Argumentation Frameworks with Higher-Order Attacks: Complexity results

Sylvie Doutre
Mickaël Lafages
Marie-Christine Lagasquie-Schiex

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

Tech. Report
IRIT/RR–2020–03–FR

Abstract

Computation and decision problems related to argumentation frameworks with higher-order attacks have not received a lot of attention so far. This paper is a step towards these issues: it investigates the complexity of decision problems associated with RAF, and with another kind of framework with higher-order attacks, the Argumentation Frameworks with Recursive Attacks (AFRA). This investigation shows that, for the higher expressiveness offered by these enriched systems, the complexity is the same as for classical argumentation frameworks.

Contents

Chapter 1

Introduction

Argumentation, by considering arguments and their interactions, is a way of reasoning that has proven successful in many contexts, multi-agent applications for instance (e.g. [5]). Considering a formal representation of this reasoning model, argumentation frameworks with higher-order attacks (e.g. [4, 15, 16, 2, 3]) are a rich extension of classical Argumentation Framework (AF) by [11]: not only they consider arguments and attacks between arguments, but also attacks on attacks.

Among these frameworks, Argumentation Frameworks with Recursive Attacks (AFRA) by [2, 3] are higher-order systems that produce acceptable sets of arguments and/or attacks using the definition of a specific defeat relation. If the base modelling ingredients are the same, but without the introduction of a specific defeat relation, Recursive Argumentation Framework (RAF) by [6] propose a direct approach regarding acceptability, which outputs sets of arguments and/or attacks (defined under the notion of structure), keeping the full expressiveness of higher-order attacks. The main difference between these two approaches is the way that is chosen for expressing the impact of an attack that can be in turn attacked. In the AFRA approach, the link between the attack and its source is lost and some drawbacks appear implying the fact that AFRA are not a conservative generalisation of AF. That is not the case in the RAF approach and this characteristics makes RAF particularly interesting to consider.

Classical decision problems for AF are adapted to AFRA and RAF, and their complexity is investigated. It shows that, even if the expressive power of the frameworks with higher-order attacks is higher, the complexity keeps the same as in AF.

Chapter 2

Abstract Argumentation Frameworks

2.1 Dung's abstract argumentation framework

In [11], Dung introduced a framework to represent argumentation in an abstract way.

Definition 1 (Dung's abstract argumentation framework [11]) *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* $S \cup Def(S) = A$.

Given an AF Γ, we denote by $\sigma(\Gamma)$ the set of extensions of Γ under semantics σ, with σ being either the complete *(co), the* grounded *(gr), the* stable *(st), the* semi-stable *(eg) or the preferred (pr) semantics.*

2.2 AFRA: Argumentation framework with recursive attacks

Definition 4 (AFRA [1]) *An Argumentation Framework with Recursive Attacks (AFRA) is a pair $\Gamma = \langle A, K \rangle$ where A is a set of arguments and K is a set of attacks, namely pairs (a, x) such that $a \in A$ and $x \in (A \cup K)$.*

Given an attack $\alpha = (a, x)$, we say that a is the source of α (denoted as $s(\alpha)$) and x is the target of α (denoted as $t(\alpha)$).

Definition 5 (Direct defeat [1]) *Let $\Gamma = \langle A, K \rangle$ be an AFRA, $\alpha \in K$ be an attack and $x \in (A \cup K)$ be an argument or an attack. We say that α directly defeats x iff $t(\alpha) = x$.*

Definition 6 (Indirect defeat [1]) *Let $\Gamma = \langle A, K \rangle$ be an AFRA, $\alpha \in K$ be an attack and $\beta \in K$ be an attack. We say that α indirectly defeats β iff $t(\alpha) = s(\beta)$.*

Definition 7 (Defeat [1]) *Let $\Gamma = \langle A, K \rangle$ be an AFRA, $\alpha \in K$ be an attacks and $x \in (A \cup K)$ be an argument or an attack. We say that α defeats β, denoted as $\alpha \rightarrow_K x$, iff α directly or indirectly defeat β.*

Let S be a subset of the elements of Γ. We denote by AFRA-$Def(S) = \{x | x \in (A \cup K), \exists \alpha \in S \text{ s.t. } \alpha \rightarrow_K x\}$ the set of all the elements defeated by S.

Definition 8 (Acceptability [1]) *Let $\Gamma = \langle A, K \rangle$ be an AFRA, $S \subseteq (A \cup K)$ be a subset of the elements of Γ and $x \in (A \cup K)$ be an argument or an attack. We say that x is acceptable w.r.t. S (or defended by S) iff $\forall \alpha \in K$ s.t. $\alpha \rightarrow_K x$, $\exists \beta \in S$ s.t. $\beta \rightarrow_K \alpha$.*

Let S be a subset of the elements of Γ. We denote by AFRA-$Acc(S) = \{x | x \in (A \cup K), \forall \alpha \in K \text{ s.t. } \alpha \rightarrow_K x, \exists \beta \in S \text{ s.t. } \beta \rightarrow_K \alpha\}$ the set of all the elements defended by S.

As for Dung's Abstract Argumentation Framework, based on this notion of acceptability, semantics have been defined for AFRAs. We will focus only on semantics we are interested in although much semantics have been defined.

Definition 9 (AFRA Semantics [1]) *Let $\Gamma = \langle A, K \rangle$ be an AFRA and $S \subseteq (A \cup K)$ be a subset of its elements. S is said to be an extension:*

1. *AFRA*-conflict-free *iff $S \cap$ AFRA-$Def(S) = \varnothing$.*
2. *AFRA*-admissible *iff it is AFRA*-conflict-free *and $S \subseteq$ AFRA-$Acc(S)$.*
3. *AFRA*-complete *iff it is AFRA*-conflict-free *and $S =$ AFRA-$Acc(S)$.*
4. *AFRA*-preferred *iff it is a $\subseteq$-maximal AFRA*-admissible *extension.*
5. *AFRA*-grounded *iff it is a $\subseteq$-minimal AFRA*-complete *extension.*
6. *AFRA*-stable *iff it is AFRA*-conflict-free *and $S \cup$ AFRA-$Def(S) = (A \cup K)$.*
7. *AFRA*-semi-stable *extension iff it is an AFRA*-complete *extension such that $S \cup$ AFRA-$Def(S)$ is maximal w.r.t. $\subseteq$.*

Given an AFRA Γ, we denote by $\sigma(\Gamma)$ the set of extensions of Γ under semantics σ, with σ being either the *complete* (*co*), the *grounded* (*gr*), the *stable* (*st*), the *semi-stable* (*ss*) or the preferred (*pr*) semantics.

There exists a way to express AFRAs as AFs.

Definition 10 (AFRA expressed as AF [1]) *Let $\Gamma = \langle A, K \rangle$ be an AFRA. The corresponding AF of Γ, $\widetilde{\Gamma} = \langle \widetilde{A}, \widetilde{K} \rangle$ is defined as following:*

- $\widetilde{A} = A \cup K$
- $\widetilde{K} = \{(a,b) | (a,b) \in (A \cup K)^2 \text{ and } a \rightarrow_K b\}$

In [1] has been shown a very important result concerning AFRAs and their corresponding AFs: there exists a one-to-one correspondence between extensions in AFRAs and their corresponding AFs for some semantics. Here we will focus on the semantics we are interested in but the result shown in [1] concerned much semantics.

Proposition 1 (Semantics correspondence: AFRA expressed as AF [1]) *Let $\Gamma = \langle A, K \rangle$ be an AFRA and $\widetilde{\Gamma} = \langle \widetilde{A}, \widetilde{K} \rangle$ its corresponding AF. Let $S \subseteq A \cup K$.*

- *S is an AFRA-*complete *extension for Γ iff S is a* complete *extension for $\widetilde{\Gamma}$.*
- *S is an AFRA-*preferred *extension for Γ iff S is a* preferred *extension for $\widetilde{\Gamma}$.*
- *S is an AFRA-*grounded *extension for Γ iff S is a* grounded *extension for $\widetilde{\Gamma}$.*
- *S is an AFRA-*stable *extension for Γ iff S is a* stable *extension for $\widetilde{\Gamma}$.*
- *S is an AFRA-*semi-stable *extension for Γ iff S is a* semi-stable *extension for $\widetilde{\Gamma}$.*

Note that this correspondence does not correspond to a conservative generalization of AF. Indeed, if we consider for instance a very simple example of AFRA with only two attacks: α from a to b and β from b to c (so this AFRA is an AF). Then the set $\{\alpha, c\}$ is an AFRA-*admissible* set whereas if we read this AFRA as an AF c cannot be accepted without a. This is due to the fact that the link between an attack and its source is broken in the AFRA semantics.

2.3 RAF: Recursive argumentation framework

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 [15, 16], 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).

RAF semantics need neither the introduction of additional elements in the framework, nor its transformation into another framework (as it is done, for instance, in [2, 3] that either creates a new defeat relation, or uses a "flattening process" that transforms an AFRA into a Dung AF and then uses AF semantics). A common point between RAF and AFRA 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 [14]). Note that all definitions and propositions related to RAF are issued from [6, 7], except for the *semi-stable* semantics which is a contribution of this paper.

Definition 11 (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.*

Definition 12 (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 13 (Defeat and Inhibition [6]) *Let $\mathcal{U} = \langle S, Q\rangle$ be a structure.*

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

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

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

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

Definition 14 (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) $s(\alpha) \in RAF\text{-}Def(\mathcal{U})$ or (ii) $\alpha \in RAF\text{-}Inh(\mathcal{U})$.*

By $RAF\text{-}Acc(\mathcal{U})$ 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 15 (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. *RAF*-conflict-free *iff $S \cap RAF\text{-}Def(\mathcal{U}) = \varnothing$ and $Q \cap RAF\text{-}Inh(\mathcal{U}) = \varnothing$.*

2. *RAF*-admissible *iff it is RAF*-conflict-free *and $(S \cup Q) \subseteq RAF\text{-}Acc(\mathcal{U})$.*

3. *RAF*-complete *iff it is RAF*-conflict-free *and $(S \cup Q) = RAF\text{-}Acc(\mathcal{U})$.*

4. *RAF*-preferred *iff it is a $\sqsubseteq$-maximal RAF*-admissible *structure.*

5. *RAF*-grounded *iff it is a $\sqsubseteq$-minimal RAF*-complete *structure.*

6. *RAF*-arg-preferred *iff it is a $\sqsubseteq_{ar}$-maximal RAF*-admissible *structure.*

7. *RAF*-stable *iff $S = A \setminus RAF\text{-}Def(\mathcal{U})$ and $Q = K \setminus RAF\text{-}Inh(\mathcal{U})$.*

In [6], Proposition 2 and 3 and Theorem 1 have been proven.

Proposition 2 ([6]) *There is always a unique RAF*-grounded *structure.*

Proposition 3 ([6]) *The set of all RAF*-admissible *structures forms a complete partial order with respect to $\sqsubseteq$. Furthermore, for every RAF*-admissible *structure $\mathcal{U}$, there exists a RAF*-preferred *(and a RAF*-arg-preferred *) $\mathcal{U}'$ such that $\mathcal{U} \sqsubseteq \mathcal{U}'$.*

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

- *every RAF*-complete *structure is also RAF*-admissible*,*
- *every RAF*-preferred *structure is also RAF*-complete*,*

- *every RAF*-stable *structure is also RAF*-preferred.

Another semantics has been defined in [10]: the RAF-*semi-stable* semantics.

Definition 16 ([10]) *Let* $\Gamma = \langle A, K, s, t \rangle$ *be a RAF and* $\mathcal{U} = \langle S, Q \rangle$ *be some structure over it.* $\mathcal{U}$ *is said to be a RAF*-semi-stable *structure iff* $\mathcal{U}$ *is a RAF*-complete *structure such that:*

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

Theorem 2 ([10]) *The following assertions hold:*

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

Theorem 3 ([10]) *Let* $\Gamma = \langle A, K, s, t \rangle$ *be a RAF. If there exists a RAF*-stable *structure, then the RAF*-semi-stable *structures coincide with the RAF*-stable *structures.*

Moreover, 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 RAF-*complete*, RAF-*grounded*, RAF-*preferred* and RAF-*stable* semantics in [6] then in [10] for the RAF-*semi-stable* semantics).

Chapter 3

Computational complexity theory

In this section is given an overview of computational complexity theory. Key notions are put forward nevertheless by sake of time it doesn't go into details. For a more complete view on computational complexity theory see [8] and for deeper explanation on Dung's argumentation framework complexities see [12].

3.1 Principles

Computational complexity theory is a field of computer science whose purpose is to cluster computational problems into "complexity classes". Problems are gathered according to some criterion on the resources required to solve them. Generally the measure used to differentiate them is the time (*i.e.* the number of steps taken by an algorithm) needed or the space (*i.e.* the amount of memory) needed to solve them, but a clustering could be based on any other resource criterion. In this report, we will consider only time complexity classes.

We say that a problem $\mathscr{P}$ belongs to the complexity class $\mathscr{C}$ (or $\mathscr{P}$ has complexity $\mathscr{C}$) if there exists an algorithm that solve $\mathscr{P}$ satisfying the resource requirements of $\mathscr{C}$. Basically, the more a problem requires resources the more it will be considered has difficult.

In order to rank them in a fair way and so, form coherent complexity classes, problems are considered in their generic form. This means that the comparison is not made on specific "problem instances" (*i.e.* the problem applied to concrete data inputs). The resource requirements are expressed according to the problem "input size".

Nevertheless, considering problem's generic form is not sufficient for a proper comparison. Indeed, one can say that solving a given problem on such or such machines (that differ for example on their software or hardware architecture) would induce different resource requirements. In order to fix this issues, in computational complexity theory, we consider that algorithms are executed on some standard "model of computation", such as the so-called *"(Deterministic) Turing Machine"* introduce by Alan Turing in [17].

For the sake of brevity, we will not explain in details how it works but simply give the intuition of it. The *Turing Machine* is an abstract model of computational machine. It is composed of a *tape* on which *symbols* (0 or 1) can read and written by an *head* that can move the tape left and right one *cell* at a time. An algorithm written for a *Turing Machine* is simply a set of *transition* going through some so-called "states". A state indicates what to do given the symbol read on the current tape cell. What is initially written on the tape corresponds to the input (*i.e.* an encoding of it using the *Turing Machine* symbols). Given an input, the number of steps made by a *Turing Machine* to execute an algorithm is used as a time measurement and the number of cells used on the tape is used as a space measurement.

For a particular problem the input size could be expressed with a more understandable measure than the encoding size. For example for AFs problems, we can use the number of arguments of a given AF or the number of attacks.

Now, saying that "a problem $\mathscr{P}_1$ has a lower time complexity than another problem $\mathscr{P}_2$" means "for inputs of size n there exists an algorithm that solve $\mathscr{P}_1$ with fewer steps than any other algorithms that solve $\mathscr{P}_2$". Notice that, is taken into account the number of steps for the worst possible case of input of size n (*i.e.* the input of size n that induces the most transitions to solve the problem). Finally, it is the asymptotic behaviour of $\mathscr{P}_1$ and $\mathscr{P}_2$ as n grows that is considered.

The comparison method being fair, problems can now be grouped in suitable complexity classes. Those correspond basically to different orders of magnitude of steps.

Now let consider the complexity classes of so called "decision problems".

3.2 Decision problem theory

A decision problem is a type of computation problem that has for output a boolean. That is, given an input the solution of the problem is whether "yes" (equivalently true or 1) or "no" (equivalently false or 0). We said that the problem "accept" or "reject" the input.

One of the most famous decision problem, for its inportance in computational complexity theory is the satisfaction problem (so-called SAT problem). It is defined as following :

"Given a propositional formula ϕ, is ϕ satisfiable [1] *?"*

Decision problems are probably the most studied type of computational problems. Over the decades a lot complexity classes and hierarchies between them have been established. Let consider some interesting complexity classes for AF problems.

3.2.1 Decision time complexity classes

3.2.1.1 Polynomial time: P and L

The polynomial time class P regroups computational problems for which there exists an algorithm that solve them in a number of steps that is polynomially related to the size of the input. Problems belonging to this class are considered as "easy" or "tractable".

P has a subclass called "logarithmic space" denoted by L that regroups the problems of P that require an amount of space (excluded the input and the output) that is logarithmically related to the size of the input.

3.2.1.2 Non-deterministic polynomial time: NP

The non-deterministic polynomial time class NP rely on the notion of *witness*.

Given an input x, a witness of x can be seen as a potential *proof by example* that the answer of the decision problem is positive for x. Let illustrate this considering the SAT problem. Given a propositional formula ϕ, a witness of SAT for ϕ is an interpretation of ϕ, *i.e.* a value assignation of the propositional variables of ϕ. Given an input x, a *valid witness* is a valid *proof by example* that the decision problem accepts x.

A problem $\mathscr{P}$ is in NP if and only if:

[1] A propositional formula is said to be satisfiable if there exists a model of it, that is a value (true or false) assignation of its propositional variables for which ϕ is true."

1. for all instance input x, x has a number of potential witnesses polynomial *w.r.t.* $|x|$ (the size of x),
2. each witness of a given input can be verified in a polynomial number of steps *w.r.t.* $|x|$,
3. given an input x, $\mathscr{P}$ accepts x if and only if x has a valid witness.

As an example, the SAT problem is in NP.

NP can also be defined as the set of problems which can be solved in polynomial time on a *non-deterministic Turing Machine*.

The difference between a *Deterministic Turing Machine* and a *non-deterministic* one is that for each step several transitions are possible simultaneously. To illustrate this, one can imagine that at each step a new *Deterministic Turing Machine* could be added (a copy of the machine in its current state) for the problem solving. While a *Deterministic Turing Machine* follows a single computation path, a *non-deterministic Turing Machine* one follow a computation tree. A given decision problem accepts x if there exists a non deterministic algorithm such that at least one computation branch followed by the *non-deterministic Turing Machine* accepts the input.

NP problems are thus computational problems for which there exists a non-deterministic algorithm that can solve them on *non-deterministic Turing Machine* and doing so, following a computation tree having a polynomial depth and a number of leaves relative to the input size.

NP is thus the class of computational problems for which a solution (a proposed *proof by example*) can be verified easily. Although there is no formal proof (at the time of writing) that $\mathtt{P} \neq \mathtt{NP}$, we will consider that this inequation holds in the following as it is the standard assumption.

3.2.1.3 The coNP class

The coNP class is class regrouping the complement problems of those of NP. As for NP, coNP rely on same notion of witness and the same witness properties, *i.e.* for each input x the number of witnesses is polynomial *w.r.t.* $|x|$ and each witness of x is verifiable in an amount of steps polynomial *w.r.t.* $|x|$. The difference between NP and coNP is that coNP regroups the decision problems for which we want all the witnesses for a certain property to be invalid.

As illustration, the coNP problem relative to the SAT problem is the following one:

"Given a propositional formula ϕ, is ϕ unsatisfiable [2] ?"

Here the property of interest is the satifiability of ϕ. UNSAT will accept ϕ if and only if no witness of ϕ (*i.e.* no value assignation of its propositional variables) makes ϕ satisfied.

3.2.1.4 The polynomial-time hierarchy

The notion of "*oracle*" is very important to understand what is the polynomial-time hierarchy. An *oracle* is a black-box abstract machine that can solve a problem of a certain complexity class in one single step. Complexity classes can be expressed via this notion.

Given a problem $\mathscr{P}$, we say that $\mathscr{P}$ is in the complexity class $\mathscr{C}^{\mathscr{D}}$, if there exists an algorithm solving $\mathscr{P}$ with a complexity $\mathscr{C}$ and calling an oracle that solves in one operation a sub-problem of complexity class $\mathscr{D}$.

As an example, let consider the $\exists_2$QBF problem. Let ϕ be a propositional formula over the set of propositional variables Ω. Let $v_1 \subset \Omega$ and $v_2 \subset \Omega$ be two subsets of propositional formula such that $\{v_1, v_2\}$ is a partition[3] of Ω. The $\exists_2$QBF is the following decision problem:

"$\exists v_1$ such that $\forall v_2$, ϕ is true?"

Which means:

"Does there exist a valuation of the variables of v_1 such that for all valuations of the variables of v_2, ϕ is true?"

Let propose a non-deterministic algorithm to solve the $\exists_2$QBF problem. Let $\mathscr{O}$ be an oracle witnessing that a given propositional formula is valid[4]. $\mathscr{O}$ is in `coNP`. Indeed, to decide if ϕ is valid is equivalent to decide if $\neg\phi$ is unsatisfiable. The algorithm $\mathscr{A}$ that non-deterministically guesses a valuation of v_1 and then verifies if for all valuations of v_2 the combined valuations (of v_1 and v_2) are models of ϕ, can be viewed an `NP` algorithm using $\mathscr{O}$ as oracle. As $\mathscr{A}$ solves $\exists_2$QBF, we have $\exists_2$QBF belonging to the class $\mathtt{NP}^{\mathtt{coNP}}$.

The polynomial hierarchy, denoted by PH, is a hierarchy of complexity classes defined by oracles defined as following:

[2] A propositional formula is said to be satisfiable *iff* there exists a model of it.

[3] $\Omega = v_1 \cup v_2$ and $v_1 \cap v_2 = \varnothing$.

[4] A formula ϕ is said to be valid if all valuations of its propositional variables are models of ϕ, *i.e.* ϕ is always true.

- $\Sigma_0^{\mathtt{P}} = \Pi_0^{\mathtt{P}} = \Theta_0^{\mathtt{P}} = \mathtt{P}$
- $\Theta_{k+1}^{\mathtt{P}} = \mathtt{P}^{\Sigma_k^{\mathtt{P}}}$
- $\Sigma_{k+1}^{\mathtt{P}} = \mathtt{NP}^{\Sigma_k^{\mathtt{P}}}$
- $\Pi_{k+1}^{\mathtt{P}} = \mathtt{coNP}^{\Sigma_k^{\mathtt{P}}}$

The polynomial hierarchy is the union of all these complexity classes:

$$\mathrm{PH} \quad = \quad \bigcup_{k=0}^{\infty} \Sigma_k^{\mathtt{P}} \quad = \quad \bigcup_{k=0}^{\infty} \Pi_k^{\mathtt{P}}$$

Figure 3.1 illustrates this hierarchy.

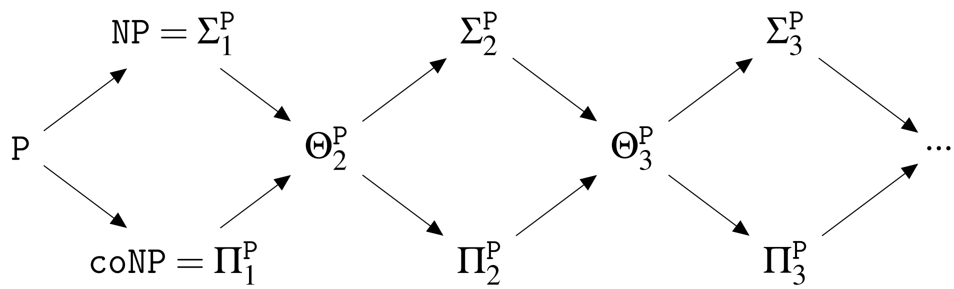


Figure 3.1: Polynomial hierarchy

Notice that calling a polynomial oracle from a non-deterministic algorithm doesn't add any complexity. As a consequence, we especially have $\mathtt{NP}^{\mathtt{P}} = \mathtt{NP}$.

Notice also that using as oracle, in an algorithm, a $\mathtt{NP}$ based oracle or $\mathtt{coNP}$ based oracle of same class *level* (*i.e.* $\Sigma_k^{\mathtt{P}}$ or $\Pi_k^{\mathtt{P}}$ for a given level k) doesn't matter. Indeed the answer of one of these oracles can be switched to correspond to the one solving the complementary problem.

As a consequence, we especially have $\mathtt{NP}^{\mathtt{coNP}} = \mathtt{NP}^{\mathtt{NP}} = \Sigma_2^{\mathtt{P}}$. And so, we have $\exists_2$QBF belonging to $\Sigma_2^{\mathtt{P}}$.

3.2.1.5 The difference class: $\mathtt{DP}$

The so-called "difference class" denoted by $\mathtt{DP}$ is a kind of conjunction of the $\mathtt{NP}$ and $\mathtt{coNP}$ classes. A problem $\mathscr{P}$ belongs to $\mathtt{DP}$ if and only if it is composed of two sub-problems, $\mathscr{P}_1$ belonging to $\mathtt{NP}$ and $\mathscr{P}_2$ belonging to $\mathtt{coNP}$, and for all input x, x is accepted by $\mathscr{P}$ if and only if x is accepted by $\mathscr{P}_1$ and $\mathscr{P}_2$.

As an illustration, the SAT-UNSAT problem belongs to $\mathtt{DP}$. It is defined as following:

"Given a couple propositional formulas $\langle \phi, \psi \rangle$, is ϕ satisfiable and ψ unsatisfiable?"

Following the polynomial hierarchy introduced in the previous section, the DP-hierarchy is defined as following :

$$\mathtt{DP}_k = \Sigma_k^{\mathtt{P}} \wedge \Pi_k^{\mathtt{P}}, \quad \text{with } k \in [\![1, +\infty[\![$$

Notice that "$\wedge$" means "conjunction of problems" as explained above. It is not the intersection of sets of problems.

3.2.2 Problem reduction, completeness and hardness

3.2.2.1 Problem reduction

Let $\mathscr{P}_1$ and $\mathscr{P}_2$ be to decision problems. We denote by $I_{\mathscr{P}_1}$ and $I_{\mathscr{P}_2}$ the sets of all the instances of $\mathscr{P}_1$ and $\mathscr{P}_2$. Let $f : I_{\mathscr{P}_1} \to I_{\mathscr{P}_2}$ be an *efficient*[5] procedure that transforms any instance of $\mathscr{P}_1$ into one instance of $\mathscr{P}_2$ such that for all $x \in I_{\mathscr{P}_1}$, $\mathscr{P}_1$ accepts x *iff* $\mathscr{P}_2$ accepts $f(x)$.

If such a procedure exists, it means that any algorithm solving $\mathscr{P}_2$ could be used to solve $\mathscr{P}_1$ by firstly converting $\mathscr{P}_1$ instances into $\mathscr{P}_2$ ones.

Now if it holds that $\mathscr{P}_2$ is in some complexity class $\mathscr{C}$, it means that $\mathscr{P}_1$ is also in $\mathscr{C}$ considering that f is an efficient problem transformer. Likewise if $\mathscr{P}_2$ is not in $\mathscr{C}$, then $\mathscr{P}_1$ is not in $\mathscr{C}$.

In computational complexity theory, polynomial reduction are considered as efficient. Polynomial reductions are thus applicable to problems in P or complexity classes above. We denote by $\mathscr{P}_1 \leq_{\mathtt{P}} \mathscr{P}_2$ the relation expressing that $\mathscr{P}_1$ is polynomially reducible to $\mathscr{P}_2$, and by $\mathscr{P}_1 \leq_{\mathtt{P}}^{f} \mathscr{P}_2$ that the relation holds by using f. Usually, we use polynomial reductions in P while studying problems in NP and harder complexity classes, and *log-space reductions*, that is procedures belonging to L (denoted by $\leq_{\mathtt{L}}$), while studying complexity classes within P.

3.2.2.2 Completeness and hardness

We consider that a problem is hard for a certain class $\mathscr{C}$ if an efficient algorithm solving it could be used to efficiently solve, by mean of reductions, all the problems in $\mathscr{C}$. It is formally defined as following: let $\mathscr{P}_1$ be a problem of complexity

[5]The complexity of f should be "*easy*" compared to the complexity of solving $\mathscr{P}_1$ or $\mathscr{P}_2$.

class $\mathscr{C}$. $\mathscr{P}_1$ is said to be hard *w.r.t.* $\mathscr{C}$, denoted by $\mathscr{C}$*-hard*, if:

$$\forall \mathscr{P}_2 \in \mathscr{C},\ \mathscr{P}_2 \leq_{\mathrm{P}} \mathscr{P}_1$$

A problem $\mathscr{P}$ is said to be complete for $\mathscr{C}$, denoted by $\mathscr{C}$*-c*, if $\mathscr{P} \in \mathscr{C}$ and $\mathscr{P}$ is $\mathscr{C}$*-hard*.

Chapter 4

Decision problems in Abstract Argumentation

4.1 Problems definition

As mentioned in Section 3.2, in *computational complexity theory*, a *decision problem* is a problem that can be posed as a yes-no question, given some input. Here bellow is a list of interesting abstract argumentation ones expressed for Dung's Abstract Argumentation Framework.

Definition 17 (Decision Problems in Abstract Argumentation)

- Credulous Acceptance $AF\text{-}Cred_{\sigma}$: *Given an AF* $\Gamma = \langle A, K \rangle$ *and an argument* $a \in A$. *Is a contained in some* $S \in \sigma(\Gamma)$ *?*

- Skeptical Acceptance $AF\text{-}Skep_{\sigma}$: *Given an AF* $\Gamma = \langle A, K \rangle$ *and an argument* $a \in A$. *Is a contained in each* $S \in \sigma(\Gamma)$ *?*

- Verification of an extension $AF\text{-}Ver_{\sigma}$: *Given an AF* $\Gamma = \langle A, K \rangle$ *and a set of arguments* $S \subseteq A$. *Is* $S \in \sigma(\Gamma)$ *?*

- Existence of an extension $AF\text{-}Exists_{\sigma}$: *Given an AF* $\Gamma = \langle A, K \rangle$. *Is* $\sigma(\Gamma) \neq \varnothing$ *?*

- Existence of a non-empty extension $AF\text{-}Exists_{\sigma}^{\neg\varnothing}$: *Given an AF* $\Gamma = \langle A, K \rangle$. *Does there exist a set* $S \neq \varnothing$ *such that* $S \in \sigma(\Gamma)$ *?*

- Uniqueness of a solution $AF\text{-}Unique_\sigma$: *Given an AF* $\Gamma = \langle A, K \rangle$. *Is there a unique set* $S \in \sigma(\Gamma)$, *i.e.* $\sigma(\Gamma) = \{S\}$ *?*

In Sections 4.3 and 4.4 those decision problems will be redefined to match AFRA and RAF specificities but the idea behind them stays the same.

4.2 Complexities in Dung's Abstract Argumentation

Table 4.1 gives the complexity class of the mentioned decision problems for the *grounded*, *complete*, *preferred* and *stable* semantics.

The table is the result of numerous works (see [13] for a synthesis of these works).

σ	*AF-*					
	$Cred_\sigma$	$Skep_\sigma$	Ver_σ	$Exists_\sigma$	$Exists_\sigma^{\neg\emptyset}$	$Unique_\sigma$
Grounded	P-c	P-c	P-c	trivial	in L	trivial
Complete	NP-c	P-c	in L	trivial	NP-c	coNP-c
Preferred	NP-c	Π_2^{P}-c	coNP-c	trivial	NP-c	coNP-c
Stable	NP-c	coNP-c	in L	NP-c	NP-c	DP-c
Semi-stable	Σ_2^{P}-c	Π_2^{P}-c	coNP-c	trivial	NP-c	in Θ_2^{P}

Table 4.1: Complexities of Dung's Abstract Framework

4.3 Complexities in AFRA

In this section we are going to present AFRA decision problems and study their complexity. Firstly, we give a definition of them, secondly, we extend Proposition 1 to encompass the case of the *semi-stable* semantics, thirdly, we show that there exists polynomial reductions from AFRAs to AFs and vice versa, and finally we use these reductions to establish all the complexities in AFRAs for decision problems and for the semantics we are interested in.

4.3.1 AFRA decision problems

As mentioned earlier, here below are the redefinitions of the AF decision problems we are interested in.

Definition 18 (Decision Problems in AFRA)

- Credulous Acceptance *AFRA-Cred*$_\sigma$*: Given an AFRA* $\Gamma = \langle A, K\rangle$ *and an element* $x \in A \cup K$*. Is* x *contained in some* $S \in \sigma(\Gamma)$ *?*

- Skeptical Acceptance *AFRA-Skep*$_\sigma$*: Given an AF* $\Gamma = \langle A, K\rangle$ *and an element* $x \in A \cup K$*. Is* x *contained in each* $S \in \sigma(\Gamma)$ *?*

- Verification of an extension *AFRA-Ver*$_\sigma$*: Given an AFRA* $\Gamma = \langle A, K\rangle$ *and a set of elements* $S \subseteq A \cup K$*. Is* $S \in \sigma(\Gamma)$ *?*

- Existence of an extension *AFRA-Exists*$_\sigma$*: Given an AFRA* $\Gamma = \langle A, K\rangle$*. Is* $\sigma(\Gamma) \neq \varnothing$ *?*

- Existence of a non-empty extension *AFRA-Exists*$_\sigma^{\neg\varnothing}$*: Given an AFRA* $\Gamma = \langle A, K\rangle$*. Does there exist a set* $S \neq \varnothing$ *such that* $S \in \sigma(\Gamma)$ *?*

- Uniqueness of a solution *AFRA-Unique*$_\sigma$*: Given an AFRA* $\Gamma = \langle A, K\rangle$*. Is there a unique set* $S \in \sigma(\Gamma)$*, i.e.* $\sigma(\Gamma) = \{S\}$ *?*

In the discussion section of [12], the authors give the intuition that complexities in AFRA are similar as complexities in AF arguing that AFRA extend AF. Since AFRA are not a conservative generalization of AF,[1] this intuition needs a more formal proof and it is that we propose here, giving the corresponding polynomial transformations.

4.3.2 Polynomial reduction: AFRAs and AFs

Let consider the functions `Afra2Af` that transforms an AFRA into an AF and `Af2Afra` that transforms an AF into an AFRA. Note that the first one has been proposed in Definition 19 of [1] (see Definition 10).

Notice that we denote in the following the set of all possible AFRAs by "Ω_{afra}" and the set of all possible AFs by "Ω_{af}".

Definition 19 ([1]) *Let* `Afra2Af` $: \Omega_{afra} \rightarrow \Omega_{af}$ *be the function transforming an AFRA into an AF.* `Afra2Af` *is defined as following:*

$$\forall\, \Gamma = \langle A, K\rangle \in \Omega_{afra},\ \texttt{Afra2Af} : \Gamma \mapsto \widetilde{\Gamma} = \langle \widetilde{A}, \widetilde{K}\rangle$$

[1]The correspondence exists only when we consider the semantics level but it is not satisfied when we consider for instance the admissibility.

With: $\widetilde{A} = A \cup K$ *and* $\widetilde{K} = \{(a,b)|(a,b) \in (A \cup K)^2$ *and* $a \rightarrow_K b\}$

Notice that, in the definition of $\widetilde{K}$, the element a is in fact an attack, and so belongs to K, due to the definition of $\rightarrow_K$.

The definition of `Af2Afra` that transforms an AF into an AFRA is trivial since an AFRA without higher-order attacks is an AF. So it is enough to name the attacks of the AF in order to obtain an AFRA.

Definition 20 *Let* $\texttt{Af2Afra} : \Omega_{af} \rightarrow \Omega_{afra}$ *be the function transforming an AF into an AFRA.* `Af2Afra` *is defined as following:*

$$\forall \Gamma = \langle A, K \rangle \in \Omega_{af},\ \texttt{Af2Afra} : \Gamma \mapsto \Gamma' = (A', K')$$

With: $A' = A$ *and* $K' = K$

Note that, following the previous definition, there cannot exist direct defeat between two attacks in the AFRA obtained by `Af2Afra`.

Using the function `Af2Afra`, the following propositions hold.

Proposition 4 *Let* $\Gamma = \langle A, K \rangle$ *be an AF and* $\Gamma' = \texttt{Af2Afra}(\Gamma)$ *be an AFRA (with* $\Gamma' = \langle A', K' \rangle$*). Let* S *be a subset of* Γ*.*

S is conflict-free *in* Γ *iff* $S \cup \{\alpha \in K$ *s.t.* $s(\alpha) \in S\}$ *is AFRA-*conflict-free *in* Γ'.

Proof. ***Assertion 1:*** *S is* conflict-free *in* Γ *implies that* $S' = S \cup \{\alpha \in K$ *s.t.* $s(\alpha) \in S\}$ *is AFRA-*conflict-free *in* Γ'.

Assume that it is false. So there exist x and $y \in S'$ s.t. $x \rightarrow_K y$. So x is an attack whose source $s(x) \in S$ and we have two cases:

- *either $y \in A'$, so $y \in S$ (since $y \in S'$) and, by definition of* `Af2Afra`*, $(s(x), y) \in K$. So there are in S two arguments $s(x)$ and y that are in conflict. Contradiction.*
- *either $y \in K'$, and $s(y) \in S$ (since $y \in S'$). Since $x \rightarrow_K y$ and since there is no higher-order attack in Γ', then $x \rightarrow_K s(y)$. That means that there exists in Γ an attack $(s(x), s(y))$. So there are in S two arguments $s(x)$ and $s(y)$ that are in conflict. Contradiction with S* conflict-free.

Assertion 2: *$S' = S \cup \{\alpha \in K$ s.t. $s(\alpha) \in S\}$ is AFRA*-conflict-free *in Γ' implies that S is* conflict-free *in Γ.*

Assume that it is false. So there exists x and $y \in S$ s.t. $\alpha = (x,y) \in K$. So in Γ' there exists $\alpha \to_K y$ with $s(\alpha) = x \in S$. So $\alpha \in S'$. Moreover, since $y \in S$ then $y \in S'$. And so there are in S' two elements y and α that are in conflict. Contradiction with S' AFRA-conflict-free . ■

Proposition 5 *Let $\Gamma = \langle A, K \rangle$ be an AF and $\Gamma' =$ `Af2Afra`(Γ) be an AFRA (with $\Gamma' = \langle A', K' \rangle$). Let S be a subset of Γ.*

$Def(S) \cup \{\alpha \in K$ s.t. $s(\alpha) \in Def(S)\} =$ AFRA-$Def(S \cup \{\alpha \in K$ s.t. $s(\alpha) \in S\})$.

PROOF. *Notation: $S' = S \cup \{\alpha \in K$ s.t. $s(\alpha) \in S\}$*

Assertion 1: *$Def(S) \cup \{\alpha \in K$ s.t. $s(\alpha) \in Def(S)\} \subseteq$ AFRA-$Def(S')$.*

Two cases are possible:

- *Consider $x \in Def(S)$. So x is an argument in Γ. Since $x \in Def(S)$, $\exists \alpha = (a,x) \in K$ s.t. $a \in S$. So we have in Γ' the defeat: $\alpha \to_K x$ with the source of α that belongs to S. So $\alpha \in S'$ and α defeats x in Γ'. So $x \in$ AFRA-$Def(S')$.*
- *Consider now $x \in \{\alpha \in K$ s.t. $s(\alpha) \in Def(S)\}$. So x is an attack in K whose source belongs to $Def(S)$. So $\exists \alpha = (a, s(x)) \in K$ s.t. $a \in S$. So we have in Γ' the defeat: $\alpha \to_K x$ with the source of α that belongs to S. So $\alpha \in S'$ and α defeats x in Γ'. So $x \in$ AFRA-$Def(S')$.*

Assertion 2: *$Def(S) \cup \{\alpha \in K$ s.t. $s(\alpha) \in Def(S)\} \supseteq$ AFRA-$Def(S')$.*

Consider $x \in$ AFRA-$Def(S')$. Two cases are possible:

- *$x \in A'$. x is defeated in Γ', so $\exists \alpha \in K'$ s.t. $\alpha \to_K x$ and $\alpha \in S'$ (so $s(\alpha) \in S$). So we have the following attack in Γ: $s(\alpha)$ attacks x. So, since $s(\alpha) \in S$, we have $x \in Def(S)$. Thus $x \in Def(S) \cup \{\alpha \in K$ s.t. $s(\alpha) \in Def(S)\}$.*
- *$x \in K'$. x is an attack. Since $x \in$ AFRA-$Def(S')$, $\exists \alpha \in K'$ s.t. $\alpha \to_K x$ (indirect defeat) and $\alpha \in S'$ (so $s(\alpha) \in S$). So we have the following attack in Γ: $s(\alpha)$ attacks $s(x)$. So, since $s(\alpha) \in S$, we have $s(x) \in Def(S)$ and so $x \in \{\alpha \in K$ s.t. $s(\alpha) \in Def(S)\}$. Thus $x \in Def(S) \cup \{\alpha \in K$ s.t. $s(\alpha) \in Def(S)\}$.*

■

Proposition 6 *Let* $\Gamma = \langle A, K\rangle$ *be an AF and* $\Gamma' = \texttt{Af2Afra}(\Gamma)$ *be an AFRA (with* $\Gamma' = \langle A', K'\rangle$*). Let S be a subset of* Γ*.*

$Acc(S) \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Acc(S)\} = AFRA\text{-}Acc(S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\})$.

Proof. *Notation:* $S' = S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$

Assertion 1: $Acc(S) \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Acc(S)\} \subseteq AFRA\text{-}Acc(S')$.

Two cases are possible:

- *Consider* $x \in Acc(S)$*. So x is an argument in* Γ*. If it is unattacked in* Γ *then it is undefeated in* Γ' *and so* $x \in AFRA\text{-}Acc(S')$*. If it is attacked in* Γ*, then, since* $x \in Acc(S)$*,* $\forall \alpha = (a, x) \in K$*,* $\exists \beta = (b, a) \in K$ *s.t.* $b \in S$*. So we have in* Γ' *the sequence of defeats:* $\beta \to_K \alpha \to_K x$ *with the source of* β *that belongs to S. So* $\beta \in S'$ *and* β *defends x against* α *in* Γ'*. So* $x \in AFRA\text{-}Acc(S')$*.*

- *Consider now* $x \in \{\alpha \in K \text{ s.t. } s(\alpha) \in Acc(S)\}$*. So x is an attack in K whose source belongs to* $Acc(S)$*. If this source is unattacked in* Γ *then x is undefeated in* Γ' *and so* $x \in AFRA\text{-}Acc(S')$*. If this source is attacked in* Γ *then, since it belongs to* $Acc(S)$*,* $\forall \alpha = (a, s(x)) \in K$*,* $\exists \beta = (b, a) \in K$ *s.t.* $b \in S$*. So we have in* Γ' *the sequence of defeats:* $\beta \to_K \alpha \to_K x$ *with the source of* β *that belongs to S. So* $\beta \in S'$ *and* β *defends x against* α *in* Γ'*. So* $x \in AFRA\text{-}Acc(S')$*.*

Assertion 2: $Acc(S) \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Acc(S)\} \supseteq AFRA\text{-}Acc(S')$.

Consider $x \in AFRA\text{-}Acc(S')$*. Two cases are possible:*

- $x \in A'$*. If x is undefeated in* Γ' *then it is unattacked in* Γ *and so* $x \in Acc(S)$*. If x is defeated in* Γ' *then, since* $x \in AFRA\text{-}Acc(S')$*,* $\forall \alpha \in K'$ *s.t.* $\alpha \to_K x$*,* $\exists \beta \in S'$ *s.t.* $\beta \to_K \alpha$ *(indirect defeat). Note that* $s(\beta) \in S$*. So in* Γ*, we have the following sequence of attacks:* $s(\beta)$ *attacks* $s(\alpha)$ *that attacks x, for each* α *attacking x. So, since* $s(\beta) \in S$*, we have* $x \in Acc(S)$*. In both cases, we have* $x \in Acc(S) \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Acc(S)\}$*.*

- $x \in K'$. *x is an attack. If x is undefeated in* Γ' *that means that its source* $s(x)$ *is unattacked in* Γ *and so* $s(x) \in Acc(S)$*; in this case,* $x \in \{\alpha \in K \text{ s.t. } s(\alpha) \in Acc(S)\}$. *If x is defeated in* Γ' *then, since* $x \in$ *AFRA-*$Acc(S')$, $\forall \alpha \in K'$ *s.t.* $\alpha \rightarrow_K x$ *(indirect defeat),* $\exists \beta \in S'$ *s.t.* $\beta \rightarrow_K \alpha$ *(indirect defeat). Note that* $s(\beta) \in S$. *So in* Γ, *we have the following sequence of attacks:* $s(\beta)$ *attacks* $s(\alpha)$ *that attacks* $s(x)$, *for each* α *attacking the source of x. So, since* $s(\beta) \in S$, *we have* $s(x) \in Acc(S)$ *and so* $x \in \{\alpha \in K \text{ s.t. } s(\alpha) \in Acc(S)\}$. *In both cases, we have* $x \in Acc(S) \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Acc(S)\}$.

■

Proposition 7 *Let* $\Gamma = \langle A, K \rangle$ *be an AF and* $\Gamma' = \texttt{Af2Afra}(\Gamma)$ *be an AFRA (with* $\Gamma' = \langle A', K' \rangle$*). Let S be an extension of* Γ.

For each semantics $\sigma \in \{$complete, semi-stable, stable, preferred, grounded$\}$, *we have: S is a* σ*-extension of* Γ *iff* $S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$ *is a* σ*-extension of* Γ'.

PROOF. *Notation:* $S' = S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$

Assertion 1: *S is a complete-extension of* Γ *iff* S' *is a complete-extension of* Γ'.

S is a complete-extension of Γ *iff S is* conflict-free *and* $S = Acc(S)$.

Following Proposition 4, S is conflict-free *iff* $S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$ *is AFRA*-conflict-free.

Following Proposition 6 and $S = Acc(S)$, *we have AFRA-*$Acc(S') = Acc(S) \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Acc(S)\} = S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} = S'$. *Moreover, if we consider that* $S' =$ *AFRA-*$Acc(S')$, *and using Proposition 6, we have* $S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} = Acc(S) \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Acc(S)\}$ *and so* $S = Acc(S)$.

Assertion 2: *S is a preferred-extension of* Γ *iff* S' *is a preferred-extension of* Γ'.

That equivalence follows directly from Assertion 1 and the fact that a preferred extension is a maximal complete extension.

Assertion 3: *S is a grounded-extension of* Γ *iff* S' *is a grounded-extension of* Γ'.

That equivalence follows directly from Assertion 1 and the fact that the grounded extension is the minimal complete extension.

Assertion 4: *S is a stable-extension of* Γ *iff* S' *is a stable-extension of* Γ'.

Following Proposition 4, S is conflict-free *in* Γ *iff* $S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$ *is* *AFRA*-conflict-free *in* Γ'.

Then, we have:

$S \cup Def(S) = A$

iff $S \cup Def(S) \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} = A \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$

iff $S' \cup Def(S) = A \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$

iff $S' \cup Def(S) \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Def(S)\} =$
$A \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Def(S)\}$

iff (since Proposition 5) $S' \cup AFRA\text{-}Def(S') =$
$A \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Def(S)\}$

iff $S' \cup AFRA\text{-}Def(S') = A \cup K$

The last equivalence uses either the fact that $S \cup Def(S) = A$ *(in the sense of* $\Rightarrow$*), or the fact that, since* $S' \cup AFRA\text{-}Def(S') = A \cup K$ *and following the definition of* S'*, the source of any attack* $s(\alpha) \in A'$ *belongs either to S or to* $Def(S)$ *(in the sense of* $\Leftarrow$*).*

So Assertion 4 is satisfied.

Assertion 5: *S is an semistable-extension of* Γ *iff* S' *is an semistable-extension of* Γ'.

Following Assertion 1, S is a complete-extension of Γ *iff* $S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$ *is a complete-extension of* Γ'.

Moreover, we have:

$S \cup Def(S)$ *is* $\subseteq$*-maximal wrt A*

iff $S \cup Def(S) \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$ *is* $\subseteq$*-maximal wrt* $A \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$

iff $S' \cup Def(S)$ *is* $\subseteq$*-maximal wrt* $A \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$

iff $S' \cup Def(S) \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Def(S)\}$ *is* $\subseteq$*-maximal wrt*
$A \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Def(S)\}$

iff (since Proposition 5) $S' \cup AFRA\text{-}Def(S')$ *is* $\subseteq$*-maximal wrt*
$A \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in Def(S)\}$

iff $S' \cup AFRA\text{-}Def(S')$ *is* $\subseteq$*-maximal wrt* $A \cup K$

So Assertion 4 is satisfied.

■

Notice that all two functions: `Afra2Af` and `Af2Afra` are trivially polynomial. More than that they are log-space.

4.3.3 Complexity results

So using the polynomial transformations described in the previous section, the following propositions hold.

The two first propositions concern the links between decision problems for an AF and decision problem for its corresponding AFRA.

Proposition 8 *Let* $\Gamma = \langle A, K \rangle$ *be an AF and* $\Gamma' = \texttt{Af2Afra}(\Gamma)$ *be an AFRA (with* $\Gamma' = \langle A', K' \rangle$*). Let* $a \in A$ *be an argument in* Γ *and an element in* Γ'*, following the definition of* `Af2Afra`*. Let* S *be an extension of* Γ.

For each semantics $\sigma \in \{\text{complete, semi-stable, stable, preferred, grounded}\}$*, we have:*

1. *AF-Cred*$_\sigma$ *accepts* (Γ, a) *iff AFRA-Cred*$_\sigma$ *accepts* (Γ', a).
2. *AF-Skep*$_\sigma$ *accepts* (Γ, a) *iff AFRA-Skep*$_\sigma$ *accepts* (Γ', a).
3. *AF-Ver*$_\sigma$ *accepts* (Γ, S) *iff AFRA-Ver*$_\sigma$ *accepts* $(\Gamma', S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$.
4. *AF-Exists*$_\sigma$ *accepts* Γ *iff AFRA-Exists*$_\sigma$ *accepts* Γ'.
5. *AF-Exists*$_\sigma^{\neg\emptyset}$ *accepts* Γ *iff AFRA-Exists*$_\sigma^{\neg\emptyset}$ *accepts* Γ.
6. *AF-Unique*$_\sigma$ *accepts* Γ *iff AFRA-Unique*$_\sigma$ *accepts* Γ'.

PROOF. ***Assertion 1:*** *AF-Cred*$_\sigma$ *accepts* (Γ, a) *iff AFRA-Cred*$_\sigma$ *accepts* (Γ', a).

AF-Cred$_\sigma$ *accepts* (Γ, a) *iff* $\exists S \in \sigma(\Gamma)$ *s.t.* $a \in S$ *iff (following Proposition 7)* $\exists S' = S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} \in \sigma(\Gamma')$ *s.t.* $a \in S'$ *iff AFRA-Cred*$_\sigma$ *accepts* (Γ', a).

Assertion 2: *AF-Skep*$_\sigma$ *accepts* (Γ, a) *iff AFRA-Skep*$_\sigma$ *accepts* (Γ', a).

AF-Skep$_\sigma$ *accepts* (Γ, a) *iff* $\forall S \in \sigma(\Gamma)$, $a \in S$ *iff (following Proposition 7)* $\forall S' = S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} \in \sigma(\Gamma')$, $a \in S'$ *iff AFRA-Skep*$_\sigma$ *accepts* (Γ', a).

Assertion 3: *AF-Ver*$_\sigma$ *accepts* (Γ, S) *iff AFRA-Ver*$_\sigma$ *accepts* $(\Gamma', S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\}$.

AF-Ver$_\sigma$ *accepts* (Γ, S) *iff* $S \in \sigma(\Gamma)$ *iff (following Proposition 7)* $S' = S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} \in \sigma(\Gamma')$ *iff AFRA-Ver*$_\sigma$ *accepts* $(\Gamma', S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\})$.

Assertion 4: *AF-Exists*$_\sigma$ *accepts* Γ *iff AFRA-Exists*$_\sigma$ *accepts* Γ'.

AF-Exists$_\sigma$ *accepts* Γ *iff* $\exists S \in \sigma(\Gamma)$ *iff (following Proposition 7)* $\exists S' = S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} \in \sigma(\Gamma')$ *iff AFRA-Exists*$_\sigma$ *accepts* Γ'.

Assertion 5: *AF-Exists*$_\sigma^{\neg\varnothing}$ *accepts* Γ *iff AFRA-Exists*$_\sigma^{\neg\varnothing}$ *accepts* Γ'.

AF-Exists$_\sigma^{\neg\varnothing}$ *accepts* Γ *iff* $\exists S \neq \varnothing \in \sigma(\Gamma)$ *iff (following Proposition 7)* $\exists S' = S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} \neq \varnothing \in \sigma(\Gamma')$ *iff AFRA-Exists*$_\sigma^{\neg\varnothing}$ *accepts* Γ'.

Assertion 6: *AF-Unique*$_\sigma$ *accepts* Γ *iff AFRA-Unique*$_\sigma$ *accepts* Γ'.

AF-Unique$_\sigma$ *accepts* Γ *iff* $\exists! S \in \sigma(\Gamma)$ *iff (following Proposition 7)* $\exists! S' = S \cup \{\alpha \in K \text{ s.t. } s(\alpha) \in S\} \in \sigma(\Gamma')$ *iff AFRA-Unique*$_\sigma$ *accepts* Γ'. ■

Proposition 9 *The complexities of AFRA decision problems are at least as hard as AF ones, for the semantics* complete, semi-stable, stable, preferred, grounded.

Proof. *Given that* `Af2Afra` *is a polynomial time, log-space function, then according to Proposition 8, for each semantics* $\sigma \in \{$complete, semi-stable, stable, preferred, grounded$\}$ *we have:*

- *AF-Cred*$_\sigma \leq_L^{\mathtt{Af2Afra}}$ *AFRA-Cred*$_\sigma$
- *AF-Skep*$_\sigma \leq_L^{\mathtt{Af2Afra}}$ *AFRA-Skep*$_\sigma$
- *AF-Ver*$_\sigma \leq_L^{\mathtt{Af2Afra}}$ *AFRA-Ver*$_\sigma$

- $AF\text{-}Exists_\sigma \leq_L^{\texttt{Af2Afra}} AFRA\text{-}Exists_\sigma$
- $AF\text{-}Exists_\sigma^{\neg\varnothing} \leq_L^{\texttt{Af2Afra}} AFRA\text{-}Exists_\sigma^{\neg\varnothing}$
- $AF\text{-}Unique_\sigma \leq_L^{\texttt{Af2Afra}} AFRA\text{-}Unique_\sigma$

■

The two next propositions concern the links between decision problems for an AFRA and decision problems for its corresponding AF.

Proposition 10 *Let* $\Gamma = \langle A, K\rangle$ *be an AFRA and* $\widetilde{\Gamma} = \texttt{Afra2Af}(\Gamma)$ *be an AF (with* $\widetilde{\Gamma} = \langle \widetilde{A}, \widetilde{K}\rangle$*). Let* $a \in A$ *be an element in* Γ *and an argument in* $\widetilde{\Gamma}$*, following the definition of* `Afra2Af`*. Let* S *be an extension of* Γ*.*

For each semantics $\sigma \in \{$complete, semi-stable, stable, preferred, grounded$\}$*, we have:*

1. *AFRA-Cred*$_\sigma$ *accepts* (Γ, a) *iff AF-Cred*$_\sigma$ *accepts* $(\widetilde{\Gamma}, a)$.
2. *AFRA-Skep*$_\sigma$ *accepts* (Γ, a) *iff AF-Skep*$_\sigma$ *accepts* $(\widetilde{\Gamma}, a)$.
3. *AFRA-Ver*$_\sigma$ *accepts* (Γ, S) *iff AF-Ver*$_\sigma$ *accepts* $(\widetilde{\Gamma}, S)$.
4. *AFRA-Exists*$_\sigma$ *accepts* Γ *iff AF-Exists*$_\sigma$ *accepts* $\widetilde{\Gamma}$.
5. *AFRA-Exists*$_\sigma^{\neg\varnothing}$ *accepts* Γ *iff AF-Exists*$_\sigma^{\neg\varnothing}$ *accepts* $\widetilde{\Gamma}$.
6. *AFRA-Unique*$_\sigma$ *accepts* Γ *iff AF-Unique*$_\sigma$ *accepts* $\widetilde{\Gamma}$.

PROOF. *This proof is trivial considering Proposition 1 (given originally in [1]).* ■

Proposition 11 *The complexities of AF decision problems are at least as hard as AFRA ones, for the semantics* complete, semi-stable, stable, preferred, grounded.

PROOF. *Given that* `Afra2Af` *is a polynomial time, log-space function, then according to Proposition 10, for each semantics* $\sigma \in \{$complete, semi-stable, stable, preferred, grounded$\}$ *we have:*

- $AFRA\text{-}Cred_\sigma \leq_L^{\texttt{Afra2Af}} AF\text{-}Cred_\sigma$
- $AFRA\text{-}Skep_\sigma \leq_L^{\texttt{Afra2Af}} AF\text{-}Skep_\sigma$
- $AFRA\text{-}Ver_\sigma \leq_L^{\texttt{Afra2Af}} AF\text{-}Ver_\sigma$
- $AFRA\text{-}Exists_\sigma \leq_L^{\texttt{Afra2Af}} AF\text{-}Exists_\sigma$
- $AFRA\text{-}Exists_\sigma^{\neg\emptyset} \leq_L^{\texttt{Afra2Af}} AF\text{-}Exists_\sigma^{\neg\emptyset}$
- $AFRA\text{-}Unique_\sigma \leq_L^{\texttt{Afra2Af}} AF\text{-}Unique_\sigma$

■

So using the previous propositions, we have:

Proposition 12 *The complexities of AFRA decision problems are the same as AF ones, for the semantics* complete*,* semi-stable*,* stable*,* preferred*,* grounded*, as stated in Table 4.2.*

PROOF. *Given that* `Afra2Af` *and* `Af2Afra` *are polynomial time procedures and that Propositions 9 and 11 holds, then all the complexities are the same.* ■

σ	*AFRA-*					
	$Cred_\sigma$	$Skep_\sigma$	Ver_σ	$Exists_\sigma$	$Exists_\sigma^{\neg\emptyset}$	$Unique_\sigma$
Grounded	P-*c*	P-*c*	P-*c*	trivial	in L	trivial
Complete	NP-*c*	P-*c*	in L	trivial	NP-*c*	coNP-*c*
Preferred	NP-*c*	Π_2^P-*c*	coNP-*c*	trivial	NP-*c*	coNP-*c*
Stable	NP-*c*	coNP-*c*	in L	NP-*c*	NP-*c*	DP-*c*
Semi-stable	Σ_2^P-*c*	Π_2^P-*c*	coNP-*c*	trivial	NP-*c*	in Θ_2^P

Table 4.2: Complexities of AFRA

4.4 Complexities in RAF

4.4.1 RAF decision problems

Definition 21 (Decision Problems in RAF)

- Credulous Acceptance *RAF-Cred*$_\sigma$*: Given an RAF* $\Gamma = \langle A, K, s, t\rangle$ *and an element* $x \in A \cup K$*. Is x contained in some* $\mathcal{U} \in \sigma(\Gamma)$ *?*

- Skeptical Acceptance *RAF-Skep*$_\sigma$*: Given an AF* $\Gamma = \langle A, K, s, t\rangle$ *and an element* $x \in A \cup K$*. Is x contained in each* $\mathcal{U} \in \sigma(\Gamma)$ *?*

- Verification of an extension *RAF-Ver*$_\sigma$*: Given an RAF* $\Gamma = \langle A, K, s, t\rangle$ *and a structure* $\mathcal{U}$*. Is* $\mathcal{U} \in \sigma(\Gamma)$ *?*

- Existence of an extension *RAF-Exists*$_\sigma$*: Given an RAF* $\Gamma = \langle A, K, s, t\rangle$*. Is* $\sigma(\Gamma) \neq \varnothing$ *?*

- Existence of a non-empty extension *RAF-Exists*$_\sigma^{\neg\varnothing}$*: Given an RAF* $\Gamma = \langle A, K, s, t\rangle$*. Does there exist a structure* $\mathcal{U} \neq \varnothing$ *such that* $\mathcal{U} \in \sigma(\Gamma)$ *?*

- Uniqueness of a solution *RAF-Unique*$_\sigma$*: Given an RAF* $\Gamma = \langle A, K, s, t\rangle$*. Is there a unique structure* $\mathcal{U} \in \sigma(\Gamma)$*, i.e.* $\sigma(\Gamma) = \{\mathcal{U}\}$ *?*

In [7], some results about complexities in RAF are given about the *RAF-Cred*$_\sigma$ problem (for *complete*, *preferred* and *stable* semantics) and about the *RAF-Skep*$_\sigma$ problem (for *preferred* and *stable* semantics).

So, we here complete these results.

4.4.2 Polynomial reduction: RAFs and AFs

Let consider the functions `Af2Raf` and `Raf2Af` that transform respectively a RAF into an AF and an AF into a RAF in a way that ensures semantics correspendence. Notice that we denote in the following the set of all possible RAFs by "Ω_{raf}".

The definition of `Af2Raf` that transforms an AF into an RAF is trivial since an RAF without higher-order attacks is an AF. So it is enough to name the attacks of the AF in order to obtain an RAF (as we did in the AFRA case).

Definition 22 *Let* $\texttt{Af2Raf} : \Omega_{af} \to \Omega_{raf}$ *be the function transforming an AF into an RAF.* `Af2Raf` *is defined as following:*

$$\forall\, \Gamma = \langle A, K\rangle \in \Omega_{af},\ \texttt{Af2Raf} : \Gamma \mapsto \Gamma' = \langle A', K', s, t\rangle$$

With: $A' = A$ *and* $K' = K$

Note that, following the previous definition, no attack can be inhibited (as none is a target) in the RAF obtained by `Af2Raf`.

The `Raf2Af` function needs more explanation than the previous one. Let first give its definition and then explain what it does.

Definition 23 *Let* $\texttt{Raf2Af} : \Omega_{raf} \to \Omega_{af}$ *be the function transforming a RAF into an AF.* `Raf2Af` *is defined as following:*

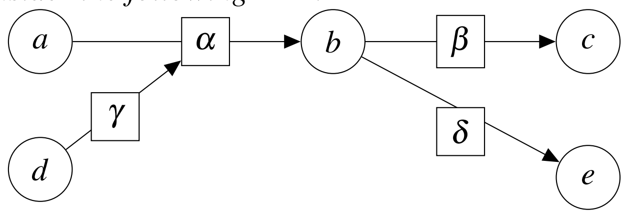
$$\forall \Gamma = \langle A, K, s, t \rangle \in \Omega_{raf}, \ \texttt{Raf2Af} : \Gamma \mapsto \Gamma' = \langle A', K' \rangle$$

$$\begin{aligned}
\textit{With: } A' &= A \cup K \cup Not_A \cup Not_K \cup And_{A,K} \\
K' &= K'_1 \cup K'_2 \cup K'_3 \cup K'_4 \cup K'_5 \\
Not_A &= \{\neg a | a \in A\} \\
Not_K &= \{\neg \beta | \beta \in K\} \\
And_{A,K} &= \{a.\beta | \beta \in K, a = s(\beta)\} \\
K'_1 &= \{(a, \neg a) | a \in A\} \\
K'_2 &= \{(\beta, \neg \beta) | \beta \in K\} \\
K'_3 &= \{(\neg a, a.\beta) | a \in A, s(\beta) = a\} \\
K'_4 &= \{(\neg \beta, a.\beta) | \beta \in K, s(\beta) = a\} \\
K'_5 &= \{(a.\beta, t(\beta)) | \beta \in K, s(\beta) = a\}
\end{aligned}$$

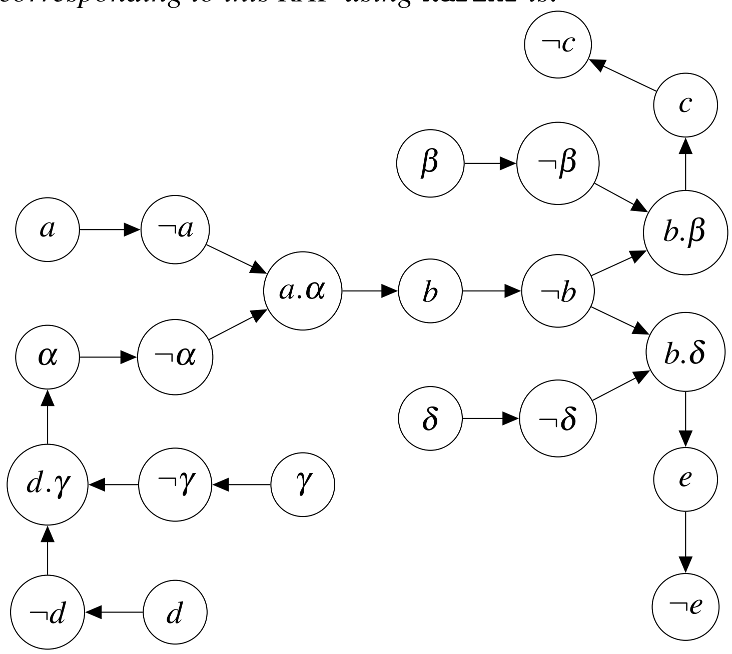
Notice that "$\neg a$", "$\neg \beta$" and "$a.\beta$" are just simple argument names that represent respectively, the negation of the argument a, the negation of the attack β and the conjunction of the attack β with its source a.

This transformation represents, with AFs, the semantics of RAF defeat relation, by mean of additional arguments. Let a be an argument attacking an element b through the attack α in the RAF Γ. Given that to b be defeated by a, α must be valid (non-inhibited) and a accepted (not defeated), we represent this by creating an additional argument named "$a.\alpha$" accepted in Γ' only when both a and α are. To do that we create two others arguments named "$\neg a$" and "$\neg \alpha$". We create an attack going from a to $\neg a$, another going from α to $\neg \alpha$, two others going from $\neg a$ to $a.\alpha$ and $\neg \alpha$ to $a.\alpha$, and finally a last one going from $a.\alpha$ to b. An argument (corresponding to an element of the original RAF) is thus defeated in the resulting AF if and only if there exists a valid attack in the original RAF whose source is accepted.

Example 1 *Consider the following RAF:*



The AF corresponding to this RAF using `Raf2Af` *is:*



Note that the structure $\mathcal{U} = (\{a,b,d\},\{\beta,\gamma,\delta\})$ *is a* σ*-structure for the RAF and this structure after removal of the* $\neg x$ *and the* $x.y$ *elements is a* σ*-extension for the corresponding AF.*

Both `Raf2Af` and `Af2Raf` are polynomial time and log-space functions.

As mentioned earlier, 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 RAF-*complete*, RAF-*grounded*, RAF-*preferred* and RAF-*stable* semantics in [6] then in [10] for the RAF-*semi-stable* semantics).

As it has already been done for `Af2Raf`, let now study the semantics properties of the new introduced transformation: `Raf2Af`.

Let $\Gamma = \langle A, K, s, t\rangle$ be a RAF and $\Gamma' = \texttt{Raf2Af}(\Gamma)$ be an AF (with $\Gamma' = \langle A', K'\rangle$). Let $\mathcal{U} = \langle S, Q\rangle$ be a structure in Γ. We denote by "$\varepsilon_{\mathcal{U}}$" the extension in Γ' corresponding to a structure $\mathcal{U}$, defined by:

$$\begin{aligned}\varepsilon_{\mathcal{U}} = S \cup Q &\cup \{\neg a \in Not_A | a \in \text{RAF-}Def(\mathcal{U})\}\\ &\cup \{\neg\beta \in Not_K | \beta \in \text{RAF-}Inh(\mathcal{U})\}\\ &\cup \{s(\beta).\beta \in And_{A,K} | \beta \in Q, s(\beta) \in S\}\end{aligned}$$

Proposition 13 *Let $\Gamma = \langle A, K, s, t\rangle$ be a RAF and $\Gamma' = \texttt{Raf2Af}(\Gamma)$ be an AF (with $\Gamma' = \langle A', K'\rangle$). Let $\mathcal{U} = \langle S, Q\rangle$ be a structure in Γ. The following properties holds:*

1. *RAF-$Def(\mathcal{U}) \cup$ RAF-$Inh(\mathcal{U}) =$*
$$Def(\varepsilon_{\mathcal{U}}) \setminus \left(\begin{array}{l}\{\neg a \in Not_A | a \in \varepsilon_{\mathcal{U}}\}\\ \cup\{\neg\beta \in Not_K | \beta \in \varepsilon_{\mathcal{U}}\}\\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathcal{U}}) \text{ or } s(\beta) \in Def(\varepsilon_{\mathcal{U}})\}\end{array}\right)$$
2. *RAF-$Acc(\mathcal{U}) = Acc(\varepsilon_{\mathcal{U}}) \setminus \left(\begin{array}{l}\{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\}\\ \cup\{\neg\beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\}\\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\}\end{array}\right)$*

PROOF. ***Assertion 1:*** *RAF-$Def(\mathcal{U}) \cup$ RAF-$Inh(\mathcal{U}) =$*
$$Def(\varepsilon_{\mathcal{U}}) \setminus \left(\begin{array}{l}\{\neg a \in Not_A | a \in \varepsilon_{\mathcal{U}}\}\\ \cup\{\neg\beta \in Not_K | \beta \in \varepsilon_{\mathcal{U}}\}\\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathcal{U}}) \text{ or } s(\beta) \in Def(\varepsilon_{\mathcal{U}})\}\end{array}\right)$$

- ***Step 1:*** *RAF-$Def(\mathcal{U}) \cup$ RAF-$Inh(\mathcal{U}) \subseteq$*
$$Def(\varepsilon_{\mathcal{U}}) \setminus \left(\begin{array}{l}\{\neg a \in Not_A | a \in \varepsilon_{\mathcal{U}}\}\\ \cup\{\neg\beta \in Not_K | \beta \in \varepsilon_{\mathcal{U}}\}\\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathcal{U}}) \text{ or } s(\beta) \in Def(\varepsilon_{\mathcal{U}})\}\end{array}\right)$$
  Let $x \in$ RAF-$Def(\mathcal{U}) \cup$ RAF-$Inh(\mathcal{U})$. There exists thus an attack $\alpha \in Q$ such that $s(\alpha) \in S$ and $t(\alpha) = x$. We have thus:
$$\alpha \in \varepsilon_{\mathcal{U}} \text{ and } s(\alpha) \in \varepsilon_{\mathcal{U}}$$
  As a consequence, following the definition of `Raf2Af`, *we have:*
$$\neg\alpha \in Def(\varepsilon_{\mathcal{U}}) \text{ and } \neg s(\alpha) \in Def(\varepsilon_{\mathcal{U}})$$

And so:

$$s(\alpha).\alpha \in \varepsilon_{\mathcal{U}}$$

Given that $s(\alpha).\alpha \in \varepsilon_{\mathcal{U}}$*, we have so:*

$$x \in Def(\varepsilon_{\mathcal{U}})$$

Given that $x \in A \cup K$*, we have, following the definition of* `Raf2Af`*:* $x \notin (Not_A \cup Not_K \cup And_{A,K})$*. As a consequence, we have:*

$$x \in Def(\varepsilon_{\mathcal{U}}) \setminus (Not_A \cup Not_K \cup And_{A,K})$$

We prove so that: $RAF\text{-}Def(\mathcal{U}) \cup RAF\text{-}Inh(\mathcal{U}) \subseteq$

$$Def(\varepsilon_{\mathcal{U}}) \setminus \left(\begin{array}{l} \{\neg a \in Not_A | a \in \varepsilon_{\mathcal{U}}\} \\ \cup\{\neg \beta \in Not_K | \beta \in \varepsilon_{\mathcal{U}}\} \\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathcal{U}}) \text{ or } s(\beta) \in Def(\varepsilon_{\mathcal{U}})\} \end{array} \right)$$

- ***Step 2:*** $RAF\text{-}Def(\mathcal{U}) \cup RAF\text{-}Inh(\mathcal{U}) \supseteq$

  $$Def(\varepsilon_{\mathcal{U}}) \setminus \left(\begin{array}{l} \{\neg a \in Not_A | a \in \varepsilon_{\mathcal{U}}\} \\ \cup\{\neg \beta \in Not_K | \beta \in \varepsilon_{\mathcal{U}}\} \\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathcal{U}}) \text{ or } s(\beta) \in Def(\varepsilon_{\mathcal{U}})\} \end{array} \right)$$

  Let $x \in Def(\varepsilon_{\mathcal{U}}) \setminus \left(\begin{array}{l} \{\neg a \in Not_A | a \in \varepsilon_{\mathcal{U}}\} \\ \cup\{\neg \beta \in Not_K | \beta \in \varepsilon_{\mathcal{U}}\} \\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathcal{U}}) \text{ or } s(\beta) \in Def(\varepsilon_{\mathcal{U}})\} \end{array} \right)$.

  Let consider four cases: $x \in Not_A$*,* $x \in Not_K$*,* $x \in And_{A,K}$ *and* $x \in A \cup K$*. Let show that the three first cases are impossible.*

  - *Let suppose that* $x \in Not_A$ *with* $x = \neg b$*. Given that* $\neg b \in Def(\varepsilon_{\mathcal{U}})$*, according to the definition of* `Raf2Af`*, we have* $b \in \varepsilon_{\mathcal{U}}$*. As a consequence, we have:* $x \in \{\neg a \in Not_A | a \in \varepsilon_{\mathcal{U}}\}$*, which is a contradiction.*
  - *Let suppose that* $x \in Not_K$ *with* $x = \neg \alpha$*. Given that* $\neg \alpha \in Def(\varepsilon_{\mathcal{U}})$*, according to the definition of* `Raf2Af`*, we have* $\alpha \in \varepsilon_{\mathcal{U}}$*. As a consequence, we have:* $x \in \{\neg \beta \in Not_K | \beta \in \varepsilon_{\mathcal{U}}\}$*, which is a contradiction.*
  - *Let suppose that* $x \in And_{A,K}$ *with* $x = s(\alpha).\alpha$*. Given that* $s(\alpha).\alpha \in Def(\varepsilon_{\mathcal{U}})$*, according to the definition of* `Raf2Af`*, we have thus:* $\neg s(\alpha) \in \varepsilon_{\mathcal{U}}$ *or* $\neg \alpha \in \varepsilon_{\mathcal{U}}$*. And so we have:* $s(\alpha) \in Def(\varepsilon_{\mathcal{U}})$ *or* $\alpha \in Def(\varepsilon_{\mathcal{U}})$*. As a consequence, we have:* $x \in \{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathcal{U}}) \text{ or } s(\beta) \in Def(\varepsilon_{\mathcal{U}})\}$*, which is a contradiction.*

We prove so that: $x \in A \cup K$.

Given that $x \in Def(\varepsilon_{\mathfrak{U}})$*, following the definition of* `Raf2Af`*, there exists thus an argument* $s(\alpha).\alpha \in (\varepsilon_{\mathfrak{U}} \cap And_{A,K})$ *attacking* x*. Given that* $s(\alpha).\alpha \in (\varepsilon_{\mathfrak{U}} \cap And_{A,K})$*, we have following the definition of* $\varepsilon_{\mathfrak{U}}$*:* $s(\alpha) \in S$ *and* $\alpha \in Q$*. As a consequence we have:* $x \in RAF\text{-}Def(\mathfrak{U}) \cup RAF\text{-}Inh(\mathfrak{U})$.

We prove so that: $RAF\text{-}Def(\mathfrak{U}) \cup RAF\text{-}Inh(\mathfrak{U}) \supseteq$

$$Def(\varepsilon_{\mathfrak{U}}) \setminus \left(\begin{array}{l} \{\neg a \in Not_A | a \in \varepsilon_{\mathfrak{U}}\} \\ \cup\{\neg\beta \in Not_K | \beta \in \varepsilon_{\mathfrak{U}}\} \\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathfrak{U}}) \text{ or } s(\beta) \in Def(\varepsilon_{\mathfrak{U}})\} \end{array} \right)$$

Assertion 2: $RAF\text{-}Acc(\mathfrak{U}) = Acc(\varepsilon_{\mathfrak{U}}) \setminus \left(\begin{array}{l} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathfrak{U}})\} \\ \cup\{\neg\beta \in Not_K | \beta \in Def(\varepsilon_{\mathfrak{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathfrak{U}}\} \end{array} \right)$

- ***Step 1:*** $RAF\text{-}Acc(\mathfrak{U}) \subseteq Acc(\varepsilon_{\mathfrak{U}}) \setminus \left(\begin{array}{l} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathfrak{U}})\} \\ \cup\{\neg\beta \in Not_K | \beta \in Def(\varepsilon_{\mathfrak{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathfrak{U}}\} \end{array} \right)$

  Let $x \in RAF\text{-}Acc(\mathfrak{U})$*. For all attacks* $\alpha \in K$ *such that* $t(\alpha) = x$*, we have thus:*

  $$\alpha \in RAF\text{-}Inh(\mathfrak{U}) \text{ or } s(\alpha) \in RAF\text{-}Def(\mathfrak{U})$$

  As Assertion 1 holds, we have so:

  $$\alpha \in Def(\varepsilon_{\mathfrak{U}}) \setminus \left(\begin{array}{l} \{\neg a \in Not_A | a \in \varepsilon_{\mathfrak{U}}\} \\ \cup\{\neg\beta \in Not_K | \beta \in \varepsilon_{\mathfrak{U}}\} \\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathfrak{U}}) \text{ or } s(\beta) \in Def(\varepsilon_{\mathfrak{U}})\} \end{array} \right)$$

  or

  $$s(\alpha) \in Def(\varepsilon_{\mathfrak{U}}) \setminus \left(\begin{array}{l} \{\neg a \in Not_A | a \in \varepsilon_{\mathfrak{U}}\} \\ \cup\{\neg\beta \in Not_K | \beta \in \varepsilon_{\mathfrak{U}}\} \\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathfrak{U}}) \text{ or } s(\beta) \in Def(\varepsilon_{\mathfrak{U}})\} \end{array} \right)$$

  As a consequence, following the definition of `Raf2Af`*, we have:*

  $$\neg\alpha \in \varepsilon_{\mathfrak{U}} \text{ or } \neg s(\alpha) \in \varepsilon_{\mathfrak{U}}$$

  Furthermore, since $s(\alpha).\alpha \in And_{A,K}$*, we have:*

  $$s(\alpha).\alpha \in Def(\varepsilon_{\mathfrak{U}}) \cap And_{A,K}$$

As a consequence, as it is the case of any attack α *attacking x, we have:*

$$x \in Acc(\varepsilon_{\mathcal{U}})$$

Given that $x \in A \cup K$*, we have, following the definition of* `Raf2Af`*:* $x \notin (Not_A \cup Not_K \cup And_{A,K})$*. As a consequence, we have:*

$$x \in Acc(\varepsilon_{\mathcal{U}}) \setminus (Not_A \cup Not_K \cup And_{A,K})$$

We prove so that:

$$RAF\text{-}Acc(\mathcal{U}) \subseteq Acc(\varepsilon_{\mathcal{U}}) \setminus \left(\begin{array}{c} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\} \end{array} \right)$$

- ***Step 2:*** $RAF\text{-}Acc(\mathcal{U}) \supseteq Acc(\varepsilon_{\mathcal{U}}) \setminus \left(\begin{array}{c} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\} \end{array} \right)$

  Let $x \in Acc(\varepsilon_{\mathcal{U}}) \setminus \left(\begin{array}{c} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\} \end{array} \right)$

  Let consider four cases: $x \in Not_A$*,* $x \in Not_K$*,* $x \in And_{A,K}$ *and* $x \in A \cup K$*. Let show that the three first cases are impossible.*

  - *Let suppose that* $x \in Not_A$ *with* $x = \neg b$*. Given that* $\neg b \in Acc(\varepsilon_{\mathcal{U}})$*, according to the definition of* `Raf2Af`*, we have* $b \in Def(\varepsilon_{\mathcal{U}})$*. As a consequence, we have:* $x \in \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\}$*, which is a contradiction.*
  - *Let suppose that* $x \in Not_K$ *with* $x = \neg\alpha$*. Given that* $\neg\alpha \in Acc(\varepsilon_{\mathcal{U}})$*, according to the definition of* `Raf2Af`*, we have* $\alpha \in Def(\varepsilon_{\mathcal{U}})$*. As a consequence, we have:* $x \in \{\neg\beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\}$*, which is a contradiction.*
  - *Let suppose that* $x \in And_{A,K}$ *with* $x = s(\alpha).\alpha$*. Given that* $s(\alpha).\alpha \in Acc(\varepsilon_{\mathcal{U}})$*, according to the definition of* `Raf2Af`*, we have thus:* $\neg s(\alpha) \in Def(\varepsilon_{\mathcal{U}})$ *and* $\neg\alpha \in Def(\varepsilon_{\mathcal{U}})$*. And so we have:* $s(\alpha) \in \varepsilon_{\mathcal{U}}$ *and* $\alpha \in \varepsilon_{\mathcal{U}}$*. As a consequence, we have:* $x \in \{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\}$*, which is a contradiction.*

We prove so that: $x \in A \cup K$.

Given that $x \in Acc(\varepsilon_{\mathcal{U}})$, *then following the definition of* `Raf2Af`, *for all arguments* $s(\alpha).\alpha$ *attacking* x, *we have:* $s(\alpha).\alpha \in Def(\varepsilon_{\mathcal{U}})$. *As a consequence, following the definition of* `Raf2Af`, *we have:*

$$\neg s(\alpha) \in \varepsilon_{\mathcal{U}} \text{ or } \neg\alpha \in \varepsilon_{\mathcal{U}}$$

And so:

$$s(\alpha) \in Def(\varepsilon_{\mathcal{U}}) \text{ or } \alpha \in Def(\varepsilon_{\mathcal{U}})$$

As Assertion 1 holds and as $s(\alpha) \in A$ *and* $\alpha \in K$, *we have:*

$$s(\alpha) \in \textit{RAF-Def}(\mathcal{U}) \text{ or } \alpha \in \textit{RAF-Inh}(\mathcal{U})$$

as it is the case of any attack α *attacking* x, *we have:*

$$x \in \textit{RAF-Acc}(\mathcal{U})$$

We prove so that:

$$\textit{RAF-Acc}(\mathcal{U}) \supseteq Acc(\varepsilon_{\mathcal{U}}) \setminus \left(\begin{array}{c} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{\neg\beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\} \end{array} \right)$$

■

Proposition 14 *Let* $\Gamma = \langle A, K, s, t\rangle$ *be a RAF and* $\Gamma' = \texttt{Raf2Af}(\Gamma)$ *be an AF (with* $\Gamma' = \langle A', K'\rangle$*). The following properties holds:*

1. $\mathcal{U} = \langle S, Q\rangle$ *is a RAF-*complete *structure in* Γ *iff* $\varepsilon_{\mathcal{U}}$ *is a* complete *extension in* Γ'.

2. $\mathcal{U} = \langle S, Q\rangle$ *is a RAF-*grounded *structure in* Γ *iff* $\varepsilon_{\mathcal{U}}$ *is a* grounded *extension in* Γ'.

3. $\mathcal{U} = \langle S, Q\rangle$ *is a RAF-*preferred *structure in* Γ *iff* $\varepsilon_{\mathcal{U}}$ *is a* preferred *extension in* Γ'.

4. $\mathcal{U} = \langle S, Q\rangle$ *is a RAF-*stable *structure in* Γ *iff* $\varepsilon_{\mathcal{U}}$ *is a* stable *extension in* Γ'.

5. $\mathcal{U} = \langle S, Q \rangle$ *is a RAF*-semi-stable *structure in* Γ *iff* $\varepsilon_{\mathcal{U}}$ *is a* semi-stable *extension in* Γ'.

Proof. ***Assertion 1:*** $\mathcal{U} = \langle S, Q \rangle$ *is a RAF*-complete *structure in* Γ *iff* $\varepsilon_{\mathcal{U}}$ *is a* complete *extension in* Γ'.

$\mathcal{U} = \langle S, Q \rangle$ *is a RAF*-complete *structure in* Γ *iff* $(S \cup Q) = RAF\text{-}Acc(\mathcal{U})$.
Following Proposition 13, we have thus:

$$\mathcal{U} = \langle S, Q \rangle \text{ is a } RAF\text{-complete structure in } \Gamma$$
$$\textit{iff}$$
$$(S \cup Q) = Acc(\varepsilon_{\mathcal{U}}) \setminus \left(\begin{array}{c} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{\neg\beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\} \end{array} \right)$$

And so:

$$\mathcal{U} = \langle S, Q \rangle \text{ is a } RAF\text{-complete structure in } \Gamma$$
$$\textit{iff}$$
$$(S \cup Q) \cup \left(\begin{array}{c} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{\neg\beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\} \end{array} \right) = Acc(\varepsilon_{\mathcal{U}}) \tag{4.1}$$

Given that, for all $s(\beta).\beta \in And_{A,K}$ *such that* $s(\beta).\beta \in \varepsilon_{\mathcal{U}}$ *we have following the definition of* $\varepsilon_{\mathcal{U}}$*:* $\beta \in Q, s(\beta) \in S$*, from 4.1, we have then:*

$$\mathcal{U} = \langle S, Q \rangle \text{ is a } RAF\text{-complete structure in } \Gamma$$
$$\textit{iff}$$
$$(S \cup Q) \cup \left(\begin{array}{c} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{\neg\beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Q, s(\beta) \in S\} \end{array} \right) = Acc(\varepsilon_{\mathcal{U}}) \tag{4.2}$$

Following the definition of `Raf2Af`*:*

$$\neg a \in Not_A \textit{ iff } a \in A$$
$$\textit{and}$$
$$\neg\beta \in Not_K \textit{ iff } \beta \in K \tag{4.3}$$

Furthermore, following Proposition 13, we have:

$$Def(\varepsilon_{\mathcal{U}}) = \begin{pmatrix} RAF\text{-}Def(\mathcal{U}) \cup RAF\text{-}Inh(\mathcal{U}) \\ \cup\{\neg a \in Not_A | a \in \varepsilon_{\mathcal{U}}\} \cup \{\neg\beta \in Not_K | \beta \in \varepsilon_{\mathcal{U}}\} \\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathcal{U}}) \text{ or } s(\beta) \in Def(\varepsilon_{\mathcal{U}})\} \end{pmatrix} \tag{4.4}$$

From 4.3 and 4.4 we have so:

$$\begin{array}{c} a \in Def(\varepsilon_{\mathcal{U}}) \text{ iff } a \in RAF\text{-}Def(\mathcal{U}) \\ \text{and} \\ \beta \in Def(\varepsilon_{\mathcal{U}}) \text{ iff } \beta \in RAF\text{-}Inh(\mathcal{U}) \end{array} \tag{4.5}$$

As a consequence, we have from 4.2 and 4.5:

$$\begin{array}{c} \mathcal{U} = \langle S, Q \rangle \text{ is a } RAF\text{-complete structure in } \Gamma \\ \text{iff} \\ (S \cup Q) \cup \begin{pmatrix} \{\neg a \in Not_A | a \in RAF\text{-}Def(\mathcal{U})\} \\ \cup\{\neg\beta \in Not_K | \beta \in RAF\text{-}Inh(\mathcal{U})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Q, s(\beta) \in S\} \end{pmatrix} = Acc(\varepsilon_{\mathcal{U}}) \end{array}$$

Following the definition of $\varepsilon_{\mathcal{U}}$*, we have thus:*

$$\begin{array}{c} \mathcal{U} = \langle S, Q \rangle \text{ is a } RAF\text{-complete structure in } \Gamma \\ \text{iff} \\ \varepsilon_{\mathcal{U}} = Acc(\varepsilon_{\mathcal{U}}) \end{array}$$

We prove so that:

$$\begin{array}{c} \mathcal{U} = \langle S, Q \rangle \text{ is a } RAF\text{-complete structure in } \Gamma \\ \text{iff} \\ \varepsilon_{\mathcal{U}} \text{ is a complete extension in } \Gamma' \end{array}$$

Assertion 2: $\mathcal{U} = \langle S, Q \rangle$ *is a RAF*-grounded *structure in* Γ *iff* $\varepsilon_{\mathcal{U}}$ *is a* grounded *extension in* Γ'.

$\varepsilon_{\mathcal{U}}$ *is a* grounded *extension in* Γ' *iff there is no* complete *extension* $\varepsilon_{\mathcal{U}'}$ *in* Γ' *(with* $\mathcal{U}' = \langle S', Q' \rangle$*) such that:* $\varepsilon_{\mathcal{U}'} \subset \varepsilon_{\mathcal{U}}$.

We have so:

$$\varepsilon_{\mathcal{U}} \in \sigma_{gr}(\Gamma') \\ \textit{iff} \\ \nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \textit{ s.t. } Acc(\varepsilon_{\mathcal{U}'}) \subset Acc(\varepsilon_{\mathcal{U}})$$

Following Proposition 13, we have:

$$\begin{gathered}\varepsilon_{\mathcal{U}} \in \sigma_{gr}(\Gamma') \\ \textit{iff} \\ \nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \textit{ s.t.} \\ \textit{RAF-Acc}(\mathcal{U}') \cup \begin{pmatrix} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}'})\} \\ \cup\{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}'})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}'}\} \end{pmatrix} \\ \subset \\ \textit{RAF-Acc}(\mathcal{U}) \cup \begin{pmatrix} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\} \end{pmatrix}\end{gathered}$$

Removing $(Not_A \cup Not_K \cup And_{A,K})$ *from both sides give us a* $\subseteq$*-inclusion:*

$$\begin{gathered}\varepsilon_{\mathcal{U}} \in \sigma_{gr}(\Gamma') \\ \textit{iff} \\ \nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \textit{ s.t.} \\ \left(\textit{RAF-Acc}(\mathcal{U}') \cup \begin{pmatrix} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}'})\} \\ \cup\{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}'})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}'}\} \end{pmatrix} \right) \setminus (Not_A \cup Not_K \cup And_{A,K}) \\ \subseteq \\ \left(\textit{RAF-Acc}(\mathcal{U}) \cup \begin{pmatrix} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\} \end{pmatrix} \right) \setminus (Not_A \cup Not_K \cup And_{A,K})\end{gathered}$$

We have thus:

$$\varepsilon_{\mathcal{U}} \in \sigma_{gr}(\Gamma') \\ \textit{iff} \\ \nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \textit{ s.t. } \textit{RAF-Acc}(\mathcal{U}') \subseteq \textit{RAF-Acc}(\mathcal{U})$$

Given that, following Assertion 1, $\varepsilon_{\mathcal{U}'}$ and $\varepsilon_{\mathcal{U}}$ are complete *iff $\mathcal{U}'$ and $\mathcal{U}$ are RAF-*complete*, we have thus:*

$$\varepsilon_{\mathcal{U}} \in \sigma_{gr}(\Gamma') \\ \textit{iff} \\ \nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \text{ s.t. } \mathcal{U}' \subseteq \mathcal{U}$$

Given that $\varepsilon_{\mathcal{U}'} \neq \varepsilon_{\mathcal{U}}$ iff $\mathcal{U}' \neq \mathcal{U}$, we have thus:

$$\varepsilon_{\mathcal{U}} \in \sigma_{gr}(\Gamma') \\ \textit{iff} \\ \nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \text{ s.t. } \mathcal{U}' \subset \mathcal{U}$$

We prove so that $\varepsilon_{\mathcal{U}}$ is a grounded *extension in Γ' iff $\mathcal{U} = \langle S, Q \rangle$ is a RAF-*grounded *structure in Γ.*

Assertion 3: *$\mathcal{U} = \langle S, Q \rangle$ is a RAF-*preferred *structure in Γ iff $\varepsilon_{\mathcal{U}}$ is a* preferred *extension in Γ'.*

$\varepsilon_{\mathcal{U}}$ is a preferred *extension in Γ' iff there is no* complete *extension $\varepsilon_{\mathcal{U}'}$ in Γ' (with $\mathcal{U}' = \langle S', Q' \rangle$) such that: $\varepsilon_{\mathcal{U}} \subset \varepsilon_{\mathcal{U}'}$.*

We have so:

$$\varepsilon_{\mathcal{U}} \in \sigma_{pr}(\Gamma') \\ \textit{iff} \\ \nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \text{ s.t. } Acc(\varepsilon_{\mathcal{U}}) \subset Acc(\varepsilon_{\mathcal{U}'})$$

Following Proposition 13, we have:

$$\varepsilon_{\mathcal{U}} \in \sigma_{pr}(\Gamma') \\ \textit{iff} \\ \nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \text{ s.t.} \\ \textit{RAF-Acc}(\mathcal{U}) \cup \begin{pmatrix} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup \{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup \{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\} \end{pmatrix} \\ \subset \\ \textit{RAF-Acc}(\mathcal{U}') \cup \begin{pmatrix} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}'})\} \\ \cup \{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}'})\} \\ \cup \{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}'}\} \end{pmatrix}$$

Removing $(Not_A \cup Not_K \cup And_{A,K})$ *from both sides give us a* $\subseteq$*-inclusion:*

$$\varepsilon_{\mathcal{U}} \in \sigma_{pr}(\Gamma')$$
$$\textit{iff}$$
$$\nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \textit{ s.t.}$$
$$\left(RAF\text{-}Acc(\mathcal{U}) \cup \left(\begin{array}{c} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\} \end{array} \right) \right) \setminus (Not_A \cup Not_K \cup And_{A,K})$$
$$\subseteq$$
$$\left(RAF\text{-}Acc(\mathcal{U}') \cup \left(\begin{array}{c} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}'})\} \\ \cup\{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}'})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}'}\} \end{array} \right) \right) \setminus (Not_A \cup Not_K \cup And_{A,K})$$

We have thus:

$$\varepsilon_{\mathcal{U}} \in \sigma_{pr}(\Gamma')$$
$$\textit{iff}$$
$$\nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \textit{ s.t. } RAF\text{-}Acc(\mathcal{U}) \subseteq RAF\text{-}Acc(\mathcal{U}')$$

Given that, following Assertion 1, $\varepsilon_{\mathcal{U}'}$ *and* $\varepsilon_{\mathcal{U}}$ *are* complete *iff* $\mathcal{U}'$ *and* $\mathcal{U}$ *are* *RAF*-complete*, we have thus:*

$$\varepsilon_{\mathcal{U}} \in \sigma_{pr}(\Gamma')$$
$$\textit{iff}$$
$$\nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \textit{ s.t. } \mathcal{U} \subseteq \mathcal{U}'$$

Given that $\varepsilon_{\mathcal{U}'} \neq \varepsilon_{\mathcal{U}}$ *iff* $\mathcal{U}' \neq \mathcal{U}$*, we have thus:*

$$\varepsilon_{\mathcal{U}} \in \sigma_{pr}(\Gamma')$$
$$\textit{iff}$$
$$\nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \textit{ s.t. } \mathcal{U} \subset \mathcal{U}'$$

We prove so that $\varepsilon_{\mathcal{U}}$ *is a* preferred *extension in* Γ' *iff* $\mathcal{U} = \langle S, Q \rangle$ *is a RAF*-preferred *structure in* Γ.

Assertion 4: $\mathcal{U} = \langle S, Q \rangle$ *is a RAF*-stable *structure in* Γ *iff* $\varepsilon_{\mathcal{U}}$ *is a* stable *extension in* Γ'.

- ***Step 1:*** *If* $\mathcal{U} = \langle S, Q \rangle$ *is a* RAF-stable *structure in* Γ *then* $\varepsilon_{\mathcal{U}}$ *is a* stable *extension in* Γ'.

  If $\mathcal{U} = \langle S, Q \rangle$ *is a* RAF-stable *structure in* Γ *then* $\nexists x \in (A \cup K)$ *such that* $x \notin \mathcal{U}$ *and* $x \notin (RAF\text{-}Def(\mathcal{U}) \cup RAF\text{-}Inh(\mathcal{U}))$. *If* $\mathcal{U}$ *is a* RAF-stable *structure in* Γ *then* $\mathcal{U}$ *is also* RAF-complete. *Following Assertion 1,* $\varepsilon_{\mathcal{U}}$ *is thus a* complete *extension in* Γ'. *Let suppose that* $\varepsilon_{\mathcal{U}}$ *is not* stable. *There exists thus* $x \in (A' \cup K')$ *such that* $x \notin \varepsilon_{\mathcal{U}}$ *and* $x \notin Def(\varepsilon_{\mathcal{U}})$.

  Let consider two cases: $x \in A \cup K$ *and* $x \notin (A \cup K)$.

  Case 1: *Given that* $\mathcal{U}$ *is* RAF-stable, *if* $x \in A \cup K$, *we have so:* $x \in (S \cup Q \cup RAF\text{-}Def(\mathcal{U}) \cup RAF\text{-}Inh(\mathcal{U}))$.

  As shown in Proof of Assertion 1 (Equivalence 4.5):

  $$\begin{array}{c} a \in A \cap Def(\varepsilon_{\mathcal{U}}) \textit{ iff } a \in RAF\text{-}Def(\mathcal{U}) \\ \textit{and} \\ \beta \in K \cap Def(\varepsilon_{\mathcal{U}}) \textit{ iff } \beta \in RAF\text{-}Inh(\mathcal{U}) \end{array} \tag{4.6}$$

  As a consequence if $x \in A \cup K$ *then:*

  $$x \in (S \cup Q \cup (A \cap Def(\varepsilon_{\mathcal{U}})) \cup (K \cap Def(\varepsilon_{\mathcal{U}})))$$

  As $x \notin \varepsilon_{\mathcal{U}}$ *and* $x \notin Def(\varepsilon_{\mathcal{U}})$ *there is a contradiction.*

  Case 2: *If* $x \notin (A \cup K)$ *then:* $x \in (Not_A \cup Not_K \cup And_{A,K})$.

  Given that:

  $$x \notin \varepsilon_{\mathcal{U}} \textit{ and } x \notin Def(\varepsilon_{\mathcal{U}})$$

  We have thus three possible cases:

  - $x \in Not_A \setminus \{\neg a \in Not_A | a \in (Def(\varepsilon_{\mathcal{U}}) \cup \varepsilon_{\mathcal{U}})\}$
  - $x \in Not_K \setminus \{\neg \beta \in Not_K | \beta \in (Def(\varepsilon_{\mathcal{U}}) \cup \varepsilon_{\mathcal{U}})\}$
  - $x \in And_{A,K} \setminus \{s(\beta).\beta \in And_{A,K} | \beta \in (Def(\varepsilon_{\mathcal{U}}) \cup \varepsilon_{\mathcal{U}})\}$

Given that following the definition of `Raf2Af`*:*

$$\begin{array}{c} \neg a \in Not_A \textit{ iff } a \in A \\ \textit{and} \\ \neg\beta \in Not_K \textit{ iff } \beta \in K \\ \textit{and} \\ s(\beta).\beta \in And_{A,K} \textit{ iff } \beta \in K \end{array} \tag{4.7}$$

We have thus:

- $x \in Not_A \setminus \{\neg a \in Not_A | a \in (A \cap Def(\varepsilon_{\mathcal{U}})) \cup (A \cap \varepsilon_{\mathcal{U}})\}$
- $x \in Not_K \setminus \{\neg\beta \in Not_K | \beta \in (K \cap Def(\varepsilon_{\mathcal{U}})) \cup (K \cap \varepsilon_{\mathcal{U}})\}$
- $x \in And_{A,K} \setminus \{s(\beta).\beta \in And_{A,K} | \beta \in (K \cap Def(\varepsilon_{\mathcal{U}})) \cup (K \cap \varepsilon_{\mathcal{U}})\}$

As shown in Proof of Assertion 1 (Equivalence 4.5):

$$\begin{array}{c} A \cap Def(\varepsilon_{\mathcal{U}}) = \textit{RAF-Def}(\mathcal{U}) \\ \textit{and} \\ K \cap Def(\varepsilon_{\mathcal{U}}) = \textit{RAF-Inh}(\mathcal{U}) \end{array} \tag{4.8}$$

We have thus:

- $x \in Not_A \setminus \{\neg a \in Not_A | a \in (\textit{RAF-Def}(\mathcal{U}) \cup S)\}$
- $x \in Not_K \setminus \{\neg\beta \in Not_K | \beta \in (\textit{RAF-Inh}(\mathcal{U}) \cup Q)\}$
- $x \in And_{A,K} \setminus \{s(\beta).\beta \in And_{A,K} | \beta \in (\textit{RAF-Inh}(\mathcal{U}) \cup Q)\}$

Given that $\mathcal{U}$ *is RAF*-stable*, we have thus:*

- $x \in Not_A \setminus Not_A = \varnothing$
- $x \in Not_K \setminus Not_K = \varnothing$
- $x \in And_{A,K} \setminus And_{A,K} = \varnothing$

There is thus a contradiction.

We prove so that if $\mathcal{U} = \langle S, Q \rangle$ *is a RAF*-stable *structure in* Γ *then* $\varepsilon_{\mathcal{U}}$ *is a* stable *extension in* Γ'.

- ***Step 2:*** *If* $\varepsilon_{\mathcal{U}}$ *is a* stable *extension in* Γ' *then* $\mathcal{U} = \langle S, Q \rangle$ *is a RAF*-stable *structure in* Γ.

  If $\varepsilon_{\mathcal{U}}$ *is a* stable *extension in* Γ' *then* $\nexists x \in (A' \cup K')$ *such that* $x \notin \varepsilon_{\mathcal{U}}$ *and* $x \notin Def(\varepsilon_{\mathcal{U}})$. *If* $\varepsilon_{\mathcal{U}}$ *is a* stable *extension in* Γ' *then* $\varepsilon_{\mathcal{U}}$ *is also* complete. *As Assertion 1 holds, then* $\mathcal{U}$ *is RAF*-complete. *Let suppose that* $\mathcal{U}$ *is not RAF*-stable. *There exists thus* $x \in (A \cup K)$ *such that* $x \notin \mathcal{U}$ *and* $x \notin (RAF\text{-}Def(\mathcal{U}) \cup RAF\text{-}Inh(\mathcal{U}))$.

  As shown in Proof of Assertion 1 (Equivalence 4.5):

$$A \cap Def(\varepsilon_{\mathcal{U}}) = RAF\text{-}Def(\mathcal{U})$$
$$and$$
$$K \cap Def(\varepsilon_{\mathcal{U}}) = RAF\text{-}Inh(\mathcal{U})$$

  We have thus:

$$x \notin \Big(\; (A \cap Def(\varepsilon_{\mathcal{U}})) \cup (K \cap Def(\varepsilon_{\mathcal{U}})) \cup S \cup Q \; \Big)$$

  Given that $\varepsilon_{\mathcal{U}}$ *is* stable*, we have thus:* $x \notin (A \cup K)$*, which is a contradiction.*

  We prove so that if $\varepsilon_{\mathcal{U}}$ *is a* stable *extension in* Γ' *then* $\mathcal{U} = \langle S, Q \rangle$ *is a RAF*-stable *structure in* Γ.

Assertion 5: $\mathcal{U} = \langle S, Q \rangle$ *is a RAF*-semi-stable *structure in* Γ *iff* $\varepsilon_{\mathcal{U}}$ *is a* semi-stable *extension in* Γ'.

$\varepsilon_{\mathcal{U}}$ *is a* semi-stable *extension in* Γ' *iff there is no* complete *extension* $\varepsilon_{\mathcal{U}'}$ *in* Γ' *(with* $\mathcal{U}' = \langle S', Q' \rangle$*) such that:* $(\varepsilon_{\mathcal{U}} \cup Def(\varepsilon_{\mathcal{U}})) \subset (\varepsilon_{\mathcal{U}'} \cup Def(\varepsilon_{\mathcal{U}'}))$.

We have so:

$$\varepsilon_{\mathcal{U}} \in \sigma_{sst}(\Gamma')$$
$$iff$$
$$\nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \text{ s.t. } (Acc(\varepsilon_{\mathcal{U}}) \cup Def(\varepsilon_{\mathcal{U}})) \subset (Acc(\varepsilon_{\mathcal{U}'}) \cup Def(\varepsilon_{\mathcal{U}'}))$$

Following Proposition 13, we have:

$$\varepsilon_{\mathcal{U}} \in \sigma_{sst}(\Gamma')$$
$$\textit{iff}$$
$$\nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \textit{ s.t.}$$
$$\begin{pmatrix} \textit{RAF-Acc}(\mathcal{U}) \\ \cup \textit{RAF-Def}(\mathcal{U}) \\ \cup \textit{RAF-Inh}(\mathcal{U}) \end{pmatrix} \cup \begin{pmatrix} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}}\} \\ \cup\{\neg a \in Not_A | a \in \varepsilon_{\mathcal{U}}\} \\ \cup\{\neg \beta \in Not_K | \beta \in \varepsilon_{\mathcal{U}}\} \\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathcal{U}}) \textit{ or } s(\beta) \in Def(\varepsilon_{\mathcal{U}})\} \end{pmatrix}$$
$$\subset$$
$$\begin{pmatrix} \textit{RAF-Acc}(\mathcal{U}') \\ \cup \textit{RAF-Def}(\mathcal{U}') \\ \cup \textit{RAF-Inh}(\mathcal{U}') \end{pmatrix} \cup \begin{pmatrix} \{\neg a \in Not_A | a \in Def(\varepsilon_{\mathcal{U}'})\} \\ \cup\{\neg \beta \in Not_K | \beta \in Def(\varepsilon_{\mathcal{U}'})\} \\ \cup\{s(\beta).\beta \in And_{A,K} | s(\beta).\beta \in \varepsilon_{\mathcal{U}'}\} \\ \cup\{\neg a \in Not_A | a \in \varepsilon_{\mathcal{U}'}\} \\ \cup\{\neg \beta \in Not_K | \beta \in \varepsilon_{\mathcal{U}'}\} \\ \cup\{s(\beta).\beta \in And_{A,K} | \beta \in Def(\varepsilon_{\mathcal{U}'}) \textit{ or } s(\beta) \in Def(\varepsilon_{\mathcal{U}'})\} \end{pmatrix}$$

Removing $(Not_A \cup Not_K \cup And_{A,K})$ *from both sides give us a* $\subseteq$*-inclusion:*

$$\varepsilon_{\mathcal{U}} \in \sigma_{sst}(\Gamma')$$
$$\textit{iff}$$
$$\nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \textit{ s.t.} \begin{pmatrix} \textit{RAF-Acc}(\mathcal{U}) \\ \cup \textit{RAF-Def}(\mathcal{U}) \\ \cup \textit{RAF-Inh}(\mathcal{U}) \end{pmatrix} \subseteq \begin{pmatrix} \textit{RAF-Acc}(\mathcal{U}') \\ \cup \textit{RAF-Def}(\mathcal{U}') \\ \cup \textit{RAF-Inh}(\mathcal{U}') \end{pmatrix}$$

Given that, following Assertion 1, $\varepsilon_{\mathcal{U}'}$ *and* $\varepsilon_{\mathcal{U}}$ *are* complete *iff* $\mathcal{U}'$ *and* $\mathcal{U}$ *are* *RAF*-complete*, we have thus:*

$$\varepsilon_{\mathcal{U}} \in \sigma_{sst}(\Gamma')$$
$$\textit{iff}$$
$$\nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \textit{ s.t. } \mathcal{U} \cup \begin{pmatrix} \textit{RAF-Def}(\mathcal{U}) \\ \cup \textit{RAF-Inh}(\mathcal{U}) \end{pmatrix} \subseteq \mathcal{U}' \cup \begin{pmatrix} \textit{RAF-Def}(\mathcal{U}') \\ \cup \textit{RAF-Inh}(\mathcal{U}') \end{pmatrix}$$

Given that $\varepsilon_{\mathcal{U}'} \neq \varepsilon_{\mathcal{U}}$ *iff* $\mathcal{U}' \neq \mathcal{U}$*, we have thus:*

$$\varepsilon_{\mathcal{U}} \in \sigma_{sst}(\Gamma')$$
$$\textit{iff}$$
$$\nexists \varepsilon_{\mathcal{U}'} \in \sigma_{co}(\Gamma') \text{ s.t. } \mathcal{U} \cup \begin{pmatrix} \textit{RAF-Def}(\mathcal{U}) \\ \cup \textit{RAF-Inh}(\mathcal{U}) \end{pmatrix} \subset \mathcal{U}' \cup \begin{pmatrix} \textit{RAF-Def}(\mathcal{U}') \\ \cup \textit{RAF-Inh}(\mathcal{U}') \end{pmatrix}$$

We prove so that $\varepsilon_{\mathcal{U}}$ *is a* semi-stable *extension in* Γ' *iff* $\mathcal{U} = \langle S, Q \rangle$ *is a RAF-*semi-stable *structure in* Γ. ■

4.4.3 Complexity results

Proposition 15 *Let* $\Gamma = \langle A, K, s, t \rangle$ *be an RAF and* $\Gamma' = \texttt{Raf2Af}(\Gamma)$ *be an AF (with* $\Gamma' = \langle A', K' \rangle$*). Let* $a \in (A \cup K)$ *be an element in* Γ *and an argument in* Γ'*, following the definition of* `Raf2Af`*. Let* $\mathcal{U} = \langle S, Q \rangle$ *be an structure of* Γ.

For each semantics $\sigma \in \{$complete, semi-stable, stable, preferred, grounded$\}$, *we have:*

1. *RAF-Cred*$_\sigma$ *accepts* (Γ, a) *iff AF-Cred*$_\sigma$ *accepts* (Γ', a).

2. *RAF-Skep*$_\sigma$ *accepts* (Γ, a) *iff AF-Skep*$_\sigma$ *accepts* (Γ', a).

3. *RAF-Ver*$_\sigma$ *accepts* $(\Gamma, \mathcal{U})$ *iff AF-Ver*$_\sigma$ *accepts* $(\Gamma', \varepsilon_{\mathcal{U}})$.

4. *RAF-Exists*$_\sigma$ *accepts* Γ *iff AF-Exists*$_\sigma$ *accepts* Γ'.

5. *RAF-Exists*$_\sigma^{\neg\emptyset}$ *accepts* Γ *iff AF-Exists*$_\sigma^{\neg\emptyset}$ *accepts* Γ'.

6. *RAF-Unique*$_\sigma$ *accepts* Γ *iff AF-Unique*$_\sigma$ *accepts* Γ'.

PROOF. ***Assertion 1:*** *RAF-Cred*$_\sigma$ *accepts* (Γ, a) *iff AF-Cred*$_\sigma$ *accepts* (Γ', a).

RAF-Cred$_\sigma$ *accepts* (Γ, a) *iff* $\exists \mathcal{U} \in \sigma(\Gamma)$ *s.t.* $a \in \mathcal{U}$
iff $\exists \varepsilon_{\mathcal{U}} \in \sigma(\Gamma')$ *s.t.* $a \in \varepsilon_{\mathcal{U}}$ *(following Proposition 14)*
iff AF-Cred$_\sigma$ *accepts* (Γ', a)

Assertion 2: *RAF-Skep$_\sigma$ accepts* (Γ, a) *iff AF-Skep$_\sigma$ accepts* (Γ', a).

$$\begin{aligned} \textit{RAF-Skep}_\sigma \textit{ accepts } (\Gamma, a) &\textit{ iff } \forall \mathfrak{U} \in \sigma(\Gamma),\ a \in \mathfrak{U} \\ &\textit{ iff } \forall \varepsilon_{\mathfrak{U}} \in \sigma(\Gamma'),\ a \in \varepsilon_{\mathfrak{U}} \quad \textit{(following Proposition 14)} \\ &\textit{ iff AF-Skep}_\sigma \textit{ accepts } (\Gamma', a) \end{aligned}$$

Assertion 3: *RAF-Ver$_\sigma$ accepts* $(\Gamma, \mathfrak{U})$ *iff AF-Ver$_\sigma$ accepts* $(\Gamma', \varepsilon_{\mathfrak{U}})$.

$$\begin{aligned} \textit{RAF-Ver}_\sigma \textit{ accepts } (\Gamma, \mathfrak{U}) &\textit{ iff } \mathfrak{U} \in \sigma(\Gamma) \\ &\textit{ iff } \varepsilon_{\mathfrak{U}} \in \sigma(\Gamma') \quad \textit{(following Proposition 14)} \\ &\textit{ iff AF-Ver}_\sigma \textit{ accepts } (\Gamma', \varepsilon_{\mathfrak{U}}) \end{aligned}$$

Assertion 4: *RAF-Exists$_\sigma$ accepts* Γ *iff AF-Exists$_\sigma$ accepts* Γ'.

$$\begin{aligned} \textit{RAF-Exists}_\sigma \textit{ accepts } \Gamma &\textit{ iff } \exists \mathfrak{U} \in \sigma(\Gamma) \\ &\textit{ iff } \exists \varepsilon_{\mathfrak{U}} \in \sigma(\Gamma') \quad \textit{(following Proposition 14)} \\ &\textit{ iff AF-Exists}_\sigma \textit{ accepts } \Gamma' \end{aligned}$$

Assertion 5: *RAF-Exists$_\sigma^{\neg\varnothing}$ accepts* Γ *iff AF-Exists$_\sigma^{\neg\varnothing}$ accepts* Γ'.

$$\begin{aligned} \textit{RAF-Exists}_\sigma^{\neg\varnothing} \textit{ accepts } \Gamma &\textit{ iff } \exists (\mathfrak{U} = \langle S, Q \rangle) \in \sigma(\Gamma) \textit{ s.t. } (S \cup Q) \neq \varnothing \\ &\textit{ iff } \exists \varepsilon_{\mathfrak{U}} \in \sigma(\Gamma') \textit{ s.t. } \varepsilon_{\mathfrak{U}} \neq \varnothing \quad \textit{(following Proposition 14)} \\ &\textit{ iff AF-Exists}_\sigma^{\neg\varnothing} \textit{ accepts } \Gamma' \end{aligned}$$

Assertion 6: *RAF-Unique$_\sigma$ accepts* Γ *iff AF-Unique$_\sigma$ accepts* Γ'.

$$\begin{aligned} \textit{RAF-Unique}_\sigma \textit{ accepts } \Gamma &\textit{ iff } \exists! \mathfrak{U} \in \sigma(\Gamma) \\ &\textit{ iff } \exists! \varepsilon_{\mathfrak{U}} \in \sigma(\Gamma') \quad \textit{(following Proposition 14)} \\ &\textit{ iff AF-Unique}_\sigma \textit{ accepts } \Gamma' \end{aligned}$$

■

Proposition 16 *The complexities of AF decision problems are at least as hard as RAF ones, for the semantics* complete, semi-stable, stable, preferred, grounded.

PROOF. *Given that* `Raf2Af` *is a polynomial time, log-space function, then according to Proposition 15, for each semantics* $\sigma \in \{$complete, semi-stable, stable, preferred, grounded$\}$ *we have:*

- $RAF\text{-}Cred_\sigma \leq_L^{\texttt{Raf2Af}} AF\text{-}Cred_\sigma$
- $RAF\text{-}Skep_\sigma \leq_L^{\texttt{Raf2Af}} AF\text{-}Skep_\sigma$
- $RAF\text{-}Ver_\sigma \leq_L^{\texttt{Raf2Af}} AF\text{-}Ver_\sigma$
- $RAF\text{-}Exists_\sigma \leq_L^{\texttt{Raf2Af}} AF\text{-}Exists_\sigma$
- $RAF\text{-}Exists_\sigma^{\neg\varnothing} \leq_L^{\texttt{Raf2Af}} AF\text{-}Exists_\sigma^{\neg\varnothing}$
- $RAF\text{-}Unique_\sigma \leq_L^{\texttt{Raf2Af}} AF\text{-}Unique_\sigma$

■

Proposition 17 *Let* $\Gamma = \langle A, K \rangle$ *be an AF and* $\Gamma' = \texttt{Af2Raf}(\Gamma)$ *be an RAF. Let* $a \in A$ *be an argument in* Γ *and in* Γ'*, following the definition of* `Af2Raf`*. For each semantics* $\sigma \in \{$complete, semi-stable, stable, preferred, grounded$\}$*, we have:*

1. $AF\text{-}Cred_\sigma$ *accepts* (Γ', a) *iff* $AF\text{-}Cred_\sigma$ *accepts* (Γ, a).
2. $AF\text{-}Skep_\sigma$ *accepts* (Γ', a) *iff* $AF\text{-}Skep_\sigma$ *accepts* (Γ, a).
3. $AF\text{-}Ver_\sigma$ *accepts* (Γ', S) *iff* $AF\text{-}Ver_\sigma$ *accepts* $(\Gamma, \mathcal{U} = \langle S, K \rangle)$.
4. $AF\text{-}Exists_\sigma$ *accepts* Γ' *iff* $AF\text{-}Exists_\sigma$ *accepts* Γ.
5. $AF\text{-}Exists_\sigma^{\neg\varnothing}$ *accepts* Γ' *iff* $AF\text{-}Exists_\sigma^{\neg\varnothing}$ *accepts* Γ.
6. $AF\text{-}Unique_\sigma$ *accepts* Γ' *iff* $AF\text{-}Unique_\sigma$ *accepts* Γ.

PROOF. *This proof is trivial considering Theorem 5 in [6] and Proposition 3 in [10].* ■

Proposition 18 *The complexities of RAF decision problems are at least as hard as AF ones, for the semantics* complete, semi-stable, stable, preferred, grounded.

PROOF. *Given that* `Af2Raf` *is a polynomial time, log-space function, then according to Proposition 17, for each semantics* $\sigma \in \{$complete, semi-stable, stable, preferred, grounded$\}$ *we have:*

- $AF\text{-}Cred_\sigma \leq_L^{\texttt{Af2Raf}} RAF\text{-}Cred_\sigma$
- $AF\text{-}Skep_\sigma \leq_L^{\texttt{Af2Raf}} RAF\text{-}Skep_\sigma$
- $AF\text{-}Ver_\sigma \leq_L^{\texttt{Af2Raf}} RAF\text{-}Ver_\sigma$
- $AF\text{-}Exists_\sigma \leq_L^{\texttt{Af2Raf}} RAF\text{-}Exists_\sigma$
- $AF\text{-}Exists_\sigma^{\neg\emptyset} \leq_L^{\texttt{Af2Raf}} RAF\text{-}Exists_\sigma^{\neg\emptyset}$
- $AF\text{-}Unique_\sigma \leq_L^{\texttt{Af2Raf}} RAF\text{-}Unique_\sigma$

■

Proposition 19 *The complexities of RAF decision problems are the same as AF ones, for the semantics* complete, semi-stable, stable, preferred, grounded, *as stated in Table 4.3.*

PROOF. *Given that* `Raf2Af` *and* `Af2Raf` *are polynomial time procedures and that Propositions 16 and 18 holds, then all the complexities are the same.* ■

σ	*RAF-*					
	$Cred_\sigma$	$Skep_\sigma$	Ver_σ	$Exists_\sigma$	$Exists_\sigma^{\neg\emptyset}$	$Unique_\sigma$
Grounded	P-*c*	P-*c*	P-*c*	trivial	in L	trivial
Complete	NP-*c*	P-*c*	in L	trivial	NP-*c*	coNP-*c*
Preferred	NP-*c*	Π_2^{P}-*c*	coNP-*c*	trivial	NP-*c*	coNP-*c*
Stable	NP-*c*	coNP-*c*	in L	NP-*c*	NP-*c*	DP-*c*
Semi-stable	Σ_2^{P}-*c*	Π_2^{P}-*c*	coNP-*c*	trivial	NP-*c*	in Θ_2^{P}

Table 4.3: Complexities of RAF

Chapter 5

Conclusion and perspectives

This paper contains a number of contributions regarding argumentation frameworks with higher-order attacks. It defines decision problems for Recursive Argumentation Frameworks (RAF) and for Argumentation Frameworks with Recursive Attacks (AFRA), and it investigates their complexity. By doing so, a new "flattening" process of a RAF to an AF is introduced. An important result is the fact that the complexities for the decision problems in the context of enriched frameworks, are the same as the one in Dung's framework, despite all the additional expressivity that is brought by the higher order attacks.

All these results pave the way for the research on algorithmic issues related to frameworks with higher-order attacks, regarding the computation of acceptable arguments, and of the decision problems. We plan to adapt a recent efficient approach that embed machine learning techniques ([9]) for Dung's framework, to these frameworks with higher-order attacks. The labelling counterpart provided for RAF will be used in this sense (see [10]).

Beside decision problems, other problems are of interest for argumentation frameworks, whether they be with higher-order attacks or not: function problems[1]. The functional counterpart of $Cred_\sigma$ and $Exists_\sigma^{\neg\emptyset}$ may turn to be particularly useful in the context of a dialogue between agents, the output being here the concerned acceptable set. We plan to define such problems, and to investigate their complexity.

[1] In computational complexity theory, a function problem is a computational problem where a single output is expected for every input, but the output is more complex than that of a decision problem: it is not simply "yes" or "no".

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] Álvaro Carrera and Carlos A Iglesias. A systematic review of argumentation techniques for multi-agent systems research. *Artificial Intelligence Review*, 44(4):509–535, 2015.

[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] Stephen A Cook. An overview of computational complexity. *Communications of the ACM*, 26(6):400–408, 1983.

[9] Sylvie Doutre, Mickaël Lafages, and Marie-Christine Lagasquie-Schiex. A distributed and clustering-based algorithm for the enumeration problem in abstract argumentation. In *PRIMA 2019: Principles and Practice of Multi-Agent Systems*, volume 11873 of *Lecture Notes in Computer Science*, pages 87–105. Springer, 2019.

[10] Sylvie Doutre, Mickaël Lafages, and Marie-Christine Lagasquie-Schiex. Argumentation Frameworks with Higher-Order Attacks: Labelling Semantics. Rapport de recherche IRIT/RR–2020–01–FR, IRIT, Université Paul Sabatier, Toulouse, janvier 2020.

[11] 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.

[12] Wolfgang Dvořák and Paul E Dunne. Computational problems in formal argumentation and their complexity. *Journal of Logics and their Applications*, 4(8):2557–2622, 2017.

[13] Wolfgang Dvorak and Paul E. Dunne. Computational problems in formal argumentation and their complexity. In *Handbook of formal argumentation*, pages 631–688. College publication, 2018.

[14] 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.

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

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

[17] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1):230–265, 1937.