langle A, K, s, t\rangle$ be a RAF and let $\mathcal{L} = \langle \ell_A, \ell_K\rangle$ be a reinstatement RAF labelling such that* $\mathtt{und}(\mathcal{L})$ *is minimal. Then* $\mathtt{Lab2Struct}(\mathcal{L})$ *is an* semi-stable *structure.*

PROOF. *Let $\Gamma = \langle A, K, s, t\rangle$ be a RAF. Let $\mathcal{L} = \langle \ell_A, \ell_K\rangle$ be a reinstatement RAF labelling such that* $\mathtt{und}(\mathcal{L})$ *is minimal. Let suppose that $\mathcal{U} = \mathtt{Lab2Struct}(\mathcal{L})$ is not an* semi-stable *structure (with $\mathcal{U} = \langle S, Q\rangle$). There exists thus an* semi-stable *structure $\mathcal{U}' = \langle S', Q'\rangle$ such that:*

$$(S \cup Q \cup Def(\mathcal{U}) \cup Inh(\mathcal{U})) \subset (S' \cup Q' \cup Def(\mathcal{U}') \cup Inh(\mathcal{U}')) \tag{4.18}$$

As a consequence, we have:

$$(A \cup K) \setminus (S \cup Q \cup Def(\mathcal{U}) \cup Inh(\mathcal{U})) \supset (A \cup K) \setminus (S' \cup Q' \cup Def(\mathcal{U}') \cup Inh(\mathcal{U}')) \tag{4.19}$$

Let $\mathcal{L}' = \mathtt{Struct2Lab}(\mathcal{U}')$. Following Equation 4.19, we have according to the definition of $\mathtt{Struct2Lab}$*:*

$$\mathtt{und}(\mathcal{L}) \supset \mathtt{und}(\mathcal{L}')$$

Then $\mathtt{und}(\mathcal{L})$ is not minimal, which is a contradiction.
We prove so that $\mathcal{U} = \mathtt{Lab2Struct}(\mathcal{L})$ is an semi-stable *structure.* ■

Theorem 19 *Let $\Gamma = \langle A, K, s, t\rangle$ be a RAF and let $\mathcal{U} = \langle S, Q\rangle$ be an* semi-stable *structure. Then $\mathcal{L} = \mathtt{Struct2Lab}(\mathcal{U})$ is a reinstatement RAF labelling such that $\mathtt{und}(\mathcal{L})$ is minimal.*

PROOF. *Let $\mathcal{U} = \langle S, Q\rangle$ be an* semi-stable *structure. By definition, $S \cup Q \cup Def(\mathcal{U}) \cup Inh(\mathcal{U})$ is thus maximal. As a consequence, $(A \cup K) \setminus (S \cup Q \cup Def(\mathcal{U}) \cup Inh(\mathcal{U}))$ is minimal. Let $\mathcal{L} = \mathtt{Struct2Lab}(\mathcal{U})$. According to the definition of* $\mathtt{Struct2Lab}$ *and following the previous statement, we have thus $\mathtt{und}(\mathcal{L})$ being minimal.*

We prove so that $\mathcal{L} = \mathtt{Struct2Lab}(\mathcal{U})$ is a reinstatement RAF labelling such that $\mathtt{und}(\mathcal{L})$ is minimal. ■

4.6 A one-to-one mapping

In this section is summarized the relations between labellings and structures in RAF and is also presented the links between labellings in AF and RAF with no recursive attacks.

4.6.1 Structures and labellings in RAF

Table 4.1 sums up the whole previous sections of Chapter 4. It shows the correspondence between structure semantics and reinstatement RAF labellings.

Restriction on Reinstatement RAF labelling	Semantics	Theorems
no restrictions	*complete* semantics	Theorems 4 and 5
empty *und*	*stable* semantics	Theorems 10 and 11
maximal *in*	*preferred* semantics	Theorems 6 and 7
maximal *out*	*preferred* semantics	Theorems 8 and 9
maximal *und*	*grounded* semantics	Theorems 12 and 13
minimal *in*	*grounded* semantics	Theorems 14 and 15
minimal *out*	*grounded* semantics	Theorems 16 and 17
minimal *und*	*semi-stable* semantics	Theorems 18 and 19

Table 4.1: Reinstatement RAF labellings and structures semantics

4.6.2 AF labellings and RAF labellings when no recursive attack exists

As stated in Section 3.1.2, there exists a one-to-one mapping between structures and extensions in RAF without recursive attacks for the *complete*, *grounded*, *preferred*, *semistable* and *stable* semantics.

[5] established a one-to-one mapping between AF extensions and AF reinstatement labellings for the mentioned semantics. In this report, as summarized in Section 4.6.1, is established a one-to-one mapping between RAF structures and reinstatement RAF labellings for the same semantics.

As a consequence, for RAF with no recursive attacks, there exists a one-to-one mapping between reinstatement labellings (AF notion) and structures (RAF notion) and also between reinstatement labellings (AF notion) and reinstatement RAF labellings (RAF notion).

Chapter 5

Conclusion and perspectives

The main contribution of this report is the extension of the notion of AF reinstatement labelling semantics to RAF. Another contribution is the extension of the *semi-stable* semantics, which was defined for AF, to RAF.

Given a RAF, a *RAF labelling* is a tuple in which the first element is a labelling over its arguments and the second one is a labelling over its attacks. These labellings are three-value based. These values indicate the degree of acceptance of an AF element (an argument or an attack). It could be accepted (*in*), rejected (*out*) or in an undecidable state (*und*).

A one-to-one mapping between structures and RAF labellings is defined using two linking functions: `Struct2Lab`, that transforms a structure into a RAF labelling and `Lab2Struct`, that transforms a RAF labelling into a structure. Given a structure and its corresponding RAF labelling, an argument (respectively an attack) will be labelled *in* if and only if it is in the structure, *out* if and only if it is defeated (respectively inhibited) by the structure and *und* if and only if is not in the structure but not defeated (respectively inhibited) by the structure.

We formally define *reinstatement RAF labellings* as coherent RAF labellings. It is shown (see Section 3.1.2) that the *complete*, *grounded*, *preferred*, *semistable* and *stable* structure semantics correspond to precise types of *reinstatement RAF labellings*. Table 4.1 gives the correspondence between structure semantics and reinstatement RAF labellings.

Moreover, we confirm by this work (see also [6]) that RAF are a conservative generalization of Dung's AF. Indeed in RAF with no recursive attacks, there is a one-to-one mapping between reinstatement labellings (AF notion) and two RAF notions (structures and reinstatement RAF labellings) for the *complete*, *grounded*, *preferred*, *semistable* and *stable* semantics.

The reason why we defined *reinstatement RAF labellings* is that they are more precise on the acceptability status of a RAF element than structure and mostly because it opens a whole new field of research for RAF solving algorithms.

As a perspective, we want to propose some solving algorithms for RAF semantics based on a labelling search.

We are also interested in the RAF semantics complexity. In [6], the complexity of the credulous and skeptical acceptance problems has been shown to be the same whether in AF or in RAF for the *complete*, *preferred* and *stable* semantics. We want to study other decision problems such as "non empty existence" (Does a given semantics produce a non empty structure ?) and "verification" (Is a given structure, a structure of a given semantics ?).

While a decision problem consists in answering a yes/no question, a "function problem" gives a more complex output. Basically, in addition to saying that a solution of a given problem exists (decision problem), the answer of the function problem associated with it will be the solution found. As in practice, argumentation solvers have to concretely compute solutions, we are also interested in the complexity of the function versions of those mentioned decision problems.

Bibliography

[1] P. Baroni, M. Caminada, and M. Giacomin. An introduction to argumentation semantics. *Knowledge Eng. Review*, 26(4):365–410, 2011.

[2] P. Baroni, F. Cerutti, P. E. Dunne, and M. Giacomin. Computing with infinite argumentation frameworks: The case of AFRAs. In *Proc. of TAFA, Revised Selected Papers*, pages 197–214, 2011.

[3] P. Baroni, F. Cerutti, M. Giacomin, and G. Guida. AFRA: Argumentation framework with recursive attacks. *Intl. Journal of Approximate Reasoning*, 52:19–37, 2011.

[4] H. Barringer, D. M. Gabbay, and J. Woods. Temporal dynamics of support and attack networks : From argumentation to zoology. In D. Hutter and W. Stephan, editors, *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday. LNAI 2605*, pages 59–98. Springer Verlag, 2005.

[5] M. Caminada. On the issue of reinstatement in argumentation. In *JELIA*, pages 111–123, 2006.

[6] C. Cayrol, J. Fandinno, L. Fariñas del Cerro, and M.-C. Lagasquie-Schiex. Valid attacks in argumentation frameworks with recursive attacks. In *13th International Symposium on Commonsense Reasoning (Commonsense)*, volume 2052. CEUR-WS : Workshop proceedings, 2017.

[7] C. Cayrol, J. Fandinno, L. Fariñas del Cerro, and M.-C. Lagasquie-Schiex. Valid attacks in Argumentation Frameworks with Recursive Attacks (long version). Rapport de recherche IRIT/RR–2019–02–FR, IRIT, 2019.

[8] Günther Charwat, Wolfgang Dvořák, Sarah Alice Gaggl, Johannes Peter Wallner, and Stefan Woltran. Methods for solving reasoning problems in abstract argumentation — A survey. *Artificial Intelligence*, 220:28–63, 2015.

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

[10] Hengfei Li and Jiachao Wu. Semantics of extended argumentation frameworks defined by renovation sets. In *International Conference on Principles and Practice of Multi-Agent Systems (PRIMA)*, pages 532–540. Springer, 2019.

[11] S. Modgil. An abstract theory of argumentation that accommodates defeasible reasoning about preferences. In *Proc. of ECSQARU*, pages 648–659, 2007.

[12] S. Modgil. Reasoning about preferences in argumentation frameworks. *Artif. Intell.*, 173:901–934, 2009.