Dynamics of Argumentation Frameworks

THÈSE

présentée et soutenue publiquement le 30 septembre 2015

en vue de l'obtention du

Doctorat de l'Université d'Artois
(Spécialité Informatique)

par

Jean-Guy Mailly

Composition du jury

Rapporteurs :	Gerhard Brewka	University of Leipzig
	Stefan Woltran	Vienna University of Technology
Examinateurs :	Salem Benferhat	Université d'Artois
	Andreas Herzig	CNRS - Université Paul Sabatier
	Anthony Hunter	University College London
Encadrants :	Sylvie Coste-Marquis	Université d'Artois
	Sébastien Konieczny	CNRS - Université d'Artois
	Pierre Marquis	Université d'Artois

Centre de Recherche en Informatique de Lens – CNRS UMR 8188
Université d'Artois, rue Jean Souvraz, S.P. 18 F-62307, Lens Cedex France
Secrétariat : Tél.: +33 (0)3 21 79 17 23 – Fax : +33 (0)3 21 79 17 70
http://www.cril.fr

Mise en page avec memcril (B. Mazure, CRIL) et thloria (D. Roegel, LORIA).

Remerciements

> *You know that in nine hundred years of time and space I've never met anybody who wasn't important before.*
>
> The Doctor – *Doctor Who - A Christmas Carol*

La thèse est une longue aventure, semée d'embûches, et que je n'aurais pas pu surmonter seul. Dans un premier temps, je ne peux pas ne pas remercier mes encadrants, Sylvie Coste-Marquis, Sébastien Konieczny, et Pierre Marquis. Trois ans à bénéficier de leurs conseils m'ont fait évoluer et, j'espère, donné quelques compétences pour la carrière qui m'attends (j'espère aussi !). Travailler avec eux a été une expérience enrichissante, à la fois d'un point de vue professionnel et scientifique, mais aussi d'un point de vue humain. Merci à tous les trois pour leur gentilesse et leur patience, lorsque mon travail ne suivait pas la route (que j'aurais voulue droite) qui va d'une piste à un résultat, mais faisait au contraire des détours par quelques culs-de-sac.

D'autres chercheurs ont manifesté de l'intérêt à mes travaux, et m'ont d'une façon ou d'une autre aidé à avancer au fil de cette aventure. Je tiens en particulier à remercier Gerhard Brewka et Stefan Woltran, qui m'ont fait l'honneur d'être rapporteurs de cette thèse, ainsi que Salem Benferhat, Andreas Herzig et Anthony Hunter, qui participent à mon jury de soutenance en tant qu'examinateurs. Je n'aurais pu rêver d'une telle assemblée face à moi au moment de soutenir cette thèse.

Les nombreuses discussions que j'ai eues avec des collègues rencontrés lors des missions m'ont apporté énormément. Il me faudrait une place déraisonnable, et un travail de mémoire phénoménal, pour n'oublier personne. Je tiens tout de même à remercier les membres du projet AMANDE, et en particulier Sylvie Doutre, qui m'a proposé de me joindre à Laurent Perrussel et elle-même pour un premier travail réalisé sans mes encadrants. Merci aussi à Jean-Marie Lagniez et Emmanuel Lonca, grâce à qui le projet CoQuiAAS a vu le jour.

Un grand merci à tous les collègues du CRIL, mes anciens professeurs, qui m'ont fait m'intéresser à l'informatique, et à la recherche en IA en particulier. Merci aussi à mes camarades doctorants, en particuliers mes partenaires du MIT : Long, Manu, Zied, Nicolas et Éric ; ainsi que ceux avec qui j'ai fait mes études, à commencer par mon binôme de Licence, Thomas. Certains d'entre eux m'ont appris beaucoup sur moi-même, en particulier les limites de ma patience ! Je suppose qu'au final, ça n'a pas de prix, ça fait aussi partie de l'expérience de la thèse !

Et car il n'y a pas que le travail dans la vie, je tiens à remercier les personnes qui m'ont permis de tenir le rythme, hors du travail. Les plus importants sont sans aucun doute mes parents, et toute ma famille, qui m'ont soutenu, bien qu'il ne soit pas toujours évident pour eux de comprendre ce que je fais (et je ne leur en veux pas, il n'y a pas si longtemps que ça, je n'aurais pas compris non plus !).

Merci aussi à toutes les personnes que j'ai fréquentées, toutes celles qui m'ont accompagné et qui m'ont apporté quelque chose durant ces années. Mes amis de l'aventure *The Geeks and The Girls* ainsi que de l'aventure *Acoustica* se reconnaîtront. Malheureusement tout ça a du s'arrêter à un moment, mais ça m'a permis de me ressourcer dans la musique quand le travail était trop envahissant, et ça m'a surtout permis de rencontrer des amis inoubliables. Merci à toute la bande du Nautilus, en particulier ses gérants et employés, qui ont créé cet endroit magique où je peux m'offrir un bol d'air frais quand j'en ai besoin.

Enfin, même s’ils ne liront pas ces mots, merci à tous les artistes dont j’ai écouté les œuvres en travaillant.

Je dédie cette thèse à mon père,
qui aurait sans doute dit "Very well, my Lord",
en voyant dans quelle langue je l'ai écrite.

Contents

Appendix

Introduction

First things first, but not necessarily in that order.
The Doctor – *Doctor Who - Meglos*

1 Non-Monotonic Reasoning in Artificial Intelligence: the Role of Argumentation and Belief Change

Artificial intelligence (AI) is the topic of computer science which aims at computing with machines some complex tasks which require some reasoning abilities, usually associated with human beings. In particular, the thematic of non-monotonic reasoning (NMR) gathers numerous approaches to deal with pieces of information which may contradict the rules of classical logic. The most famous example to illustrate and motivate researches in this domain is Tweety example. If Tweety is a bird, then Tweety can fly. However, we know that penguins are birds, and that they cannot fly, and it is the case that Tweety is a penguin. Using classical logic, we deduce from these sentences that Tweety can fly and Tweety cannot fly, which is not consistent. Numerous ways to model this kind of problem have been pointed out. For instance, default logic [Rei80] allows to define the sentence "**In general**, birds can fly", as it is the case with most of the non-monotonic reasoning inference relations [KLM90]. So, the fact that Tweety is a penguin is an exception to the general case, and we can deduce without any contradiction that Tweety cannot fly.

In this thesis, we are interested in a specific non-monotonic reasoning setting: argumentation, especially abstract argumentation [Dun95]. Dung defines an argumentation framework as a set of abstract entities, called *arguments*, representing any piece of information (beliefs, actions to be performed,...), linked with some *attacks*, which indicate the existence of conflicts between the arguments. Conflicts are generally oriented, so that an attack from argument a_i to argument a_j means that if a_i is accepted, then a_j should be rejected. From such an argumentation framework, some sets of arguments which can be considered as accepted can be deduced (for instance, the agent's beliefs or some actions to be done). There are called *extensions*. In the case of Tweety, we can model the situation with two arguments: $a_1 =$ "Tweety is a bird, so it can fly"; $a_2 =$ "Tweety is a penguin, and penguin cannot fly", linked by an attack from a_2 to a_1. Then, reasoning with this argumentation framework, we can accept the argument a_2, and deduce that Tweety cannot fly.

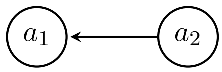


Figure 1: The Tweety example

This setting is particulary interesting in a multiagent scenario: when several agents share some pieces of information, even if each of them is consistent are considered separately, their dialog can lead to gather some conflicting pieces of information. It is even more often the case when the agents have some divergent goals. A very usual example is the one concerning a group of friends who try to choose an activity for the evening: John wants to watch a movie, Paul prefers to go to a good restaurant, and George prefers to go to a concert. Ringo, at last, is exhausted and wants to go home without looking like he is neglecting his friends. Each of John, Paul and George has to state some arguments to convince the other ones that his solution is the one which should be accepted, while Ringo has to convince them that none of these solutions is good. There are numerous works on argumentative dialog protocols [KMM04, AH06, KBM$^+$13, KBMM14].

Another setting related to non-monotonic reasoning is belief revision. It is quite usual, for an individual, to have to change her beliefs about the world when some reliable new piece of information contradicts her previous beliefs. Using again Tweety example, if an ornithologist tells us "This bird cannot fly", then we can suppose that this information is reliable. So it is reasonable to change our beliefs to deduce that Tweety cannot fly. Belief revision approaches, as well as other kind of belief change operations, have been characterized in the 1980s and 1990s when the agent's beliefs are represented in a logical setting [AGM85, KM91, KM92]: some rationality postulates have been stated (axiomatic characterization), and methods to define exactly the change operators which satisfy the postulates have been identified (constructivist characterization).

2 Research Question

Recently, the researchers working on argumentation have studied a question which is a kind of bridge between belief change and argumentation: the *dynamics of argumentation frameworks*. The aim of these first studies is to incorporate a new piece of information in an argumentation framework, sometimes borrowing some properties to belief change theory, like minimal change or success principle. But none of them has deeply studied the links between belief change and argumentation.
For instance, the first works on this topic are concerned with some theoretical properties of change operators in argumentation frameworks [BKvdT09b, BKvdT09a, CdSCLS10, BCdSCLS11]. These works study what happens in the argumentation framework when some new argument is added to the framework.
Another approach of dynamics of argumentation frameworks is enforcement [BB10, Bau12]. It has been defined as change in an argumentation framework to ensure that a given set of arguments is an extension. Existing enforcement operators are useful when the scenario is a classical debate. The debate players state their arguments, one after the other, with the attacks concerning them, without modifying the existing attack relation between the arguments which have been previously stated. The aim of enforcement is to know how some new argument must be added to satisfy some goal concerning the arguments acceptance.

In this thesis, we tackle several aspects of the dynamics of argumentation frameworks. First of all, we propose to study the impact of a new piece of information about the arguments statuses, like it is the case with enforcement approaches. But contrary to enforcement, we do not want to express that "this particular set of arguments should be an extension", we want to be able to incorporate more complex information about the arguments statuses. For instance, an agent should be able to incorporate to her beliefs (represented by the argumentation framework and its extensions) an information such that "this particular argument must not be accepted, unless if this other one is accepted", or "these two arguments must be accepted together, while this third one is not". This is while we have been, in a first time, interested in belief change: *is it possible to use the AGM framework, suited to logical settings, in the setting of abstract argumentation? And if it is the case, how can we use it to revise argumentation frameworks by such complex pieces of information?* These questions are at the origin of our two first contributions.

Even if they are sometimes discussed and questionned [Fer99], the properties from the AGM framework are accepted by the community of researchers on belief change as the foundation of most work on rational change. So it seems very useful to adapt these properties to abstract argumentation. This adaptation particulary makes sense when an agents considers that the extensions of the argumentation framework, which express the acceptance statuses of arguments, are the most important pieces of information for her, rather than the structure of the argumentation framework. Then, it is possible to make a parallel between the classical model-based approach from the AGM framework (where the belief change operators are represented as a ranking between propositional interpretations) and an extension-based approach to revise argumentation frameworks, where the argumentation framework is revised by a propositional formula which expresses some information about arguments statuses.

Revision of argumentation frameworks can benefit from the AGM framework in another way. Indeed, since it is well-known in logical settings, we can benefit from existing work on the logical encodings of argumentation frameworks to represent an argumentation framework revision as a three-step process: encoding the argumentation framework in a logical setting, then using classical revision in the logical setting, and finally decoding the result of the revision to obtain the result of the revision. Depending on the exact encoding which is used, the nature of the revision formulae may be different. We propose to study a propositional encoding which makes the link between the structure of the argumentation framework (the attacks) and the skeptically accepted arguments (which are the arguments belonging to every extension).

As we explained previously, enforcement as it has been studied until now is really useful in a debate-like scenario: several agents discuss a topic and add some new arguments to support their point of view. However, every scenario related to argumentation does not share this characteristic. For instance, an interesting scenario concerns a single agent use of argumentation to decide between different pieces of information she received, possibly inconsistent due to the divergence of the sources of information. In this case, it is conceivable that the agent receives a new piece of information which is not a new argument, but an information such that "a given set of arguments should be an extension". In this case, which is related to belief revision, there is no need of new arguments, as it is for Baumann and Brewka's enforcement. It is perfectly sensible to question the attacks between the arguments previously known. We study the problem of enforcing a set of arguments, highlighting some limitations of the existing enforcement operators, and we present some new ones which are more satisfying in our scenario. To do so, we apply a method similar to our second revision approach: the use of propositional encodings to represent the argumentation framework and the semantics. Then, performing an enforcement is equivalent to solving a satisfaction or an optimization problem.

With the contributions described previously, we propose several new families of change operators for argumentation frameworks, more or less inspired by belief revision operators. These new operators allow to tackle new scenarios of the dynamics of argumentation frameworks, giving to the agent the capability to incorporate different kinds of complex information to her beliefs; moreover these complex informations can be incorporated in many different ways, depending on the properties that the agent expects from the change operator.

3 Organization of the Thesis

We present in **Part I (State of the Art)** the notions required for a good understanding of our work.
Chapter 1 (Abstract Argumentation) presents the bases of the theory of abstract argumentation. We focus on Dung's framework [Dun95]. We give a brief overview of the main acceptability semantics and a presentation of some logical encodings defined by [BD04], which are the bases of some of our works.
Chapter 2 (Belief Change) is a detailed presentation of the AGM framework [AGM85]: rationality postulates for the different belief change operations are given, together with some representation theorems. Then we focus on the adaptation of this framework to belief change in propositional logic [KM91, KM92]. This particular version of the AGM framework is used in our works.
Chapter 3 (Existing Approaches on Dynamics of Argumentation Frameworks) describes some of the numerous aspects of change in argumentation frameworks which have been studied in the recent years.

In this thesis, we consider change in argumentation frameworks under some new points of view, which are presented in **Part II (Contributions to the Dynamics of Argumentation Frameworks)**. First of all, we study different ways to benefit from the well-known belief revision operators from logical settings. Our methods ensure that our revision operators satisfy some desirable properties which are usual for belief revision: primacy of update, consistency and minimal change.
Our first approach is developped in **Chapter 4 (Adapting the AGM Framework for Abstract Argumentation)**. It consists of a faithful adaptation of the AGM framework, especially the version suited to finite propositional logic proposed by Katsuno and Mendelzon [KM91]. It is well suited for some scenarios when the new piece of information only concerns the agents beliefs, meaning the acceptance statuses of arguments. Rationality postulates ensure that the revision operators satisfy some expected properties. These revision operators do not revise directly the argumentation framework, but they perform a two step process. As a first step, the "beliefs" of the agent are revised, represented by the set of extensions of the argumentation framework. Then, a second step generates the revised argumentation frameworks from the revised extensions.
Chapter 5 (AGM Revision as a Tool to Revise Argumentation Frameworks) presents a translation-based approach using logical encodings to revise argumentation frameworks. In this approach, information about the structure of the framework (the attack relation) and its semantics (the arguments acceptance statuses) are encoded into a logical formula. Then, we use revision operators from the AGM framework (especially, the Katsuno and Mendelzon's version) on this encoding. A decoding step allow to obtain the revised argumentation frameworks from the revised formula. This translation-based method has the advantage to allow argumentation frameworks to change because new pieces of information about arguments acceptance statuses and about the attacks can be considered simultaneously.
Our third contribution, described in **Chapter 6 (Extension Enforcement)** concerns the notion of enforcement of a set of arguments in an argumentation framework. We study the properties of these enforcement operators, and we are interested in some of their limitations. We prove that there are some situations when strict enforcement of a set of arguments is impossible with the existing enforcement approaches. Moreover, even if the existing enforcement operators are well-suited to most debate-like argumentation scenarios, they are not sufficient for some scenarios. Especially, when the agent does not have some new arguments available. We present some new operators which ensure that an agent can perform an enforcement in any situation without any possibility of failure. Some links with our second contribution on belief revision appear. In particular, we use some logical encodings of enforcement operators (the ones previously defined by Baumann and Brewka, and the new ones that we introduce) to translate them into some constraint optimization problems. These well-known problems can be solved with some very efficient methods, so we can benefit from the power of some state-of-the-art constraint

solvers to implement enforcement operators.
Last, **Chapter 7 (On Constraints and Change in Argumentation)** presents a categorization of the different kinds of constraints which can be enforced in argumentation frameworks, and which kinds of change must be performed to satisfy these constraints. Each kind of constraints is exemplified with the existing approaches about dynamics of argumentation frameworks which correspond to it. We also sketch some new kinds of constraints which have not been considered until now, leading to some interesting tracks for future research. Dung's argumentation framework using an extension-based or a labelling-based semantics has been the main study cases in the existing work, but we also point out some original ideas to enforce a constraint in some enrichments of Dung's framework, showing that our typology is general enough to work with any kind of abstract argumentation framework. We present a first step to define a generalized family of constraint enforcement operators through logical encodings.

The **Conclusion** of the thesis points out some very interesting future works which are related to the contributions described in this document, and also some completely new research tracks which have not been explored until now.

The appendices present some additional material. First, **Appendix A (Background Notions)** introduces some concepts concerning set-theoretical relations, propositional logic and graph theory. A more substantial part of this chapter is the introduction to complexity theory, which is required to evaluate the computational hardness of the tasks that we are studying. This part is not mandatory if the reader is already familiar with these notions, but it can help a non-expert reader to understand some mathematical concepts. Many notations and writing conventions are defined in this appendix. The other parts of the appendices present the proofs of the propositions, which are separated from the contribution chapters for a matter of readability.

Part I

State of the Art

Chapter 1

Abstract Argumentation

The purpose of argument is to change the nature of truth.
Frank Herbert – *Children of Dune*

Argumentation is a field of research particulary interesting for the Artificial Intelligence community. It concerns the ability, for an agent (a human being, a machine, a piece of software,. . .), to understand a topic, to reason about it, taking into account every (possibly contradictory) piece of information on this topic. Of course, argumentation is also a way to convince another agent to change her mind. In both cases, argumentation can be a tool for decision making. More generally, argumentation can be used to model any kind of reasoning, as soon as there is some notion of *incompatibility* between the data of the problem.

In this chapter, we introduce the basic notions of abstract argumentation as presented by [Dun95]. We explain the different ways to deduce some information from an argumentation framework, defined by acceptability semantics. These semantics allow to generate some sets of arguments, called extensions, which can be accepted together. Extensions can be refined through the notion of labellings [Cam06], which map each argument in the framework to a label indicating if it has to be accepted, rejected, or if the agent cannot decide between the first two options. We explain then the approach from [BD04] to map each pair of an argumentation framework and a semantics to a propositional formula. This kind of encoding is at the core of some of our contributions.

Lastly, we present some examples of applications of argumentation frameworks, like the case of decision making [ADM08] and goal-oriented persuasion [BMM14]. To conclude, we present an example of application of Dung's framework which concerns an unusual scenario: resource allocation.

Contents

1.1 Dung's Framework

There is not a single formalization of argumentation. Dung's abstract framework focuses on the relations between arguments exchanged by the agents rather than their origin and how they are built. The question here is not to know how to obtain the arguments, but to know how to define their acceptance status. This section aims at presenting the main approaches to determine the statuses of arguments.

1.1.1 Argumentation Framework and Acceptability of Arguments

In [Dun95], Dung formalizes argumentation through an abstract framework which focuses on the relation between arguments rather than their nature. An argumentation framework is composed of two kinds of elements:

- *arguments* are abstract entities representing some pieces of informations which may be conflicting with each other;
- *attacks* are pair of arguments which are conflicting; such a pair is oriented, meaning that the first argument is the one which attacks the second one.

Formally:

Definition 1 (Abstract Argumentation Framework)**.**
Given a set A of arguments and a binary relation R on A ($R \subseteq A \times A$), we call the pair $\langle A, R \rangle$ an (abstract) argumentation framework (AF).
Given two argumentation frameworks $F = \langle A, R \rangle$ and $F' = \langle A', R' \rangle$, we use the following notations:

- $F \sqsubseteq F'$ if and only if $A \subseteq A'$ and $R \subseteq R'$
- $F \sqcup F' = \langle A \cup A', R \cup R' \rangle$

We suppose in this thesis that A is finite.

Intuitively, if a_i and a_j are two elements from A, $(a_i, a_j) \in R$ means that if the agent considers that a_i can be accepted, then the argument a_j should be rejected. Argumentation frameworks are naturally represented as digraphs, the nodes being the arguments and the edges representing the attacks.

Example 1 (Example borrowed from [Tou58])**.**
For instance, we have three pieces of information:

(a) Harry was born in Bermuda, so he is a British citizen.

(b) Harry lives in United States, he may have received the US nationality.

(c) Harry has obtained a Green Card, so he can live in United States without the US nationality.

The first piece of information can be represented as an argument a. It is attacked by the argument b: Harry may have US nationality, if it is the case, he is not (not any longer) a British citizen. However, a third piece of information c attacks b, and so *defends* a against b.
The graph corresponding to these arguments is given at Figure 1.1.

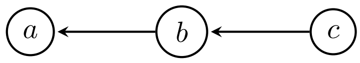


Figure 1.1: Example of Argumentation Framework as a Digraph

We define formally the notions of attack and defense:

Definition 2 (Attack, Defense)**.**
Let $F = \langle A, R \rangle$ be an argumentation framework.

- Let $a_i \in A$ be an argument. We say that $a_j \in A$ (respectively the set of arguments $S \subseteq A$) attacks a_i if $(a_j, a_i) \in R$ (respectively if $\exists a_j \in S$ such that a_j attacks a).
- Let $a_i, a_j \in A$ be two arguments such that a_j attacks a_i. We say that argument $a_k \in A$ (respectively the set of arguments $S \subseteq A$) defends a_i against a_j if a_k (respectively S) attacks a_j.

Generally speaking, it is not easy to deduce from an argumentation framework which arguments can be accepted. For this purpose, Dung defines several notions allowing to select "good" sets of arguments. Depending on the properties expected to be satisfied by these good sets, we define different *acceptability semantics*, each of them given a (possibly) different set of *extensions* as the outcome of the argumentation process.

The first of these interesting properties is conflict-freeness: it is desirable to accept together some arguments if they do not attack each other.

Definition 3 (Conflict-free Set)**.**
Let $F = \langle A, R \rangle$ be an argumentation framework. The set of arguments $S \subseteq A$ is *conflict-free* in F if and only if $\forall a \in A, \forall b \in A, (a, b) \notin R$. The set of all conflict-free sets of arguments in F is denoted $cf(F)$.

Example 1 Continued.
In the argumentation framework, the set $\{a, c\}$ is conflict-free (neither (a, c) nor (c, a) belong to the attack relation), but $\{a, b, c\}$ is not conflict-free (because of the attacks (b, a) and (c, b)).

Dung extends the notion of defense to define the acceptability of an argument with respect to a set of arguments:

Definition 4 (Acceptable Argument)**.**
Let $F = \langle A, R \rangle$ be an argumentation framework. An argument $a \in A$ is *acceptable* with respect to $S \subseteq A$ in F if and only if for each argument $b \in A$ such that b attacks a, S attacks b.

Example 1 Continued.
In the previous argumentation framework, argument a is acceptable with respect to $S = \{c\}$ since S defends a against each attack (here, the attack from b to a).

These two notions are used to define admissibility. Admissibility is usually the minimal condition to accept a set of arguments.

Definition 5 (Admissible Set)**.**
Let $F = \langle A, R \rangle$ be an argumentation framework. A set of arguments $S \subseteq A$ is *admissible* in F if and only if S is conflict-free in F, and for each argument $a \in S$, a is acceptable with respect to S in F.

Example 1 Continued.
In the argumentation framework from the previous example, the set $S = \{a, c\}$ is conflict-free, and a and c are acceptable with respect to S, so S is admissible.

If it is true that admissibility is required for an agent to accept a set of arguments (to avoid accepting conflicting pieces of information – conflict-freeness – or some pieces of information unable to defend themselves against attacks – acceptability), it is not always enough. Dung's semantics refine admissibility to chose admissible sets which satisfy some additional properties.

Definition 6 (Extensions)**.**
Let $F = \langle A, R \rangle$ be an argumentation framework.

- The set of arguments $S \subseteq A$ is a complete extension of F if and only if it is an admissible set of F, and for each argument a which is acceptable with respect to S, $a \in S$.
- The set of arguments $S \subseteq A$ is a preferred extension of F if and only if S is maximal (with respect to set-theoretical inclusion $\subseteq$) among admissible sets of F.
- The set of arguments $S \subseteq A$ is a stable extension of F if and only if it is conflift-free in F and S attacks each argument $a \in A \backslash S$.
- The set of arguments $S \subseteq A$ is a grounded extension of F if and only if it is a minimal element (with respect to set-theoretical inclusion $\subseteq$) among the complete extensions of F.

In this thesis, we use the abbreviations co, pr, st, gr to denote respectively the complete, preferred, stable and grounded semantics. Some works consider that admissibility is enough to define a semantics. Then, ad denotes the admissible semantics.
We consider only semantics which satisfy admissibility: the set of extensions must be a subset of the admissible sets of the argumentation framework.

It is also known that preferred extensions are exactly the maximal elements (with respect to $\subseteq$) among the complete extensions. Other acceptability semantics have been defined;[1] we do not introduce them here because we focus on the four classical Dung's semantics in the rest of this thesis, but all our work can be adapted to any semantics. Given a semantics σ, $Ext_\sigma(F)$ denotes the set of σ-extensions of the argumentation framework F.

Dung has presented some interesting properties for his semantics:

Proposition 1 ([Dun95])**.**

1. *Every argumentation framework possesses at least one preferred extension.*
2. *Every argumentation framework possesses a single grounded extension.*
3. *Each stable extension of an argumentation framework F is a preferred extension of F (but the converse is not true).*
4. *Each preferred extension of an argumentation framework F is a complete extension of F (but the converse is not true).*

We can represent inclusions between semantics as in Figure 1.2. $X \longrightarrow Y$ means: for all argumentation framework F, $Ext_X(F) \subseteq Ext_Y(F)$.

[1] For instance, see [DMT07] for the ideal semantics and [Cam07] for the eager semantics.

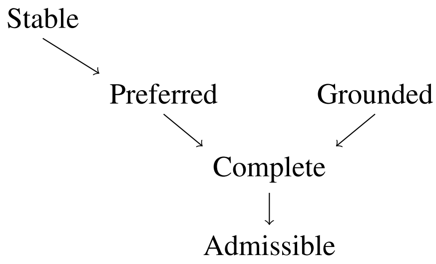


Figure 1.2: Inclusion between Dung's Semantics

In the general case, these semantics give different sets of extensions, as we can see on the following example.

Example 2 (Different Extensions for Different Semantics)**.**
Let F be the argumentation framework given on Figure 1.3.

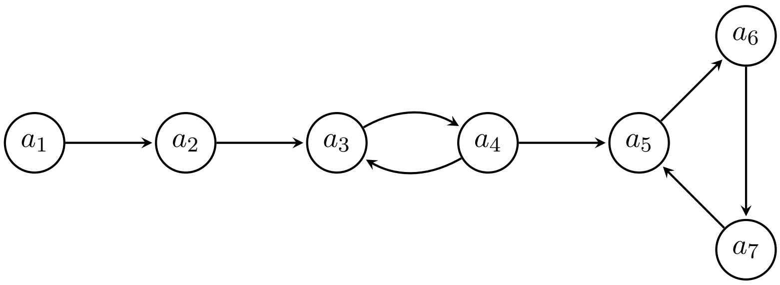


Figure 1.3: An Argumentation Framework with Non-Matching Semantics

The extensions for the different semantics are:

- $Ext_{gr}(F) = \{\{a_1\}\}$
- $Ext_{st}(F) = \{\{a_1, a_4, a_6\}\}$
- $Ext_{pr}(F) = \{\{a_1, a_3\}, \{a_1, a_4, a_6\}\}$
- $Ext_{co}(F) = \{\{a_1\}, \{a_1, a_3\}, \{a_1, a_4, a_6\}\}$

In some cases, different semantics match with each other. Dung has given some sufficient conditions for some semantics to coincide:

Definition 7 (Well-Founded Argumentation Framework)**.**
Let $F = \langle A, R \rangle$ be an argumentation framework. F is well-founded if and only if there is no infinite sequence of arguments $a_0, a_1, \dots, a_k, \dots$ from A such that $\forall i \in \mathbb{N}, (a_i, a_{i+1}) \in R$. [2]

Proposition 2.
Every well-founded argumentation framework has exactly one extension which is grounded, stable, preferred and complete.

This particular case allows to decide which arguments are accepted through a simple algorithm;

- accept the unattacked arguments;

[2]For finite argumentation framework, it corresponds to acyclicity.

- accept the arguments which are defended by the accepted arguments, reject the ones which are attacked by an accepted argument;
- iterate until each argument is accepted or rejected.

Example 3.
Let us consider the argumentation framework represented on Figure 1.4. a and b are not attacked, so they

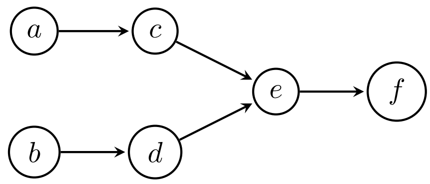


Figure 1.4: A Well-Founded Argumentation Framework

are accepted. Consequently, c and d must be rejected. It allows e to be accepted, since it is defended against all its attackers. So, f is rejected, due to the attack from e. Hence, the single extension for all the semantics is $\{a, b, e\}$.

Let us continue this part on background of argumentation with a presentation of labellings. This notion is a refinement of the notion of extension. Rather than associating each argument a status "accepted" or "rejected", a labelling determines if the argument is accepted (in), rejected (out), or none of them ($undec$). Intuitively, the difference between out and $undec$ is that, in the first case, the agent has a good reason to reject the argument; while in the second case, the agent cannot accept the argument, but she does not have a good reason to reject it either.
Such a mapping from arguments to labels makes sense if it takes into account the attack relation (for instance, labelling in two arguments which attack each other does not make sense, nor labelling out an argument which is not attacked).
The notion of reinstatement labelling allows to ensure that the mapping takes the attack relation into account: an argument is in if and only if all its attackers are out; an argument is out if and only if it is attacked by at least on argument in; it is $undec$ in the remaining case.

Definition 8 (Labelling, Reinstatement Labelling [Cam06])**.**
Let $F = \langle A, R \rangle$ be an argumentation framework.

- L is a labelling of F if and only if L is a mapping from A to $\{in, out, undec\}$.
- The labelling L is a reinstatement labelling of F if and only if
  - $\forall a \in A$, $L(a) = out$ if and only if $\exists b \in A$ such that $(b, a) \in R$ and $L(b) = in$;
  - $\forall a \in A$, $L(a) = in$ if and only if $\forall b \in A$ such that $(b, a) \in R$, $L(b) = out$.

L can also be represented by the set of pairs $\{(a, L(a)) \mid a \in A\}$.

Such a reinstatement labelling L can be used to partition the set of arguments A:

- $in(L) = \{a \in A \mid L(a) = in\}$
- $out(L) = \{a \in A \mid L(a) = out\}$

- $undec(L) = \{a \in A \mid L(a) = undec\}$

Caminada proved that there is a matching between reinstatement labellings and complete extensions. Indeed, each complete extension can be built from a reinstatement labelling L keeping only the arguments which are in in L; conversely, a labelling can be built from an extension ε: the arguments in ε are in, while the arguments attacked by ε are out and the remaining ones are $undec$.

Proposition 3 (Matching between Reinstatement Labellings and Complete Extensions [Cam06])**.**
Let $F = \langle A, R\rangle$ be an argumentation framework.

- *Let L be a reinstatement labelling of F. $E(L) = in(L)$ is a complete extension of F.*
- *Let ε be a complete extension of F. We denote*

$$\begin{aligned} &IN = \{(a, in) \mid a \in \varepsilon\} \\ &OUT = \{(a, out) \mid a \textit{ is attacked by } \varepsilon\} \\ &UNDEC = \{(a, undec) \mid a \notin \varepsilon \textit{ and } a \textit{ is not attacked by } \varepsilon\} \end{aligned}$$

  $Lab(\varepsilon) = IN \cup OUT \cup UNDEC$ is a reinstatement labelling of F.

So, reinstatement labellings match complete extensions in a bijective way. Caminada showed that such relations exist between extensions for the other semantics and some particular families of reinstatement labellings. For this reason, we use σ-labellings in the rest of this document to name the labellings corresponding to the σ-extensions of an argumentation framework, and reinstatement labellings are called complete labellings.

Proposition 4 (Matching between Labellings and σ-Extensions [Cam06])**.**
Let $F = \langle A, R\rangle$ be an argumentation framework. Given ε a σ-extension of F,

- *if σ is the stable semantics, $L(\varepsilon)$ is a complete labelling such that $undec(L) = \emptyset$;*
- *if σ is the preferred semantics, $L(\varepsilon)$ is a complete labelling such that $in(L)$ is maximal;*
- *if σ is the grounded semantics, $L(\varepsilon)$ is a complete labelling such that $in(L)$ is minimal.*

Given L a complete labelling of F,

- *if $undec(L) = \emptyset$, then $E(L)$ is a stable extension of F;*
- *if $in(L)$ is maximal, then $E(L)$ is a preferred extension of F;*
- *if $in(L)$ is minimal, then $E(L)$ is the grounded extension of F.*

For each semantics σ, $Labs_\sigma(F)$ denotes the set of labellings associated to the argumentation framework with respect to σ.

Example 2 Continued.
Let us consider again the argumentation framework given at Figure 1.3. The labellings for the different semantics are:

- $Labs_{gr}(F) = \{L_1^{gr}\}$ with

$$in(L_1^{gr}) = \{a_1\}; out(L_1^{gr}) = \{a_2\}; undec(L_1^{gr}) = \{a_3, a_4, a_5, a_6, a_7\}$$

- $Labs_{st}(F) = \{L_1^{st}\}$ with

$$in(L_1^{st}) = \{a_1, a_4, a_6\}; out(L_1^{st}) = \{a_2, a_3, a_5, a_7\}; undec(L_1^{st}) = \emptyset$$

- $Labs_{pr}(F) = \{L_1^{pr}, L_2^{pr}\}$ with

$$in(L_1^{pr}) = \{a_1, a_3\}; out(L_1^{pr}) = \{a_2, a_4\}; undec(L_1^{pr}) = \{a_5, a_6, a_7\}$$
$$in(L_2^{pr}) = \{a_1, a_4, a_6\}; out(L_2^{pr}) = \{a_2, a_3, a_5, a_7\}; undec(L_2^{pr}) = \emptyset$$

- $Labs_{co}(F) = \{L_1^{co}, L_2^{co}, L_3^{co}\}$ with

$$in(L_1^{co}) = \{a_1\}; out(L_1^{co}) = \{a_2\}; undec(L_1^{co}) = \{a_3, a_4, a_5, a_6, a_7\}$$
$$in(L_2^{co}) = \{a_1, a_3\}; out(L_2^{co}) = \{a_2, a_4\}; undec(L_2^{co}) = \{a_5, a_6, a_7\}$$
$$in(L_3^{co}) = \{a_1, a_4, a_6\}; out(L_3^{co}) = \{a_2, a_3, a_5, a_7\}; undec(L_3^{co}) = \emptyset$$

The very value of labellings is to be able to distinguish, among the arguments which do not belong to an extension (those which are not in), those which are "really" rejected (out) from those which are not formally rejected, but are not reliable enough to be accepted ($undec$). This can be useful to define more expressive distances for instance.

1.1.2 Inference Tasks and their Complexity

Several inference tasks can interest an agent who reasons with argumentation frameworks. The most usual one is to determine whether an argument is accepted or not. We can distinguish between two kinds of acceptance. Given an argumentation framework $F = \langle A, R \rangle$:

Credulous Acceptance (`DC`) Given $a \in A$ an argument and σ a semantics does it exist some $\varepsilon \in Ext_\sigma(F)$ such that $a \in \varepsilon$?

Skeptical Acceptance (`DS`) Given $a \in A$ an argument and σ a semantics, does $\forall \varepsilon \in Ext_\sigma(F)$, $a \in \varepsilon$ hold?

Intuitively, the credulously accepted arguments are the ones which are supported by at least one extension. They can be accepted if the agent does not need an absolute certainty about their status. On the opposite, the skeptically accepted arguments[3] are the arguments which belong to each extension, and then cannot be questionned. Both inference policies have some weakness, depending on the particular application where they are used. When using credulous acceptance, the agent may accept two arguments a_1 and a_2 even if they are conflicting, as soon as each of them belongs to an extension. On the opposite, using the skeptical acceptance, the agent may reject every argument, since it is not always the case that an argument belongs to each extension.
The set of credulously accepted arguments and the set of skeptically accepted arguments are respectively defined like this:

$$Cr_\sigma(F) = \bigcup_{\varepsilon \in Ext_\sigma(F)} \varepsilon$$

$$Sc_\sigma(F) = \bigcap_{\varepsilon \in Ext_\sigma(F)} \varepsilon$$

Some other interesting inference problems have been defined:

[3] We also call them " skeptical consequences" of an argumentation framework.

Existence (`Exist`) Given σ a semantics, does it exist a σ-extension of F?

Non Trivial Existence ($\texttt{Exist}_\emptyset$) Given σ a semantics, does it exist a non-empty σ-extension of F?

Verification (`Ver`) Given $e \subseteq A$ a set of arguments and σ a semantics, does $e \in Ext_\sigma(F)$ hold?

Studies have been conducted to identify the complexity of these problems for different semantics.

Proposition 5 ([CDM05, DW09]).
The complexity of DC, DS, Exist, $\texttt{Exist}_\emptyset$ *and* Ver, *for the grounded, stable, preferred and complete semantics, are the ones given at Table 1.1. Given a complexity class* C, C$-c$ *means that the problem is complete for the class* C. *"Trivial" means that the solution of the problem is known from the definition of the semantics.*

Semantics	gr	st	pr	co
DC	P	NP$-c$	NP$-c$	NP$-c$
DS	P	coNP$-c$	$\Pi_2^P - c$	P$-c$
Exist	Trivial	NP$-c$	Trivial	Trivial
$\texttt{Exist}_\emptyset$	P	NP$-c$	NP$-c$	NP$-c$
Ver	P	P	coNP$-c$	P

Table 1.1: Complexity of Inference Problems for the Usual Semantics

1.1.3 Propositional Encoding of Argumentation Frameworks

In this section, we present some results from [BD04] on the encoding, through propositional logic, of the most usual semantics. The main idea is to check, given an argumentation framework F, a semantics σ and a set of arguments E, if E is a σ-extension of F. Three different approaches are presented. The first one consists in verifying that the set E satisfies some particular equation. We do not explain this approach in details.

We present more in depth the second approach, which checks if a particular propositional formula, depending on the parameters F, σ and E, is satisfiable. Besnard and Doutre call this approach "Satisfiability Checking Approach". The Boolean variables defined in this encoding correspond to the arguments in the set $A = \{a_1, \dots, a_n\}$, and their intuitive meaning is that argument a_i is accepted if the associated Boolean variable is $true$.

Let us illustate this method on a simple example: conflict-freeness. First, we recall that a set $E \subseteq A$ is conflict-free in $F = \langle A, R \rangle$ if and only if there is no pair of arguments $c = (a_i, a_j)$ such that both arguments belong to E and c belongs to the attack relation R. So, we can define the propositional formula $\Psi^E_{F,cf}$ which is satisfiable if and only if E is conflict-free in F:

$$\Psi^E_{F,cf} = \bigwedge_{a_i \in E} [a_i \wedge (\bigwedge_{a_j : (a_j, ai) \in R} \neg a_j)]$$

An example may help to understand:

Example 4.
Let $F = \langle A, R \rangle$ be the argumentation framework given on Figure 1.5. Given the set of arguments $E_1 = \{a_1, a_3\}$ (which is not conflict-free), we build the propositional formula $\Psi^{E_1}_{F,cf} = [a_1 \wedge (\neg a_2 \wedge$

$\neg a_3)] \wedge [a_3 \wedge (\neg a_4)]$, which is obviously not satisfiable.
The conflict-free set $E_2 = \{a_1, a_4\}$ is associated with the formula $\Psi^{E_2}_{F,cf} = [a_1 \wedge (\neg a_2 \wedge \neg a_3)] \wedge [a_4]$, which is satisfiable. For instance, the valuation of a_1 and a_4 as $true$; a_2 and a_3 as $false$ is a model.

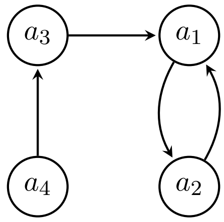


Figure 1.5: The Argumentation Framework F

Similarly, Besnard and Doutre presented encodings for the stable, complete and admissible semantics: E is a σ-extension of F is and only if $\Psi^{E}_{F,\sigma}$ is satisfiable.
This approach is extended in [BDH14], which gives the required tools to define $\Psi^{E}_{F,\sigma}$ when the semantics σ is defined through a notion of set-theoretical minimality or maximality (as it is the case of preferred and grounded semantics).

Last, we finish this presentation of logical encodings by the third approach proposed by Besnard and Doutre: the "Model Checking Approach". Now, the propositional formula $\Phi_{F,\sigma}$ such that each of its models corresponds to an extension of F is built. So, for the verification problem, propagating the truth values of Boolean variables allows to know if a particular set of arguments is a σ-extension.
We can also consider some other applications of this approach: thanks to the power of modern SAT solvers, computing an extension – or even the set of all the extensions – of an argumentation framework can often be done very efficiently.

We give here the details about the encodings for the stable and complete semantics, and we illustrate them on a concrete example.

Proposition 6 ([BD04]).
Let $F = \langle A, R\rangle$ be an argumentation framework. A set of arguments $E \subseteq A$ is a stable extension of F if and only if E is a model of the formula

$$\Phi_{F,st} = \bigwedge_{a_i \in A} [a_i \Leftrightarrow (\bigwedge_{a_j : (a_j, ai) \in R} \neg a_j)]$$

Example 4 Continued.
Let us come back to the argumentation framework described at Figure 1.5. The encoding of the stable semantics for this argumentation framework gives:

$$\begin{array}{rlcl} \Phi_{F,st} = & [a_1 & \Leftrightarrow & (\neg a_2 \wedge \neg a_3)] \\ & \wedge[a_2 & \Leftrightarrow & (\neg a_1)] \\ & \wedge[a_3 & \Leftrightarrow & (\neg a_4)] \\ & \wedge[a_4 & \Leftrightarrow & (true)] \end{array}$$

It is easy to check that its models are exactly

$$\mathrm{Mod}(\Phi_{F,st}) = \{\{a_1, a_4\}, \{a_2, a_4\}\}$$

which correspond to the stable extensions of F.

Proposition 7 ([BD04]).
Let $F = \langle A, R\rangle$ be an argumentation framework. A set of arguments $E \subseteq A$ is a complete extension of F if and only if E is a model of the formula

$$\Phi_{F,co} = \bigwedge_{a_i \in A} [(a_i \Rightarrow (\bigwedge_{a_j:(a_j,ai)\in R} \neg a_j)) \wedge (a_i \Leftrightarrow (\bigwedge_{a_j:(a_j,ai)\in R} \bigvee_{a_l:(a_l,aj)\in R} a_l))]$$

Example 4 Continued.
Continuing the previous example, the encoding of the complete semantics for this argumentation framework F gives:

$$\begin{aligned}\Phi_{F,co} = \quad & [(a_1 & \Rightarrow & \quad (\neg a_2 \wedge \neg a_3)) \wedge (a_1 \Leftrightarrow (a_1 \wedge a_4))] \\ & \wedge[(a_2 & \Rightarrow & \quad (\neg a_1)) \wedge (a_2 \Leftrightarrow (a_2)] \\ & \wedge[(a_3 & \Rightarrow & \quad (\neg a_4)) \wedge (a_3 \Leftrightarrow (\mathit{false})] \\ & \wedge[(a_4 & \Rightarrow & \quad (\mathit{true})) \wedge (a_4 \Leftrightarrow (\mathit{true})]\end{aligned}$$

This time, the models are

$$\mathrm{Mod}(\Phi_{F,st}) = \{\{a_1, a_4\}, \{a_2, a_4\}, \{a_4\}\}$$

which correspond to the complete extensions of F.

Besnard and Doutre also pointed out some encodings for the preferred and grounded semantics, but there is not exact correspondance between the extensions and the models of the formula. This comes from the notion of minimality and maximality which define these semantics. The authors presented an encoding of admissible sets of arguments (we note this encoding $\Phi_{F,ad}$). A set E is then a preferred extension of the argumentation framework F if and only if E is a maximal model (with respect to $\subseteq$) of the formula $\Phi_{F,ad}$.
In a similar way, E is the grounded extension of the argumentation framework F if and only if it is the minimal model of $\Phi_{F,co}$ with respect to inclusion.

1.2 Realizability of a Set of Candidates

A last notion which is related to the contributions presented in this thesis is the realizability of a set of sets of arguments called *extension-sets*[4][DDLW14].

Definition 9 (σ-Realizability).
Let σ be a semantics. The extension-set $\mathcal{C}$ is said to be *realizable with respect to* σ (or σ-realizable) if and only if there exists an argumentation framework F such that $Ext_\sigma(F) = \mathcal{C}$.

It is not always the case that an extension-set corresponds to the set of extensions of an argumentation framework with respect to a given semantics. This is illustrated by Example 5.

Example 5.
Given the set of arguments $A = \{a_1, a_2, a_3, a_4, a_5\}$, the extension-set $\mathcal{C} = \{\emptyset, \{a_2, a_4\}, \{a_1, a_3, a_5\}\}$ is realizable with respect to the complete semantics (for instance, it is the set of complete extensions of the argumentation framework given at Figure 1.6), but it is not realizable with respect to the preferred semantics (since the empty set is not maximal with respect to set inclusion).

Dunne and colleagues study the expressiveness of the different semantics. To this aim, they define the signature of a semantics.

[4]In our contributions, we call a set of arguments a *candidate*, meaning "candidate to be an extension", since it cannot be considered as an extension if it is not related to an argumentation framework and a semantics.

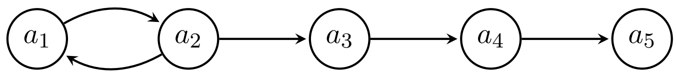


Figure 1.6: An Argumentation Framework Corresponding to $\mathcal{C}$

Definition 10 (σ-Signature)**.**
The *signature* of the semantics σ (or σ-signature) Σ_σ is the set of extension-sets $\{\mathcal{C}_1, \dots, \mathcal{C}_n\}$ which are σ-realizable.

Then, they identify exactly the properties of the σ-signature for the most usual semantics. First, let us introduce some properties of extension-sets.

Definition 11 (Downward-closure and Incomparability)**.**
Let $\mathcal{C}$ be an extension-set. The *downward-closure* of $\mathcal{C}$ is $dcl(\mathcal{C}) = \{c' \subseteq c \mid c \in \mathcal{C}\}$. $\mathcal{C}$ is called *downward-closed* if and only if $\mathcal{C} = dcl(\mathcal{C})$, and *incomparable* if and only if for each $c, c' \in \mathcal{C}$, $c \subseteq c'$ implies $c = c'$.

Definition 12 (Tightness)**.**
Given an extension-set $\mathcal{C}$, $Arg_\mathcal{C}$ denotes $\bigcup_{c \in \mathcal{C}} c$ while $Pairs_\mathcal{C}$ denotes the set $\{(a_i, a_j) \mid \exists c \in \mathcal{C}$ such that $\{a, b\} \in c\}$.
$\mathcal{C}$ is *tight* if and only if for each $c \in \mathcal{C}$ and each $a_i \in Arg_\mathcal{C}$, if $c \cup \{a_i\} \notin \mathcal{C}$ then there exists an $a_j \in c$ such that $(a_i, a_j) \notin Pairs_\mathcal{C}$.

Definition 13 (adm-Closeness)**.**
An extension-set $\mathcal{C}$ is called adm-closed if and only if for each $c_1, c_2 \in \mathcal{C}$, if $(a_i, a_j) \in Pairs_\mathcal{C}$ for each $a_i, a_j \in c_1 \cup c_2$, then $c_1 \cup c_2 \in \mathcal{C}$.

Now, we identify the signatures of the usual semantics.

Proposition 8 (σ-Signature of the Usual Semantics)**.**
The following collections of extension-sets are the signatures of the considered semantics.[5]

- $\Sigma_{cf} = \{\mathcal{C} \mid \mathcal{C}$ *is downward-closed and tight* $\}$
- $\Sigma_{stb} = \{\mathcal{C} \mid \mathcal{C}$ *is incomparable and tight* $\}$
- $\Sigma_{adm} = \{\mathcal{C} \mid \mathcal{C}$ *is* adm*-closed and contains* $\emptyset\}$
- $\Sigma_{pref} = \{\mathcal{C} \mid \mathcal{C}$ *is incomparable and* adm*-closed* $\}$

The complete-signature is not fully characterized, but it is known that $\Sigma_{adm} \subset \Sigma_{co}$.

We conclude the introduction to realizability with an interesting complexity result.

Proposition 9.
For the semantics $\sigma \in \{cf, stb, adm, pref\}$*, given a set of candidates* $\mathcal{C}$*, testing if* $\mathcal{C} \in \Sigma_\sigma$ *is a polynomial-time decision problem.*[6]

[5]Dunne and colleagues also identify the signatures of other semantics which are not presented in this thesis.
[6]Let us notice that it is also the case for the other semantics considered in [DDLW14], except the complete semantics.

1.3 Applications of Argumentation

In this last section, we introduce in an informal way several possible applications of abstract argumentation frameworks. These toy examples are related to problems where argumentation is useful, and we can use them to explain and motivate the intervention of the dynamics of argumentation frameworks.
First, argumentation can be used to determine, among a set of conficting options, which one must be chosen. A typical kind of scenario is the choice of an action to perform.
A second application of argumentation is a form of persuasion. When several agents have to take a common decision, but disagree about which option is the best, a rational way to find an agreement is to discuss, each agent setting in the debate some arguments to persuade the other agents that her option is the best one.
At last, we present a less intuitive application of abstract argumentation: resource allocation. We show that the allocations of some resources between several agents can be represented as some conflicts, similar to attacks in an argumentation scenario. Then, the different semantics can be used to choose the "good" allocation.

1.3.1 Argumentation and Decision Making

Let us suppose that an agent has to choose between several conflicting options an action (or a list of actions) to perform. The reasons to choose an option rather than another one can be modelled through arguments attacking each other, some of them representing directly a particular action. If one of these arguments is accepted, with respect to the semantics used by the agent, then the action associated with the argument is performed by the agent.

This application requires the agent to have a set of options along with the argumentation framework $F = \langle A, R \rangle$, each one representing an action to perform: $O = \{o_1, \dots, o_p\}$ and a set of rules r_{a_i,o_j} meaning intuitively "if argument a_i is accepted,[7] then option o_j has to be chosen". The actions can be mutually exclusive or not. When it is the case, it is required that the arguments supporting these options attack each other to ensure that two conflicting options are not chosen together. We may also have some tie-break rules if there are some *ex aequo* solutions (meaning several extensions for the chosen semantics), as we exemplify below.

Let us imagine the following scenario: John et Yoko want to go to the movie. They disagree about the film they want to watch. John wants to watch an action movie (option o_1) because it received some good reviews from the critics (argument a_1). Yoko prefers to watch a family comedy (option o_2) because she thinks that this movie is more interesting (argument a_2). Moreover, she recalls to John that they have to take care of their children, and the action movie is not well-suited to children (argument a_3). John answers that they can hire a baby-sitter for the evening (option o_3), and so they do not have to worry about their children (argument a_4). Yoko stays in a disagreement about the action movie, and tells John that one of her friends has already seen it, and did not like it (argument a_5). This scenario is represented by the argumentation framework and the options given on Figure 1.7.

The next step is to decide which semantics and kind of acceptance is used to organize the evening. With the grounded semantics, the result is not convincing for John and Yoko: the result is $\{\{a_4\}\}$, meaning that they have to hire the baby-sitter, but not to go to the cinema. It is obviously not a satisfying

[7] Here, the meaning of "accepted" is not explicitely specified, and depends of the agent (skeptically, credulously, with respect to any semantics,. . .).

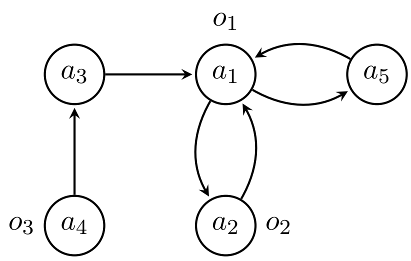


Figure 1.7: Graphical Representation of the Debate between John and Yoko

solution. With another semantics (for instance, the stable semantics), we get the following set of extensions: $\{\{a_1, a_4\}, \{a_2, a_4, a_5\}\}$. With skeptical reasoning, we obtain the same result as the grounded semantics. A credulous reasoning leads John and Yoko to watch both movies. This solution is not either a satisfying solution for them, except if they have enough free time to attend two screenings.
However, it is possible to refine the result, for instance if the agents agree on a tie-break rule, a single extension can be selected. A possible one, here, is to consider that option o_2 is stronger than option o_1, since the argument which supports o_1 is attacked by two arguments in the extension $\{a_2, a_4, a_5\}$, while the argument which supports o_2 is only attacked by a single argument when the extension $\{a_1, a_4\}$ is considered. This is a simple rule which can be defined for this particular application, but the agents can agree on any kind of tie-break rule, even the one which picks randomly one of the possible solutions. Of course, such a tie-break rule has to be decided by the agent before the study of the extension, to avoid a manipulation of the result by chosing a tie-break rule which is more in favor of one of the agents.

For more details about this kind of scenario, we refer the reader to the works presented in [ADM08, AP09, AV12].

Dynamics of Argumentation and Decision Making The dynamics of argumentation frameworks is useful in many ways in such a scenario. The simplest one is the fact that one of the agents can learn (or remember) a new argument or attack, and put it in the debate, to make the issue more favorable to its preferences. For instance, if John learns a new argument against the family comedy, then he will state this argument to prevent this movie to be chosen.
A more subtle scenario, in which we are interested, is to determine what happen if the agents learn a new information about the status of an argument. For instance, if John and Yoko learn from a trustworthy person that they should not watch the family comedy, they have to modify their argumentation framework to incorporate this information, but without adding a new argument to explain this change of status of an argument.

1.3.2 Argumentation and Goal-Oriented Persuasion

A second possible application of argumentation is the persuasion of an agent (or a group of agents) by another one. This case is interesting when the beliefs and the goals of the agents are not compatible with each other. In this situation, each agent can influence the outcome of the debate to lead the group to accept or reject some particular arguments. The way to influence the debate has to be optimized by each agent to guarantee that the result is the expected one. It is not the case in the scenario described in Section 1.3.1, where John and Yoko set all their arguments in the debate, then analyse the situation and decide what has to be chosen.

Let us switch to John and Paul, who disagree about the place where they will play their next concert. John wants to play in New-York City (a_1), while Paul prefers Chicago (a_2). Contrary to Paul, John knows that the sound is not as good in the New-York City concert hall than in Chicago (we can suppose that John has some other reason to prefer this city), it is the argument a_3. Our agents agree that a night in New-York City is much more expensive than a night in Chicago, which is not a good point for New-York City either (argument a_4), but Paul knows that it is possible to have a cheaper night in the hotel associated with the concert hall ($a5$). Figure 1.8 presents the set of arguments in this scenario. Black arguments and attacks are those known by both agents, while the blue ones are those which are known only by Paul, and the green ones are those that only John knows.

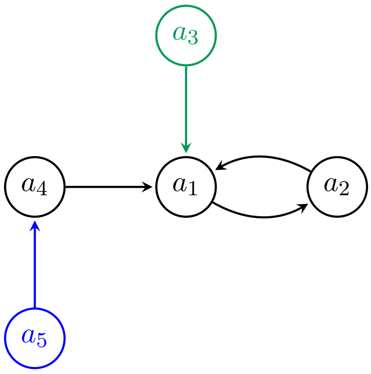


Figure 1.8: Graphical Representation of the Debate between John and Paul

We suppose that the debate starts with the arguments concerning John and Paul's preferences (a_1 et a_2 are set in the debate, presented on Figure 1.9.

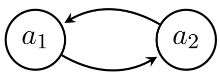


Figure 1.9: First Step of the Debate

It is not interesting for them to stop at this point, because there is no solution (grounded semantics gives the empty set as its extension, and the other usual semantics give the set $\{\{a_1\}, \{a_2\}\}$). As Paul prefers to go to Chicago it is rational for him to recall John the prices of the hotel rooms in New-York, so we obtain the graph presented on Figure 1.10. Now there is a possible solution: $\{a_2, a_4\}$ is the single

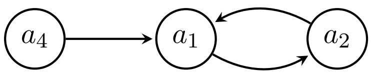


Figure 1.10: Second Step of the Debate

extension. Even if this situation is not satisfying for John, he does not want to add another argument, because the only one that he can put in the debate is a_3, which is not in favor of his preferred option. Similarly, Paul possesses an information that he does not want to share with John about the possible cheaper price of the hotel room: indeed, putting this information in the debate would take away Paul from his goal (play in Chicago) because $\{\{a_5\}\}$ would be the outcome of the debate with respect to the grounded semantics, and $\{\{a_1, a_5\}, \{a_2, a_5\}\}$ for the other ones.

So, it is important, when an agent wishes a particular outcome for the debate, to choose wisely the arguments that she puts in the debate, even if it means that she has to hide some information.
This kind of approach has been formalized, for instance by [BMM14].

Dynamics of Argumentation and Goal-Oriented Persuasion The application of dynamics of argumentation that we describe previously, in the section about John and Yoko choosing a movie, still apply here. We can also remark that since the goal of the agents is important here, one of them can try to manipulate the result of the debate. The simplest way for an agent to manipulate the debate, again, is the addition of new arguments to achieve her goal. But it is not always the case that additional arguments are available. So we also consider possible that the agent will try to convince her opponent that some of the attacks between the existing arguments are not correctly stated, and then should be added or removed.

1.3.3 Argumentation and Resources Allocation

In this example, Dung's framework is not used to deal directly with an argumentative scenario. We wish to model and solve a problem consisting in allocating some resources $\{r_1, \ldots, r_n\}$ to some users $\{u_1, \ldots, u_m\}$. (r_i, u_j) denotes the allocation of a resource r_i to an user u_j.
There may be conflicts between some allocations, for instance it is possible that a particular resource r_i cannot be shared between several users (then, (r_i, u_j) and (r_i, u_k) are conflicting for each $j \neq k$). In some cases, the users can be unable to use several resources that ones (then, $((r_i, u_j)$ and (r_k, u_j) are conflicting). Moreover, there may be a "preference" for a particular allocation on another one (for instance, a user can be more skilled to use a particular resource than another one).

We can see an intuitive way to solve our problem through argumentation frameworks: arguments correspond to the allocations, incompatibility between allocations are mutual attacks, while an incompatibility with a preference is a direct attack (the preferred allocation attacks the other one). If there is some forbidden allocation, then it is a self-attack.

We exemplify this modelling on a simple case: we have to share some toys (the resources) between several children (the users). There is a set of three toys $\{console, board\_game, figurine\}$ to be shared between three children $\{John, Paul, George\}$. The figurine cannot be shared, while the console and the board game can. The parents think that Paul is too young to understand the rules of the board game, so this allocation attacks itself. Last, the parents consider that it is better to give children a figurine than a console, because it helps them to develop their imagination, and they also prefer the board game because it develops their sociability and communication skills. So, each allocation of the console is attacked by the allocations of the board game and the allocations of the figurine.
This scenario is modeled by the argumentation framework described on Figure 1.11, where the names of the toys and children are replaced by their first letter.

Solving this problem gives the outcome $\{\{(g, J), (g, G), (f, P)\}\}$ for the stable semantics, meaning that Paul will play with the figurine while John and George will play with the board game.
Grounded semantics is much more cautious. On this scenario, it results in an empty extension, meaning that the children will not play at all!
We also remark that the agent can use the complete semantics, and then obtain the set of extensions $\{\emptyset, \{(g, J)\}, \{(g, J), (g, G)\}, \{(g, J), (f, G)\}, \{(g, G), (f, J)\}, \{(g, G)\}, \{(g, J), (g, G), (f, P)\}\}$, or the preferred semantics which gives the same result as the stable semantics.

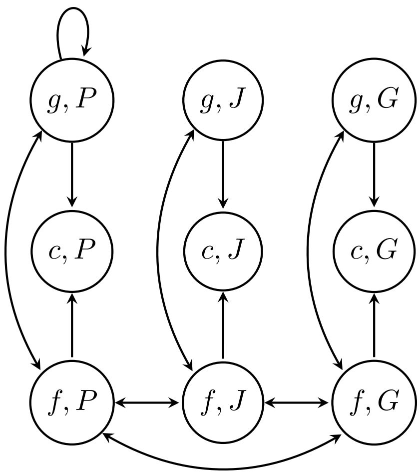


Figure 1.11: Model of the Toys Allocation

Dynamics of Argumentation and Resources Allocation This scenario can benefit from different approaches of the dynamics of argumentation frameworks. The integration of new users or new resources leads to the addition of arguments and attacks in the argumentation framework.
If an information about the outcome of the process is received, then changing the argumentation framework without adding arguments can be considered. For instance, if the parents accept to let Paul play the board game with the other children, then the self-attack on (g, P) must be removed. Similarly, if the parents learn that playing the console is a valuable outcome of the problem, then it means that the arguments about the video games should attack the other ones, leading to an extensions which contains the three of them.

1.4 Conclusion

In this chapter, we introduce the theory of abtract argumentation. From the notion of abstract argumentation framework proposed by Dung, it is possible to reason using different semantics to compute so-called extensions, which are the sets of jointly acceptable arguments. The notion of extension can be refined through the concept of labellings, which introduce a new status for the arguments, between acceptance and complete rejection.
We also describe two existing works which are useful to some of our contributions. First, we present how argumentation semantics can be encoded into propositional formulae; and then we give some results about the realizability of an extension-set.
Finally, toy examples of applications of arguments are given to sketch the way abstract argumentation can be useful for real problems, and how the dynamics of argumentation frameworks can take a part in these real problems. This topic is more formally presented in Chapter 3, and of course in the chapters dedicated to the contributions of this thesis.

Chapter 2

Belief Change

So the universe is not quite as you thought it was. You'd better rearrange your beliefs, then. Because you certainly can't rearrange the universe.

Isaac Asimov – *Nightfall*

Modelling an "intelligent" reasoning requires to study some more or less complex processes which can be performed on a belief base: checking if the beliefs are consistent (SAT problem), or if a given piece of belief is a consequence of the belief base (inference problem). These are static processes: the agent who is using the belief base only needs to interrogate the belief base. Other kinds of processes can be studied, linked to an evolution of the agent's beliefs. This chapter gives a brief description of some of these kinds of reasoning.
First, we present the AGM framework for belief change [AGM85]. In this work, the authors describe three different kinds of dynamic processes, and they give a set of rationality postulates that are expected to be satisfied by any belief change operator. This set of postulates can be associated with a family of operators, which are described in a constructive way.
Then, we present the work by Katsuno and Mendelzon about the adaptation of belief revision to the setting of finite propositional logic [KM91]. The same authors have also been interested in belief update [KM92]. We introduce this operation and explain the difference between revision and update.
A brief presentation of the Dynamic Logic of Propositional Assignments [HLMT11, BHT13], which is a possible way to encode belief change operations, finishes this chapter.

Contents

2.1 AGM Framework

We can represent the information known by an agent as a logical knowledge (or belief) base. It is very likely, in the life of an agent, that her beliefs change, for different reasons. For instance, if the agent learns that some of her beliefs are not consistent with the real state of the world, she changes it to ensure consistency with the world.
We present here one of the most influential work on belief change, the *AGM framework*, from its promoters Carlos Alchourrón, Peter Gärdenfors and David Makinson, which uses theories to represent the agent's beliefs.

2.1.1 Belief Status, Belief Change

When an agent's beliefs are represented through a logical belief base K, a formula α can have three different statuses for the agent:

- $K \vdash \alpha$: the agent can deduce α from her beliefs, so she considers that α is true; we say that α is *accepted*.
- $K \vdash \neg\alpha$: the agent can deduce the negation of α from her beliefs, so she considers that α is false; we say that α is *rejected*.
- $K \nvdash \alpha$ and $K \nvdash \neg\alpha$: the agent cannot deduce anything about α nor its negation; we say that α is *undetermined*.

In the AGM framework, the beliefs of the agent are represented through a theory, which is a deductively closed set of formulae: $K = \{\varphi \mid K \vdash \varphi\}$. So for this particular case, $K \vdash \alpha$ is equivalent to $\alpha \in K$.

During the agent's life, it is possible that her beliefs about the world change. For instance, some new pieces of information, considered to be more faithful than the agent's previous beliefs (because they are more recent, or because their source is reliable) contradict these previous beliefs.

Example 6.
Let us suppose that John believes that Paul owns a dog and a cat. His belief base is $K = \{dog, cat\}$. If Paul tells John "I do not think that it is possible to own a dog and a cat together, they would always fight with each other", then John has to change his beliefs, because his current belief base is not consistent with the new piece of information from Paul. Formally, it means that John has to incorporate the formula $\neg(dog \wedge cat)$ to his beliefs.
Moreover, if Paul talks about his goldfish, John can add $goldFish$ in his belief base, though there were no piece of information about a goldfish until then.

A belief status change can have a different kind of nature, depending on the statuses which are concerned by this change. For instance, moving a belief from undetermined to accepted or refused can be seen as an expansion (the fact to add some new beliefs to the base).
The opposite change can be seen as a contraction (the fact to forget a belief in the base). Last, moving a belief from accepted to refused (or vice-versa) is called a revision (the fact to add a new belief which is conflicting with the previous beliefs).

Some intuitive principles are expected to be satisfied by these operations:

- primacy of update: the new piece of information concerned by the belief change must have the expected status in the outcome;

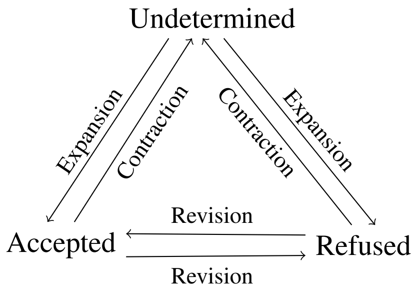


Figure 2.1: Transitions between Beliefs States

- consistency: the result must be consistent if the new piece of information is consistent; this property is not satisfied by expansion;
- minimal change: the agent must avoid to forget or add too much information.

In the rest of this section, we present the formal characterization of these three kinds of belief change in the AGM framework. First, each of them is associated with a set of *rationality postulates*, that are some logical properties which have to be satisfied by any "good" operator. Then, we present some ways to define change operators which satisfy the postulates.

2.1.2 Rationality Postulates and Links between Operations

Rationality Postulates

For each of the three operations briefly described previously, the AGM framework presents a set of rationality postulates, that are some properties which have to be satisfied by an operator to be considered as a "good" one.
For instance, if the operation is supposed to add $goldFish$ to the belief base, it does not seem rational to forget everything else and to keep only the new piece of information. It would obviously violate the minimal change principle.

Formally, an AGM expansion operator is a function which maps a theory K and a logical formula α to new theory $K + \alpha$ such that:

- **(K+1)** $K + \alpha$ is a theory
- **(K+2)** $\alpha \in K + \alpha$
- **(K+3)** $K \subseteq K + \alpha$
- **(K+4)** If $\alpha \in K$, then $K + \alpha = K$
- **(K+5)** If $K' \subseteq K$, then $K' + \alpha \subseteq K + \alpha$
- **(K+6)** $K + \alpha$ is the smaller theory satisfying **(K+1)** - **(K+5)**

Each postulate can be explained intuitively. The first one guarantees that the result of the operation is a theory, so the beliefs are represented in the same way after the expansion as before. **(K+2)** ensures that the new piece of information is actually accepted in the outcome theory. **(K+3)** orders not to

remove pieces of information from the theory, but only to add some new ones, which justifies the name "expansion". **(K+4)** expresses that it is not required to change the theory if it already contains the new piece of information. Postulate **(K+5)** asks the expansion to be monotonic. **(K+6)** expresses minimal change: no belief which is not justified by the addition of α can be added.
From these postulates, Gärdenfors deduces that there can be only one rational expansion operator.

Theorem 1 ([Gär88]).
*The expansion operator + satisfies postulates **(K+1)** - **(K+6)** if and only if* $K + \alpha = Cn(K \cup \{\alpha\})$.

Gärdenfors proved that the expansion is a very simple operation: making a conjunction with the former theory and the new piece of information proves enough. This operation does not satisfy the consistency principle.

Example 7.
When John learns that Paul does not want to own a dog and a cat at once, using the expansion operator with the new piece of information $\neg(cat \wedge dog)$ is not a good idea, since this new piece of information is not consistent with John's beliefs, and so the expanded theory is the trivial set $K_\perp$, which contains every possible formula from the language.
However, since John did not have any belief about the presence (or not) of a gold fish at Paul's house, it is not a problem to expand the belief base K by $goldFish$.

Another way to tackle the problem of John and Paul is to do a contraction: now the aim is to remove from the theory a piece of information which is supposed to be guaranteed to be correct. Since it was not correct to believe that Paul owns a dog and a cat together, a contraction by $dog \wedge cat$ leads to a new theory which does not imply this piece of information any longer.
A contraction operator $\div$ is a mapping from a theory and a formula to a new theory. It satisfies the following properties:

- **(K÷1)** $K \div \alpha$ is a theory
- **(K÷2)** $K \div \alpha \subseteq K$
- **(K÷3)** If $\alpha \notin K$, then $K \div \alpha = K$
- **(K÷4)** If $\nvdash \alpha$, then $\alpha \notin K \div \alpha$
- **(K÷5)** If $\alpha \in K$, then $K \subseteq (K \div \alpha) + \alpha$
- **(K÷6)** If$\vdash \alpha \leftrightarrow \beta$, then $K \div \alpha = K \div \beta$
- **(K÷7)** $(K \div \alpha) \cap (K \div \beta) \subseteq K \div (\alpha \wedge \beta)$
- **(K÷8)** If $\alpha \notin K \div (\alpha \wedge \beta)$, then $K \div (\alpha \wedge \beta) \subseteq K \div \alpha$

(K÷1) meaning is the same one as for **(K+1)**. **(K÷2)** guarantees that contraction does not add any new piece of information. **(K÷3)** prevents from performing some change to the theory if the target piece of information does not belong to the agent's belief. It is a counter-part of **(K+4)**. **(K÷4)** ensures the success of the operation: if the belief α is not a valid formula, then α cannot belong to the new theory. Postulate **(K÷5)** implies that contracting a theory K then expanding the result by the same belief α leads to the input theory K if α belongs to K. **(K÷6)** expresses the irrelevance of syntax. These six postulates are called basic postulates for contraction.

The last two postulates are the additional postulates. It is possible to define AGM contraction operators which only satisfy the six basic postulates. **(K÷7)** states that the beliefs which belong to the theory resulting from a contraction by α and to the theory resulting from a contraction by β also belong to the from obtained when contracting by the conjunction of α and β. **(K÷8)** expresses minimal change with respect to the conjunction; if $\alpha \wedge \beta$ has to be removed, it means that either α or β has to be removed. If α is actually removed, then there has been as much removal as if only α had just been the target of the contraction.

Now, the last operation of the AGM framework is revision. Similarly to expansion, the aim of revision is to add a new piece of information to the theory. The difference is that revision satisfies the consistency principle: if the new piece of information is inconsistent with the input theory, then such (minimal) changes must be performed to include the new piece of information without leading to the trivial theory.
An AGM revision operator $*$ is a mapping from a theory K and a formula α to a new theory $K * \alpha$ which satisfies the following postulates:

- **(K∗1)** $K * \alpha$ is a theory
- **(K∗2)** $\alpha \in K * \alpha$
- **(K∗3)** $K * \alpha \subseteq K + \alpha$
- **(K∗4)** If $\neg\alpha \notin K$, then $K + \alpha \subseteq K * \alpha$
- **(K∗5)** $K * \alpha = K_\perp$ if and only if $\vdash \neg\alpha$
- **(K∗6)** If $\vdash \alpha \leftrightarrow \beta$, then $K * \alpha = K * \beta$
- **(K∗7)** $K * (\alpha \wedge \beta) \subseteq (K * \alpha) + \beta$
- **(K∗8)** If $\neg\beta \notin K * \alpha$, then $(K * \alpha) + \beta \subseteq K * (\alpha \wedge \beta)$

As it was the case for the previous operations, **(K∗1)** ensures that the result of the revision is a theory. **(K∗2)** indicates that the new piece of information must be a conclusion of the revised belief base. **(K∗3)** means that the pieces of information in the outcome of revision are either some beliefs from the input belief base, or some consequences of the new belief. This same postulate, joined with postulate **(K∗4)**, says that if the new belief does not contradict the original belief base, then revision is equivalent to expansion. **(K∗5)** means that the result is expected to be consistent, unless the new piece of information is inconsistent. **(K∗6)** expresses the irrelevance of the syntax during the revision process.
These are the six basic postulates. The two additional postulates guarantee a good behavior of the revision with respect to conjunctions. **(K∗7)** indicates that revising the theory K by the conjunction of two pieces of information gives a result which is included in the theory revised by the first piece of informaiton, and expanded by the second one. **(K∗8)** asks the opposite inclusion to be true when expansion does not lead to an inconsistent theory.

Example 8.
To incorporate the piece of information that he deduces from his discussion with Paul to his beliefs, John can revise his belief base by $\neg(cat \wedge dog)$. Since the revision operator has to maintain consistency, the problem existing with the expansion does not occur here.

Links between Operations

It has been proven that some links exist between revision and contraction. Indeed, given an AGM contraction operator, it is possible to define an AGM revision operator, and vice-versa:

Levi's identity $K * \alpha = (K \div \neg\alpha) + \alpha$

Harper's identity $K \div \alpha = K \cap (K * \neg\alpha)$

Intuitively, Levi's identity means that we can revise the theory K by α with a contraction step to remove everything which concerns $\neg\alpha$, and an expansion step to add α.
In the opposite, Harper's identity explains how to contract K by α through a revision. It is done by keeping the intersection of K itself with the revision of K by $\neg\alpha$. In this way, we keep the beliefs from the initial theory which are not related to α or $\neg\alpha$.

Gärdenfors [Gär88] proved that a revision operator defined via Levi's identity and an AGM contraction operator is an AGM revision operator, and the converse also holds: a contraction operator defined though Harper's identity and an AGM revision operator satisfies AGM postulates.

Theorem 2 ([Gär88]).
*If the contraction operator $\div$ satisfies **(K÷1)** - **(K÷4)** and **(K÷6)**, and if the expansion operator $+$ satisfies **(K+1)** - **(K+6)**, then the operator $*$ defined though Levi's identity satisfies **(K∗1)** - **(K∗6)**.*
*Moreover, if $\div$ satisfies **(K÷7)** (respectively **(K÷8)**) then $*$ satisfies **(K∗7)** (respectively **(K∗8)**).*

Theorem 3 ([Gär88]).
*If the revision operator $*$ satisfies **(K∗1)** - **(K∗6)**, then the operator $\div$ defined through Harper's identity satisfies **(K÷1)** -**(K÷6)**.*
*Moreover, if $*$ satisfies **(K∗7)** (respectively **(K∗8)**) then $\div$ satisfies **(K÷7)** (respectively. **(K÷8)**).*

2.1.3 Representation Theorems

A representation theorem associates a set of rationality postulates (which are an axiomatic characterization of change operators) with a concrete family of operators (which gives a constructive characterization). It is possible to propose different representation theorems for the same family of operators. Moreover, since contraction and revision are linked by Levi's identity and Harper's identity, a representation theorem for one of these operations also defines a family of operators for the other operation. In this section, we only present two representation theorems: partial meet contraction and revision by systems of spheres.

Partial Meet Contraction

Partial meet contraction [AGM85] keeps each maximal subset of the theory which does not imply the piece of information expected to be removed. A skeptical policy is applied to these subsets, and so the agent keeps their intersection.

Definition 14 ($K \perp \alpha$).
Let K be a theory and α a formula. The set of maximal subsets of K not implying α, noted $K \perp \alpha$, is the set of every K' such that:

- $K' \subseteq K$
- $K' \nvdash \alpha$

- $\forall K''$ such that $K' \subset K'' \subseteq K, K'' \vdash \alpha$

Definition 15 (Full Meet Contraction)**.**
A full meet contraction function is defined by

$$K \div_f \alpha = \begin{cases} \cap(K \bot \alpha) & \text{if } K \bot \alpha \neq \emptyset \\ K & \text{in the other case} \end{cases}$$

The full meet contraction operator presents a weakness: if we define a revision operator with Levi's identity, and the full meet contraction as the underlying contraction operator, then the result is the set of all the formulae in K which are consequences of $\neg\alpha$.

Theorem 4 ([AM82])**.**
A revision operator $*$ defined from a full meet contraction function and Levi's identity is equal to $K * \alpha = Cn(\alpha)$ for each α such that $\neg\alpha \in K$.

Concretely, this theorem means that a revision operator buildt from full meet contraction forgets all the former beliefs, and only keeps the new piece of information and its consequences. Even if it satisfies the postulates, this behavior is not interesting for a real application. The reason of this problem is that full meet contraction removes too much information. Before processing the intersection of the maximal subsets, some of them must be removed, keeping only the "best" ones.

Definition 16 (Selection Function)**.**
We call selection function γ a mapping from each formula α and each theory K to the set $\gamma(K \bot \alpha)$, which is a non-empty subset of $K \bot \alpha$ if $K \bot \alpha \neq \emptyset$, and $\{K\}$ otherwise.

Definition 17 (Partial Meet Contraction)**.**
A partial meet contraction function is defined by

$$K \div \alpha = \cap\gamma(K \bot \alpha)$$

The behavior of partial meet contraction is similar to the behavior of full meet contraction, but now only the subsets selected by γ are taken into account by the intersection. In fact, full meet contraction is a particular case of partial meet contraction, with γ selected each $K' \in K \bot \alpha$.

Now, let us present the representation theorem from Alchourrón, Gärdenfors and Makinson about partial meet contraction functions.

Theorem 5 ([AGM85])**.**
*An operator $\div$ is a partial meet contraction function if and only if it satisfies **(K÷1)** - **(K÷6)**.*

It is possible to add some contraints on γ to ensure that the contraction operator satisfies the postulates **(K÷7)** and **(K÷8)**.

Definition 18 (Relational Selection Function)**.**
A selection function γ is relational if and only for each belief base K, it is possible to define a relation $\leq$ over $K \times K$ such that

$$\gamma(K \bot \alpha) = \{K' \in K \bot \alpha | K' \leq K'', \forall K'' \in K \bot \alpha\}$$

Moreover, if $\leq$ is transitive, then γ and the associated partial meet contraction function are called transitively relational.

There is also a representation theorem for these functions:

Theorem 6 ([AGM85])**.**
*An operator $\div$ is a transitively relational partial meet contraction if and only if it satisfies **(K÷1)** - **(K÷8)**.*

Revision by Systems of Spheres

Now, let us present an approach for revising a belief base. The main idea of the systems of spheres [Gro88] is to keep the possible worlds that are compatible with the input of the revision, which are the most plausible ones for the agent.

A possible world of a theory is a subset of the language which is maximal among the consistent subsets, and such that the formulae from the theory are $true$ in this subset. Each possible world is a way to describe completely the world which is consistent with the agent's beliefs.

Definition 19 (Possible Worlds of a Theory)**.**
A possible world is a maximal consistent subset of the language $\mathcal{L}$; $M_{\mathcal{L}}$ denotes the set of all possible worlds from $\mathcal{L}$.

- Let K be a theory.

$$[K] = \begin{cases} \emptyset & \text{if} \quad K = K_{\perp} \\ \{M \in M_{\mathcal{L}} | K \subseteq M\} & \text{otherwise} \end{cases}$$

- Let $S \subseteq M_{\mathcal{L}}$ be a set of possible worlds, we define $K_S = \cap\{M | M \in S\}$

Definition 20 (Systems of Spheres)**.**
A system of spheres centered on $[K]$ is a set S of subsets of $M_{\mathcal{L}}$ such that

- **(S1)** If $s, s' \in S$ then $s \subseteq s'$ or $s' \subseteq s$
- **(S2)** $[K] \in S$
- **(S3)** $\forall s \in S, [K] \subseteq s$
- **(S4)** $M_{\mathcal{L}} \in S$
- **(S5)** If α is a formula and $[\alpha]$ intersects a sphere of S, there there is a minimal sphere which intersects $[\alpha]$ (noted $C(\alpha) = [\alpha] \cap S_{\alpha}$)

A sphere is a set of possible worlds, and the system of spheres centered on $[K]$ is built in the following way.

- The spheres are included each one into the others.
- The set of possible worlds of K is the smallest sphere.
- The set of all possible worlds $M_{\mathcal{L}}$ is the largest sphere.

The intuitive meaning of the system of spheres is that a possible world w is more plausible than another one w' if w is contained into a smaller sphere than w'.

Theorem 7 ([Gro88])**.**
Given a theory K, there exists a system of spheres S centered on $[K]$ such that for each formula α, $K * \alpha = K_{C(\alpha)}$ if and only if $*$ is a revision operator which satisfies ***(K*1) - (K*8)****.*

Revising a theory by a system of spheres can be represented graphically. Figure 2.2 presents an example of system of spheres centered on $[K]$: the worlds which are consistent with the agent's beliefs are the most plausible ones. Then, the possible worlds are ordered thanks to the inclusion of spheres: worlds from sphere S_1 are less plausible than the worlds from $[K]$, but more plausible than worlds from sphere S_2.
To revise K by a piece of information α, the agent keeps the possible worlds consistent with α which belong to the "lowest" sphere, meaning the most plausible ones.

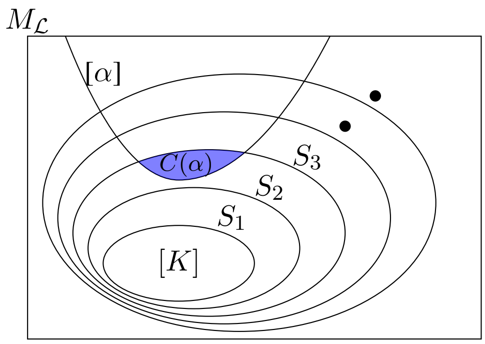


Figure 2.2: Revision of K by α through a System of Spheres centered on $[K]$

2.2 Belief Change in Propositional Logic

Finite propositional logic is a well-known setting for reasoning. Its expressiveness along with its simplicity make it a very convenient way to represent an agent's beliefs. Rather than using a deductively closed belief set, we use now a single propositional formula to represent the agent's beliefs. In this section, we present the adaptation of the AGM framework in the setting of propositional logic. Katsuno and Mendelzon [KM91] have shown that it is possible to rewrite the AGM rationality postulates in this simple setting, and to give a representation theorem for these new postulates. At last, we introduce the update operation. Even if there is some intuitive connection between revision and update, Katsuno and Mendelzon [KM92] have shown that there is a fundamental difference between these two operations.

2.2.1 From Theories to Propositional Formulae

The first step, before introducing belief change in propositional logic, is to express the link between a theory K and a propositional formula φ_K which represents the same beliefs.

Definition 21.
Given a theory K, we say that φ_K is a propositional formula corresponding to K if and only if $K = Cn(\{\varphi_K\})$.
Given a propositional formula φ, we say that K_φ is the theory corresponding to φ if and only if $K_\varphi = Cn(\{\varphi\})$.

There may be different formulae $\varphi_K^1, \varphi_K^2, \ldots, \varphi_K^n$ associated with the same theory K, as soon as these formulae admit the same set of models.

Example 9.
Let K be the theory

$$Cn(\{x \vee (y \wedge z), \neg x \vee \neg y \vee \neg z\})$$

which admits the models

$$\{y, z\}, \{x\}, \{x, z\}, \{x, y\}.$$

It can be associated with the formula $\varphi_K^1 = [x \vee (y \wedge z)] \wedge (\neg x \vee \neg y \vee \neg z)$ which has the same set of models as K. Another equivalent formula is $\varphi_K^2 = \neg x \Leftrightarrow (y \wedge z)$.

So, given an AGM revision operator $*$ (respectively contraction operator $\div$) on theories, we define the AGM revision operator $\circ$ (respectively contraction operator $-$) on propositional formulae as the operator such that, for any propositional formulae φ and α, $\varphi \circ \alpha = \varphi_{K_\varphi * \alpha}$ (respectively $\varphi - \alpha = \varphi_{K_\varphi \div \alpha}$). In the opposite way, we define $*$ (respectively $\div$) from $\circ$ (respectively $-$) as the operator such that, for any theory K and any formula α, $K * \alpha = K_{\varphi_K \circ \alpha}$ (respectively $K \div \alpha = K_{\varphi_K - \alpha}$).

Definition 22.
Let $\bullet \in \{*, +, \div\}$ be an AGM operator. The corresponding propositional operator $\triangleright$ is the operator such that, for any propositional formulae φ, α, $Cn(\varphi \triangleright \alpha) = Cn(\varphi) \bullet \alpha$.

As explained by Theorem 1, there is a single operator which satisfies the postulates for the expansion: for each belief base K and each formula α, $K + \alpha = Cn(K \cup \{\alpha\})$. In the setting of propositional logic, this operator is only a conjunction. So, we do not present in detail the definition of the expansion operator in the rest of this section.
We present now how to define an AGM revision or contraction operator from the propositional counterpart.

Definition 23.
Let $\triangleright \in \{\circ, -\}$ be a propositional operator. The corresponding AGM operator $\bullet$ is the operator such that, for any theory K and any formula α, $K \bullet \alpha = Cn(\varphi \triangleright \alpha)$.

2.2.2 Belief Revision in Propositional Logic

The work from Katsuno and Mendelzon [KM91] adapts AGM revision in the setting of finite propositional logic. First of all, they reformulate the rationality postulates in a simpler way, well-suited to this setting.

A belief revision operator is a mapping from two propositional formulae φ and α to a new formula $\varphi \circ \alpha$ such that:

(R1) $\varphi \circ \alpha \vdash \alpha$

(R2) If $\varphi \wedge \alpha$ is consistent, then $\varphi \circ \alpha \equiv \varphi \wedge \alpha$

(R3) If α is consistent then $\varphi \circ \alpha$ is consistent

(R4) If $\varphi \equiv \psi$ and $\alpha \equiv \beta$ then $\varphi \circ \alpha \equiv \psi \circ \beta$

(R5) $(\varphi \circ \alpha) \wedge \psi \vdash \varphi \circ (\alpha \wedge \psi)$

(R6) If $(\varphi \circ \alpha) \wedge \psi$ is consistent then $\varphi \circ (\alpha \wedge \psi) \vdash (\varphi \circ \alpha) \wedge \psi$

The first postulate expresses the success principle: the new piece of information must be believed by the agent after revision. Postulate **(R2)** says that revision is just conjunction if this does not lead to an inconsistent formula. **(R3)** expresses the consistency principle. **(R4)** says that the revision is independent to the syntax of formulae. The two last postulates are the counterparts of AGM's **(K$*$7)** and **(K$*$8)**, they describe the behaviour of revision operators with respect to conjunctions.

In the rest of the thesis, we call these postulates the KM postulates, and any revision operator which satisfies them is called a KM revision operator.

Katsuno and Mendelzon proved the existence of a mapping between these postulates and AGM ones:

Theorem 8 ([KM91])**.**
Let $\star$ be a revision operator on theories, and $\circ$ a revision operator on propositional formulae corresponding to $\star$.
*$\star$ satisfies postulates **(K∗1)-(K∗8)** if and only if $\circ$ satisfies **(R1)-(R6)**.*

Katsuno and Mendelzon have presented an adaption of Grove's systems of spheres [Gro88], suited to propositional logic. The underlying idea is the same: we can order the interpretations with respect to their plausibility for the agent, and choose the models of the new piece of information which are the best ones with respect to this ranking. The ranking between interpretations has to satisfy some properties:

Definition 24 (Faithful Assignment)**.**
A faithful assignment is a mapping from each propositional formula φ to a total pre-order $\leq_\varphi$ such that:

- if $\omega \models \varphi$ and $\omega' \models \varphi$, then $\omega \simeq \omega'$;
- if $\omega \models \varphi$ and $\omega' \not\models \varphi$, then $\omega < \omega'$;
- if $\varphi_1 \equiv \varphi_2$, then $\leq_{\varphi_1} = \leq_{\varphi_2}$.

As soon as we have such a pre-order $\leq_\varphi$ associated with the formula φ which represents the agent's beliefs, revising by α consists in choosing the models of α which have the best rank with respect to $\leq_\varphi$.

Theorem 9 (Representation Theorem [KM91])**.**
*A revision operator $\circ$ satisfies the postulates **(R1)-(R6)** if and only if there exists a faithful assignment which maps each formula φ to a total pre-order $\leq_\varphi$ such that, for each formula α:*

$$\text{Mod}(\varphi \circ \alpha) = \min(\text{Mod}(\alpha), \leq_\varphi)$$

Example 10.
Figure 2.3 gives an example of a pre-order associated with a formula φ: dots at level L_0 represent the models of φ. The other ones represent the other interpretations, which are less and less plausible while their level grows. The shaded area represents the models of the formula α. The red points are the minimal models of α with respect to $\leq_\varphi$, so they are the models of $\varphi \circ \alpha$.

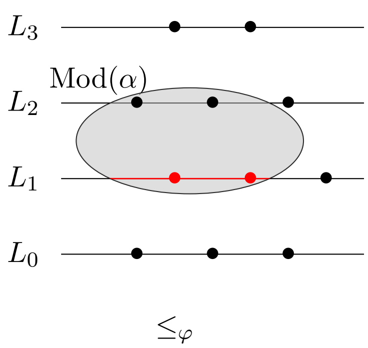


Figure 2.3: Minimal Models of α with respect to $\leq_\varphi$

Now, let us present a particular family of revision operators which satisfy the KM postulates, and one of the most well-known operators from this family. These are the distance-based revision operators, which use a distance between interpretations to define the faithful assignment, and Dalal's operator [Dal88], which can be expressed through such a distance. Defined several years before the publication of Katsuno and Mendelzon's work, Dalal's operator satisfies the KM postulate. Even if it was not defined through the notion of faithful assignment, we reformulate it in this way.

Definition 25 (Distance-based Revision Operator)**.**
Let d be a distance between interpretations over a set of Boolean variables V. Given a formula ψ in the propositional languange $\mathcal{L}$ built up on V, the pre-order $\leq_\psi$ is defined by

$$\omega \leq_\psi \omega' \text{ if and only if } d(\omega, \mathrm{Mod}(\psi)) \leq d(\omega', \mathrm{Mod}(\psi))$$

For any formulae $\psi, \alpha \in \mathcal{L}$, the distance-based KM revision operator $\circ_d$ is defined by

$$\mathrm{Mod}(\psi \circ_d \alpha) = \min(\mathrm{Mod}(\alpha), \leq_\psi)$$

Now, we need to recall the notion of Hamming distance [Ham50] which is used to define Dalal's operator.

Definition 26.
Let ω, ω' be two interpretations over a set of Boolean variables V. The Hamming distance between these interpretations is defined by

$$d_H(\omega, \omega') = |(\omega \backslash \omega') \cup (\omega' \backslash \omega)|$$

This distance counts the number of elements which appear in the interpretation ω but not in ω', and vice-versa. Said in another way, this distance counts the number of variables which are valued to $true$ for one of the interpretation and to $false$ for the other one.
This distance can be extended to define a measure of the difference between an interpretation ω and a set of interpretations Ω. This measure is not formally a distance, since it does not satisfy properties of distances[8].

$$d_H(\omega, \Omega) = \min_{\omega' \in \Omega}(d_h(\omega, \omega'))$$

Now, we have all the prerequisite notions to define Dalal's operator:

Definition 27 (Dalal Revision Operator)**.**
Dalal's revision operator $\circ_D$ is the distance-based revision operator defined from the Hamming distance.

It is easy to prove that the pre-order associated with the Hamming distance satisfies the properties of faithful assignments, and so, that $\circ_D$ is a KM revision operator[9].

2.2.3 Belief Update: Another Way to Incorporate a New Piece of Information

Belief update is an operation aiming at changing the status of a belief in a way close to revision. It aims at changing the agent's beliefs to lead to acceptance of a previously refused piece of information (or the converse). The subtle difference is the reason which leads to change the agent's beliefs. The operations

[8] It is not even used to measure the difference between similar objects, while a distance on a set E is a mapping from two elements in E to a real number.

[9] More generally, every distance-based pre-order satisfies the properties of faithful assignments, this is why distance-based revision operators satisfy the KM postulates.

described in the previous sections (expansion, contraction and revision) consider a change of the agent's beliefs about a static world: the agent changes her beliefs because she learnt that it was wrong to belief something. With updates scenario, it is supposed that the new piece of information concerns a change of the world. Whether the previous beliefs were true or not does not matter, it is required to change the agent's beliefs to be consistent with the new state of the world.

Example 11.
Let us come back to the case John's beliefs about Paul's pets. John believes $cat \wedge dog$.
If John hears Paul saying that he does not want to own a dog and a cat, fearing that they would not like each other, then John has to revise his beliefs to be consistent with the new piece of information $\neg(cat \wedge dog)$. In this scenario, the world does not evolve, this is why John should use a revision operator. Now, if Paul tells John that he moves to a new home where it is forbidden to own more than one pet, then John knows that Paul cannot own a dog and a cat together. It seems to be the same piece of information $\neg(cat \wedge dog)$ which is learnt by John. However, this piece of information comes from a change of the world, and so the operation is not the same one: an update operator should be used.

It may seem that there is no difference between both operations, since the new piece of information explaining the change is the same one. Katsuno and Mendelzon explain the difference between them in [KM92].

Let us first recall that revising a formula φ by a formula α leads to keep the subset of the models of α which are the most plausible ones with respect to the previous beliefs. This method cannot be used for update operation, because it could not give the same consideration to each model of the former beliefs φ. Let us explain it on Katsuno and Mendelzon's example.

Example 12 (Borrowed from [KM92])**.**
We know that five books are in a room, either on a table or on a bookcase. The Boolean variable x_i ($i \in \{1, 2, 3, 4, 5\}$) means "The book i is on the table". So $\neg x_i$ means that the book i is on the bookcase. We know that either the book 1 is the single one on the table, or the books 3, 4 and 5 are. It is expressed by the formula $\varphi = (x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg x_4 \wedge \neg x_5) \vee (\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4 \wedge x_5)$, which has two models: $\omega_1 = \{x_1\}$ et $\omega_2 = \{x_3, x_4, x_5\}$.
We send a robot to clear the room, with the mission to make as few effort as possible to put all the books at the same place. Two models are possible after than the robot has finished his mission: the one corresponding to all books on the table ($\omega'_1 = \{x_1, x_2, x_3, x_4, x_5\}$) and the one corresponding to all books on the bookcase ($\omega'_2 = \emptyset$). A revision operator would select among ω'_1 and ω'_2 the models which are the closest to φ taken as a whole. For instance, if we rank the interpretations with respect to Dalal's distance, which is given at Table 2.1, then the single model after the revision is ω'_2, since its distance to

	ω_1	ω_2
ω'_1	4	2
ω'_2	1	3

Table 2.1: Hamming Distance between the Models of φ and the Possible Models after the Change

the models of φ is 1. This operation suggests that the real world (after the change) is the one where each book is on the bookcase, and so only the book 1 was on the table in the previous state. But, since there were no information to decide which model among ω_1 and ω_2 was the real world before the mission of the robot, it is not reasonable not to consider any possible world which "succeeds" to ω_2.
If the state of the world before the mission of the robot was represented by ω_2, then the minimal change for the robot to perform its task leads to consider ω'_1 as a possible world after the update.

Now let us explain how Katsuno et Mendelzon modelled this process. We first introduce the notion of *complete* formula, and then we give an axiomatic characterization of update operation.

Definition 28.
Let φ be a propositional formula. φ is complete if and only if, for each propositional formula α, $\varphi \vdash \alpha$ or $\varphi \vdash \neg\alpha$.

Now we present the KM postulates for update. An update operator is a mapping from two propositional formulae φ and α to a new formula $\varphi \diamond \alpha$ such that:

(U1) $\varphi \diamond \alpha \vdash \alpha$

(U2) If $\varphi \vdash \alpha$, then $\varphi \diamond \alpha \equiv \varphi$

(U3) If φ and α are consistent, then $\varphi \diamond \alpha$ is consistent

(U4) If $\varphi_1 \equiv \varphi_2$ and $\alpha_1 \equiv \alpha_2$ then $\varphi_1 \diamond \alpha_1 \equiv \varphi_2 \diamond \alpha_2$

(U5) $(\varphi \diamond \alpha) \wedge \beta \vdash \varphi \diamond (\alpha \wedge \beta)$

(U6) If $(\varphi \diamond \alpha) \vdash \beta$ and $(\varphi \diamond \beta) \vdash \alpha$ then $(\varphi \diamond \alpha) \equiv (\varphi \diamond \beta)$

(U7) If φ is complete, then $(\varphi \diamond \alpha) \wedge (\varphi \diamond \beta) \vdash \varphi \diamond (\alpha \vee \beta)$

(U8) $(\varphi_1 \vee \varphi_2) \diamond \alpha \equiv (\varphi_1 \diamond \alpha) \vee (\varphi_2 \diamond \alpha)$

Postulates **(U1)**-**(U5)** are the immediate counterparts of the five first revision postulates. We still remark the difference between **(R2)** and **(U2)**. When updating an inconsistent formula, the result is inconsistent even if α is consistent (because the formula φ always satisfies $\varphi \vdash \alpha$, whatever α), while the revision is consistent as soon as α is consistent.
Revision postulate **(R6)** does not have a counterpart, but is replaced by three postulates for update. **(U6)** means that when the update of φ by α implies β, and vice-versa, then both updates are identical. **(U7)** concerns complete formulae. It says that for such a formula φ, if a model is common to the update of φ by α and the update of φ by β, then it must be a model of the update of φ by the disjunction of α and β. **(U8)** expresses the principle of distinct update of each model, as we illustrated in Example 12: each model of the formula φ (so each possible world with respect to the agent's former beliefs) is updated independently from the other ones.

Similarly to revision operators, the update operators can be characterised thanks to some particular pre-order between interpretations. Instead of associating each formula with a ranking between interpretation, Katsuno and Mendelzon consider for update some particular pre-order associated with interpretations.

Definition 29.
A faithful assignment is a mapping from each interpretation ω to a partial pre-order $\leq_\omega$ such that for each $\omega' \neq \omega$, $\omega <_\omega \omega'$.

Theorem 10 (Representation Theorem)**.**
*The update operator $\diamond$ satisfies **(U1)**-**(U8)** if and only if there exists a faithful assignment that maps each interpretation ω to a partial pre-order $\leq_\omega$ such that*

$$\mathrm{Mod}(\varphi \diamond \alpha) = \bigcup_{\omega \in \mathrm{Mod}(\varphi)} \min(\mathrm{Mod}(\alpha), \leq_\omega)$$

For instance Forbus' operator for updating propositional formulae [For89] is the update operator defined thanks to the faithful assignment based on the Hamming distance:

Definition 30 (Forbus' Update Operator)**.**
The update operator $\diamond_F$ is defined by

$$\mathrm{Mod}(\varphi \diamond_F \alpha) = \bigcup_{\omega \in \mathrm{Mod}(\varphi)} \min(\mathrm{Mod}(\alpha), \leq_\omega)$$

with $\leq_\omega$ defined, for each interpretation ω, by

$$\forall \omega', \omega'', \omega' \leq_\omega \omega'' \text{ if and only if } d_H(\omega, \omega') \leq d_H(\omega, \omega'')$$

2.2.4 Dynamic Logic of Propositional Assignments and Belief Change

The Dynamic Logic of Propositional Assignments (DL-PA) is a logic which makes operations on the valuations over a propositional language. DL-PA allows to write programs which vary the truth values of Boolean variables. Such programs can be used to model belief revision and belief update operators.

Syntax and Semantics of DL-PA The language DL-PA is defined by the grammar:

$$\begin{array}{lll} \pi & ::= & x \leftarrow \top \mid x \leftarrow \bot \mid \varphi? \mid \pi;\pi \mid \pi \cup \pi \mid \pi^- \\ \varphi & ::= & x \mid \top \mid \bot \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle \pi \rangle \varphi \end{array}$$

The atomic formulae x, $\top$, $\bot$, and the negation and disjunction connectives are used in the classical way. They lead to the definition of conjunction ($\wedge$), implication ($\Rightarrow$), equivalence ($\Leftrightarrow$) and exclusive or ($\oplus$).

The atomic programs $x \leftarrow \top$ and $x \leftarrow \bot$ change the value of the variable x to respectively $true$ and $false$. The test $\varphi?$ checks if the formula φ is $true$, the sequential composition $\pi;\pi$ applies two programs one after the other, and the non-deterministic composition $\pi \cup \pi$ applies one of the programs.

The most interesting part is the new formula $\langle \pi \rangle \varphi$, which means "after some execution of π, φ is $true$". It can be used to define $[\pi]\varphi = \neg\langle \pi \rangle \neg\varphi$, which means "after each execution of π, φ is $true$". Finally, π^- is the converse operator. Its meaning is that the formula $[\pi^-]\varphi$ stands for "before each execution of π, φ was $true$", and $\langle \pi^- \rangle \varphi$ stands for "before at least one execution of π, φ was $true$".

Several useful programs can be defined from the definition of the language. First, `skip` abbreviate the program $\top?$ and means "nothing happens". Then, for each $n \geq 0$, the programs π^n and $\pi^{\leq n}$ are defined inductively by

$$\pi^n = \begin{cases} \texttt{skip} & \text{if } n = 0 \\ \pi;\pi^{n-1} & \text{if } n > 0 \end{cases}$$

and

$$\pi^{\leq n} = \begin{cases} \texttt{skip} & \text{if } n = 0 \\ (\texttt{skip} \cup \pi);\pi^{n-1} & \text{if } n > 0 \end{cases}$$

which means respectively that π must be repeated n times, or at most n times. The following programs are used to assign to a variable the truth value of another one :

$$\begin{array}{rcl} x \leftarrow y & = & (y?; x \leftarrow \top) \cup (\neg y?; x \leftarrow \bot) \\ x \leftarrow \neg y & = & (y?; x \leftarrow \bot) \cup (\neg y?; x \leftarrow \top) \end{array}$$

Now, let us present the semantics of DL−PA. The models of formulae from DL−PA are simply models in the meaning of classical propositional logic, that is a valuation of each variable such that the truth value of the formula is $true$. The semantics of the connectives is defined as usually.
The interpretations of programs are pairs of propositional interpretations $p = (\omega_1, \omega_2)$, such that p satisfies π if and only if, when the current value of Boolean variables is determined by ω_1, then the execution of π leads to ω_2. The semantics of the different program connectives is described at Table 2.2.

$\mathrm{Mod}(x \leftarrow \top)$	$=$	$\{(\omega_1, \omega_2) \mid \omega_2 = \omega_1 \cup \{x\}\}$
$\mathrm{Mod}(x \leftarrow \bot)$	$=$	$\{(\omega_1, \omega_2) \mid \omega_2 = \omega_1 \backslash \{x\}\}$
$\mathrm{Mod}(\varphi?)$	$=$	$\{(\omega, \omega) \mid \omega \models \varphi\}$
$\mathrm{Mod}(\pi_1; \pi_2)$	$=$	$\{(\omega_1, \omega_3) \mid \exists \omega_2$ such that $(\omega_1, \omega_2) \in \mathrm{Mod}(\pi_1)$ and $(\omega_2, \omega_3) \in \mathrm{Mod}(\pi_2)\}$
$\mathrm{Mod}(\pi_1 \cup \pi_2)$	$=$	$\mathrm{Mod}(\pi_1) \cup \mathrm{Mod}(\pi_2)$
$\mathrm{Mod}(\pi^-)$	$=$	$\{(v_2, v_1) \mid (v_1; v_2) \in \mathrm{Mod}(\pi)\}$

Table 2.2: Semantics of Program Connectives in DL−PA

Two programs π_1 and π_2 are said to be equivalent if and only if $\mathrm{Mod}(\pi_1) = \mathrm{Mod}(\pi_2)$, which is noted $\pi_1 \equiv \pi_2$.

Belief Change Through DL−PA programs In [Her14], Andreas Herzig explains how DL−PA can be used to model belief change operators. First, let us present some useful DL−PA programs. For each program described at Table 2.3, it is supposed that $V = \{x_1, \ldots, x_n\}$. For $n = 0$, these programs are equivalent to skip. For each of these programs, $\pi(\{x\})$ is written $\pi(x)$.

$\mathtt{flip}(V)$	$=$	$x_1 \leftarrow \neg x_1 \cup \cdots \cup x_n \leftarrow \neg x_n$
$\mathtt{flip}^{\geq 0}(V)$	$=$	$(x_1 \leftarrow \top \cup x_1 \leftarrow \bot); \ldots; (x_n \leftarrow \top \cup x_n \leftarrow \bot)$

Table 2.3: Some Useful DL−PA Programs

Similarly, we need to present some useful DL−PA formulae, given at Table 2.4. V_φ denotes the set of Boolean variables which appear in φ, V' is a strict subset of V_φ, and m is an integer such that $m \leq |V_\varphi|$.

$\mathtt{Sat}(\varphi)$	$=$	$\langle \mathtt{flip}^{\geq 0}(V_\varphi) \rangle \varphi$	
$H(\varphi, \geq m)$	$=$	$\top$	if $m = 0$
		$\neg \langle \mathtt{flip}(V_\varphi) \rangle^{\geq m-1}$	if $m > 0$
$H(\varphi, V', \geq m)$	$=$	$\top$	if $m = 0$
		$\neg \langle \mathtt{flip}(V') \rangle^{\geq m-1}$	if $m > 0$

Table 2.4: Some Useful DL−PA Formulae

$H(\varphi, \geq m)$ is $true$ if and only the Hamming distance between the current valuation and any model of φ is greater or equal to m. The predicate $H(\varphi, V', \geq m)$ has the same meaning when we only consider a subset of the variables from the formula, meaning that the values of the variables in $V_\varphi \backslash V'$ remain unchanged. This predicate is not used in the rest of this section, but it will be useful when we will present the application of DL−PA to update argumentation frameworks (see Section 3.5).

Now, Dalal's revision operator and Forbus' update operator can be defined through DL−PA.

Proposition 10 (Dalal's Operator Expressed in DL−PA)**.**

Let $\pi^D_{\alpha,\beta}$ be the following DL-PA *program:*

$$\mathtt{flip}^{\geq 0}(V_\varphi); \varphi?; \left(\bigcup_{0 \leq m \leq |V_\alpha|} [\mathtt{flip}^{\geq 0}(V_\varphi); \varphi?] H(\alpha, \geq m)?; \mathtt{flip}(V_\alpha)^m \right); A?$$

Then $\varphi \circ_D \alpha = \mathrm{Mod}((\neg \mathtt{Sat}(B) \wedge \alpha) \vee \langle (\pi^D_{\alpha,\beta})^- \rangle \top)$.

Proposition 11 (Forbus' Operator Expressed in DL-PA)**.**
Let π^F_α be the following DL-PA *program:*

$$\left(\bigcup_{0 \leq m \leq |V_\alpha|} H(\alpha, \geq m)?; \mathtt{flip}(V_\alpha)^m \right); A?$$

Then $\varphi \diamond_F \alpha = \mathrm{Mod}(\langle (\pi^F_\alpha)^- \rangle \varphi)$.

2.3 Conclusion

This chapter introduces the belief change theory. The well-known AGM framework, which represents the beliefs of an agent as a deductively closed set of formulae, is one of the most influencial works on this topic. In particular, for revising a theory by a formula, we present the system of spheres approach. This method is based on a ranking between the possible worlds, and it keeps as the result of revision the minimal (with respect to this ranking) possible worlds which agree with the revision formula. This approach has been adapted for the case of finite propositional logic, by Katsuno and Mendelzon. This KM revision approach is at the origin of two of the contributions described in this thesis. This chapter also presents the ranking-based approach for belief update, in order to be able to distinguish the differences between the revision of argumentation frameworks discribed in this thesis and some update approaches proposed by other authors.

Chapter 3

Existing Approaches on Dynamics of Argumentation Frameworks

He who knows others is wise; he who knows himself is enlightened.

Lao-Tzu

The dynamics of argumentation frameworks has been widely studied in the recent years. Such scenarios, which make evolve an argumentation framework to incorporate a new piece of information, is likely to happen in many different applications, as we have explained in the Introduction, and exemplified in Section 1.3 with some "toy" examples. There are many other cases in which an agent may receive a new piece of information, and different ways to incorporate it in her argumentation frameworks. This chapter presents the previous contributions on this topic.

We start this presentation by some of the first studies on the impact on change in argumentation frameworks, concerning the impact on the extensions of addition or deletion of an argument (with the set of attacks related to it) or an attack in an argumentation framework.

Then we present the expansion of an argumentation framework, which is the addition of some new arguments and attacks to the argumentation framework respecting some constraints. We show how the different kinds of expansion can be used to enforce a set of arguments as an extension, and present a characterization of minimal change for these enforcement operations.

We also present some works which make a link between change in argumentation frameworks and change in causal Bayesian networks. Another interesting approach is a method to compute the minimal number of changes to perform in an argumentation framework to ensure that some goal on the acceptance statuses is reached.

Finally, we present some approaches which are more related to our work. These works take advantage of belief change theory to update or revise argumentation frameworks.

We conclude this chapter by a brief presentation of the new kinds of change we want to perform in an argumentation framework, and why they are motivated by reasonable applications.

Contents

3.1 Properties of Atomic Change in Argumentation Frameworks

We call an *atomic* change in an argumentation framework the addition (or removal) of atomic elements of the framework, meaning either an attack, or an argument (with the set of attacks concerning it). The first works on the topic of change in argumentation frameworks studied the effects of such a change, depending on the semantics. Contrary to approaches based on belief change theory, here the authors study the consequences on the set of extensions (which is the outcome of the framework) of a particular change on the structure on the graph.

In particular, [BKvdT09b, BKvdT09a] define some properties which can be satisfied when a refinement (that is, the addition of arguments or attacks) or an abstraction (that is, the removal of arguments or attacks) is performed. These properties are called *refinement principles* and *abstraction principles*, and the authors list which ones are satisfied when the considered semantics is the grounded semantics. In a nutshell, these principles establish that if the refinement or abstraction which is performed satisfies some conditions, then the extension of the argumentation framework stays unchanged (all these principles consider only single-extension semantics, in particular the grounded semantics).
Since the authors of these works only consider single-extension semantics like the grounded semantics, we use the notation $in_\sigma(F)$ to denote the arguments which are accepted by the argumentation framework F with respect to the chosen semantics (meaning, the ones which belong to the unique extension), $out_\sigma(F)$ denotes the arguments which are rejected (meaning, the ones which are attacked by an argument from the extension) and $undec_\sigma(F)$ the other ones.

Then, we describe the works from [CdSCLS10, BCdSCLS11], in which the authors present some properties which can be satisfied by atomic changes and study their satisfaction by some of the most usual semantics. In particular, they present a typology of the effect that such a change can have on the set of extensions (modification of their number, inclusion relation between the former ones and the new ones,. . .) or the statuses of arguments.

3.1.1 Refinement and Abstraction Principles

Refinement of an Argumentation Framework

First, let us define formally a refinement [BKvdT09b].

Definition 31 (Refinement)**.**
Let $F = \langle A, R\rangle$ and $F' = \langle A', R'\rangle$ be two argumentation frameworks.

- F' is an *argument refinement* from F if and only if $A \subseteq A'$, and $\forall a_i, a_j \in A, (a_i, a_j) \in R'$ only if $(a_i, a_j) \in R$.
- F' is an *attack refinement* from F if and only if $A = A'$ and $R \subseteq R'$.
- F' is an *argument-attack refinement* from F if and only if $A \subseteq A'$ and $R \subseteq R'$.

Now, let us present the different refinement principles introduced by Boella, Kaci and Van der Torre.

Definition 32 (Attack Refinement Principles)**.**
A semantics σ satisfies the $X - Y$ attack refinement principle, with $X, Y \in \{in, out, undec\}$, if for every argumentation framework $F = \langle A, R\rangle$, $\forall a_i \in X_\sigma(F), \forall a_j \in Y_\sigma(F)$, $in_\sigma(\langle A, R \cup \{(a_i, a_j)\}\rangle) = in_\sigma(F)$.

It holds that the grounded semantics satisfies five among the nine attack refinement principles following this definition.

Proposition 12.
The grounded semantics satisfies the $in - out$, $out - out$, $undec - out$, $out - undec$ and $undec - undec$ attack refinement principles. It does not satisfy the other ones.

Concretey, given X and Y, a semantics satisfies the $X - Y$ attack refinement principle when it is possible to add an attack from any argument which is labelled X to any argument which is labelled Y, without modifying the content of the extension. It expresses a kind of indifference of the semantics to this particular addition of attacks. All of the principles states by the authors are similar to this one: given the fact that the input argumentation framework satisfies some properties, and given the statuses of some arguments, it is possible to add or remove arguments or attacks without changing the extension.
For instance, the next principle considers the fact that adding an attack may create a cycle in the argumentation framework, which could change the extension. Then, the extension remains the same only if it is sure that the new attack will not create such a cycle.

Definition 33 (Acyclic Attack Refinement Principles)**.**
A semantics σ satisfies the acyclic $X - Y$ attack refinement principle, with $X, Y \in \{in, out, undec\}$, if for every argumentation framework $F = \langle A, R\rangle$, $\forall a_i \in X_\sigma(F), \forall a_j \in Y_\sigma(F)$, if there is no odd-length sequence of attacks from a_j to a_i, then $in_\sigma(\langle A, R \cup \{(a_i, a_j)\}\rangle) = in_\sigma(F)$.

At least one of the corresponding principles is satisfied by the grounded semantics.

Proposition 13.
The grounded semantics satisfies the acyclic $out - in$ attack refinement principle.

We do not analyse in depth all the principles proposed by Boella, Kaci and Van der Torre. Let us just illustrate the abstraction principles.

Abstraction Principles

Now, we present the counterpart of refinement principles for abstraction, that is the removal of some elements from the argumentation framework [BKvdT09a].

Definition 34 (Abstraction)**.**
Let $F = \langle A, R\rangle$ and $F' = \langle A', R'\rangle$ be two argumentation frameworks.

- F' is an *argument abstraction* from F if and only if $A' \subseteq A$, and $\forall a_i, a_j \in A, (a_i, a_j) \in R'$ only if $(a_i, a_j) \in R$.
- F' is an *attack abstraction* from F if and only if $A = A'$ and $R' \subseteq R$.
- F' is an *argument-attack abstraction* from F if and only if $A' \subseteq A$ and $R' \subseteq R$.

Definition 35 (Attack Abstraction Principles)**.**
A semantics σ satisfies the $X - Y$ attack abstraction principle, with $X, Y \in \{in, out, undec\}$, if for every argumentation framework $F = \langle A, R \rangle$, $\forall a_i \in X_\sigma(F), \forall a_j \in Y_\sigma(F)$, $in_\sigma(\langle A, R \backslash \{(a_i, a_j)\}\rangle) = in_\sigma(F)$.

The authors identify among the nine corresponding abstraction principles which ones are satisfied by the grounded semantics.

Proposition 14.
The grounded semantics satisfies the $in - in$, $in - undec$, $undec - in$, $undec - out$, $out - in$, $out - undec$ and $out - out$ attack abstraction principles. It does not satisfy the other ones.

Similarly to the refinement principles, the authors propose some more elaborated abstraction principles, to establish in which condition the semantics is indifferent to the deletion of an attack, depending on the structure of the argumentation framework.
Then, the authors also propose several argument abstraction principles, to characterize what happens when an argument and the attacks which concern it are removed. First, let us introduce a notation for the set of attacks related to an argument: for each $a_i \in A$, $R_{a_i} = \{(a_j, a_k) \in R \mid a_j = a_i \text{ or } a_k = a_i\}$.

Definition 36 (Argument Abstraction Principles)**.**
A semantics σ satisfies the X argument abstraction principle, with $X \in \{in, out, undec\}$, if for every argumentation framework $F = \langle A, R \rangle$, if $a_i \in X_\sigma(F)$, then $in_\sigma(\langle A \backslash \{a_i\}, R \backslash R_{a_j}\}\rangle) = in_\sigma(F) \backslash \{a_i\}$.

Proposition 15.
The grounded semantics satisfies the out argument abstraction principle. It does not satisfy the in and $undec$ argument abstraction principles.

Intuitively, this means that rejected arguments do not matter for the evaluation of arguments statuses, and so they may be removed. Then, different situations where undecided and accepted arguments can be removed are listed in the paper. For more details about refinement and abstraction principles, we refer the reader to the original publications [BKvdT09a, BKvdT09b].

3.1.2 Adding or Removing an Argument

[CdSCLS10] describes different kinds of atomic changes that can occur in an argumentation framework, and presents some properties that can be satisfied by such a change operation. In particular, they are interested in the consequences of an atomic change on the set of extensions of the argumentation framework. Then, they focus on the properties which are satisfied by the addition of an argument (with some attacks concerning it), depending on different semantics.

Typology of Atomic Changes Several kinds of change on the structure of an argumentation framework can be performed. They concern both elements of the argumentation frameworks (arguments and attacks) and have two different natures (addition and removal).

Definition 37 (Atomic Change Operations)**.**
Let $F = \langle A, R\rangle$ be an argumentation framework. We define the following change operations:

- the addition of an attack (a_i, a_j) with $a_i \in A$ and $a_j \in A$ is defined by

$$F \oplus_i (a_i, a_j) = \langle A, R \cup \{(a_i, a_j)\}\rangle$$

- the removal of an attack (a_i, a_j) with $a_i \in A$, $a_j \in A$ and $(a_i, a_j) \in R$ is defined by

$$F \ominus_i (a_i, a_j) = \langle A, R \backslash \{(a_i, a_j)\}\rangle$$

- the addition of an argument $a_k \notin A$ with a set of attacks concerning it, noted R_{a_k}, is defined by:

$$F \oplus_i^a (a_k, R_{a_k}) = \langle A \cup \{a_k\}, R \cup R_{a_k}\rangle$$

  We suppose that R_{a_k} is a non-empty set of attacks concerning a_k, meaning that $R_{a_k} \subseteq A \times A$ such that $\forall (a_l, a_m) \in R_{a_k}$, $a_l = a_k$ or $a_m = a_k$.

- the removal of an argument $a_k \in A$ with the attacks concerning it is defined by:

$$F \ominus_i^a (a_k, R_{a_k}) = \langle A \backslash \{a_k\}, R \backslash R_{a_k}\rangle$$

  R_{a_k} is defined as the set $\{(a_k, a_l) \in R\} \cup \{(a_l, a_k) \in R\}$.

We remark that addition (respectively removal) of an argument or an attack correspond to the refinement (respectively the abstraction) of the argumentation framework, as defined by Boella *et al*.
The following results concern the addition and removal of an argument which interacts with the other ones ($R_{a_k} \neq \emptyset$), the remaining case being straightforward.
Properties presented below can be classified into two categories. They deal either with the structure of the set of extensions (for instance, their number or some inclusion relation between them) or the acceptance statuses of some arguments.

Structural Properties In the general case, extension-based semantics may lead to a non-unique result: several extensions are associated with the argumentation framework, and there is not a single way to decide if an argument is accepted. It is also possible that an argumentation framework does not lead to accept some arguments, since it does not admit a non-empty extension. So, a change is called decisive if and only if it allows to decide precisely which arguments are accepted. This means that the outcome of the change is an argumentation framework which admits a single non-empty extension.

Definition 38 (Decisive Change)**.**
The change from F to F' is called *decisive* with respect to the semantics σ if and only if $|Ext_\sigma(F)| > 1$ or $Ext_\sigma(F) \in \{\emptyset, \{\emptyset\}\}$ and $|Ext_\sigma(F')| = 1$, with $Ext_\sigma(F') \neq \{\emptyset\}$.

A weakened version of this property describes a change which reduces the number of extensions of the argumentation framework. This does not allow to decide if an argument a_i is accepted, but it is "easier" to accept an argument, since it requires a_i to belong to m extensions rather than n extensions (with $m < n$). From a computational point of view, it means that checking if a_i is skeptically accepted by the argumentation framework can be done more efficiently, since there are less extension to enumerate.

Definition 39 (Restrictive Change)**.**
The change from F to F' is called *restrictive* with respect to the semantics σ if and only if $1 < |Ext_\sigma(F')| < |Ext_\sigma(F)|$.

The opposite property concerns a change which brings some ambiguity in the evaluation of arguments statuses, since it increases the number of extensions.

Definition 40 (Questioning Change)**.**
The change from F to F' is called *questioning* with respect to the semantics σ if and only if $|Ext_\sigma(F)| < |Ext_\sigma(F')|$.

When the input argumentation framework admits at least one non empty extension and the output argumentation framework does not, the change is called destrictive since it is not possible anymore to accept arguments (even credulously), while it was the case before the change.

Definition 41 (Destructive Change)**.**
The change from F to F' is called *destructive* with respect to the semantics σ if and only if $|Ext_\sigma(F)| > 1$, $Ext_\sigma(F) \neq \{\emptyset\}$, and $Ext_\sigma(F') \in \{\emptyset, \{\emptyset\}\}$.

All these properties on change have an impact on the number of extensions. The following ones do not change the number of extensions, but change their content.

Definition 42 (Expansive Change)**.**
The change from F to F' is called *expansive* with respect to the semantics σ if and only if $|Ext_\sigma(F)| = |Ext_\sigma(F')|$ and $\forall \varepsilon' \in Ext_\sigma(F')$, $\exists \varepsilon \in Ext_\sigma(F)$ such that $\varepsilon \subset \varepsilon'$.

Then, a particular kind of change is the one which guarantees the standard equivalence between the argumentation frameworks.

Definition 43 (Conservative Change)**.**
The change from F to F' is called *conservative* with respect to the semantics σ if and only if $Ext_\sigma(F) = Ext_\sigma(F')$.

The last property about the structure of the set of extensions concerns changes which do not modify the number of extensions, but modify at least one of them.

Definition 44 (Altering Change)**.**
The change from F to F' is called *altering* with respect to the semantics σ if and only if $|Ext_\sigma(F)| = |Ext_\sigma(F')|$ and $\exists \varepsilon \in Ext_\sigma(F)$ such that $\forall \varepsilon' \in Ext_\sigma(F'), \varepsilon \not\subseteq \varepsilon'$.

Properties on the Arguments Statuses The authors of the study point out two properties on the arguments statuses. First, the monotony property expresses that the arguments which were accepted before the change are still accepted after it. Three different kinds of monotony can be defined.

Definition 45 (Monotony, Credulous Monotony, Skeptical Monotony)**.**

- The change from F to F' satisfies *monotony* with respect to the semantics σ if and only if each σ-extension of F is included in at least one σ-extension of F'.
- The change from F to F' satisfies *credulous monotony* with respect to the semantics σ if and only if the union of the σ-extensions of F is included in the union of the σ-extensions of F'.
- The change from F to F' satisfies *skeptical monotony* with respect to the semantics σ if and only if the intersection of the σ-extensions of F is included in the intersection of the σ-extensions of F'.

These properties are defined at the extensions level, but a counterpart at the arguments level is also defined.

Definition 46 (Partial Monotony for an Argument)**.**
Given a_i an argument, the change from F to F' satisfies *partial monotony for* a_i with respect to the semantics σ if and only if when a_i belongs to a σ-extension of F, it also belongs to a σ-extension of F'.

The second property about arguments statuses is borrowed from belief change *success postulate* [AGM85]. It is relevant when a new argument is added to the argumentation framework ($\oplus_i^a$).

Definition 47 (Priority to Recency)**.**
The change $\oplus_i^a$ from F to F' satisfies *priority to recency* with respect to the semantics σ if and only if F' admits at least one σ-extension, and the new argument a_k belongs to each σ-extension of F'.

Then, the authors prove that the different properties are related. In particular, if some properties on the structure of the set of extensions are satisfied by a given change operation, then it implies that this operation also satisfy some properties on the arguments statuses. Finally, this paper gives some results for the grounded and the preferred semantics, establishing some conditions to ensure that a the change operation $\oplus_i^a$ (that is, the addition of an argument with some attacks related to it) satisfies some of the properties that they have proposed.

Removal of an argument [BCdSCLS11] presents similar properties with the operation $\ominus_i^a$, which removes an argument and the attacks concerning it. Some of the properties studied in the case of the addition of an argument are still relevant for the case of the removal of an argument. They also present a new property which relevant to argument removal. This property is the dual of expansive change: rather than obtaining supersets of the initial extensions, now the extensions of the outcome are expected to be subsets of the initial extensions.

Definition 48 (Narrowing Change)**.**
The change from F to F' is called *narrowing* with respect to the semantics σ if and only if:

- $Ext_\sigma(F) \neq \emptyset$, $|Ext_\sigma(F)| = |Ext_\sigma(F')|$
- $\forall \varepsilon' \in Ext_\sigma(F')$, $\exists \varepsilon \in Ext_\sigma(F)$ such that $\varepsilon' \subset \varepsilon$
- $\forall \varepsilon \in Ext_\sigma(F)$, $\exists \varepsilon' \in Ext_\sigma(F')$ such that $\varepsilon' \subset \varepsilon$

For instance, if the extensions remain the same, except the argument a_k which is present in each extension of F (but obviously no in the extensions of $F \ominus_i^a a_k$), then the change is narrowing.

The authors give some sufficient conditions for argument removal to satisfy (or not satisfy) monotony.

Proposition 16 (Monotony of $\ominus_i^a$)**.**
When an argument a_k is removed from F, under the preferred, stable or grounded semantics,

- *if $\exists \varepsilon_i \in Ext_\sigma(F)$ such that $a_k \in \varepsilon_i$, then $\exists \varepsilon_j \in Ext_\sigma(F)$ such that $\forall \varepsilon' \in Ext_\sigma(F')$, $\varepsilon_i \not\subset \varepsilon'$*
- *if $\nexists \varepsilon_i \in Ext_\sigma(F)$ such that $a_k \in \varepsilon_i$, then $\forall \varepsilon_j \in E$, $\exists \varepsilon' \in Ext_\sigma(F')$ such that $\varepsilon_j \subseteq \varepsilon'$.*

A property of weak monotony, similar to argument abstraction principle, is also proposed.

Proposition 17.
When an argument a_k is removed from the argumentation framework F, is a_k does not attack any argument in F, then

- *for all preferred extension ε of F, $\varepsilon\backslash\{a_k\}$ is an admissible set of F', and then it exists a preferred extension ε' of F' such that $\varepsilon\backslash\{a_k\} \subseteq \varepsilon'$.*
- *for all stable extension ε of F, $\varepsilon\backslash\{a_k\}$ is a stable extension of F'.*

In a similar way to the results about the additioni of an argument, [BCdSCLS11] gives some properties of the removal operation $\ominus_i^a$ depending on the semantics and the properties of the input argumentation framework.

3.2 Extension Enforcement

Enforcing a set of arguments E is defined in [BB10] as a change from an argumentation framework F to another one F' such that E is an extension of F' or is included in an extension of F'. Several enforcement methods are presented, based on the notion of *expansion* of an argumentation framework. An expansion is the addition of new arguments and new attacks to an argumentation framework, respecting some constraints. The enforcement of E in F is then defined as an expansion of F such that E is (or is included in) an extension of the expansion.

3.2.1 Normal, Strong and Weak Expansion

[BB10] defines the expansion of an argumentation framework as the natural modification of this argumentation framework when two (or more) agents are debating about a subject. It consists in adding new arguments, with some attacks between them (and possibly between the new arguments and the former ones). Baumann and Brewka make the hypothesis that the agents agree about the existing arguments and the attacks between them, so no attack can be added between the arguments which belong to the argumentation framework before the expansion, and no attack can be removed. This is the ground for the definition of the different expansion approaches. Additional constraints on the attacks which can occur between the new arguments and the older ones lead to the definition of two other forms of expansions.

Definition 49 (Normal, Weak and Strong Expansion)**.**
Given an argumentation framework $F = \langle A, R\rangle$, $F' = \langle A', R'\rangle$ is a *normal expansion* of F (denoted $F \preceq^N F'$) if and only if $A \subset A'$ and $R' = R \cup R_{A'}$, with $R_{A'}$ a set of attacks $(a_i, a_j) \in A' \times A'$ such that either $a_i \in A'\backslash A$ or $a_j \in A'\backslash A$. Moreover,

- F' is a *weak expansion* of F (denoted $F \preceq^N_{,W} F'$) if and only if F' is a normal expansion of F such that there is no new attack directed from a new argument to a former one. Formally, $\forall (a_i, a_j) \in R_{A'}, (a_i, a_j) \notin (A'\backslash A) \times A$.
- F' is a *strong expansion* of F (denoted $F \preceq^N_{,W} F'$) if and only if F' is a normal expansion of F such that there is no new attack directed from an old argument to a new one. Formally, $\forall (a_i, a_j) \in R_{A'}, (a_i, a_j) \notin A \times (A'\backslash A)$.

A weak expansion only adds *weak arguments*, which are arguments that can be attacked by the former ones, but do not attack some of them (the weak arguments can possibly attack each other). Similarly, a strong expansion only adds *strong arguments*, which are not attacked by the older ones. These expansions are particular cases of argument refinement as described in Section 3.1.1, and argument addition as described in Section 3.1.2.

More generally, an expansion of an argumentation framework $F = \langle A, R\rangle$ is any argumentation framework $F' = \langle A', R'\rangle$ such that $A \subseteq A'$, $R \subseteq R'$, and $F \neq F'$ (denoted $F \preceq F'$).

Example 13.
Figure 3.1 describes an argumentation framework F_1 and several expansions of F_1. The argumentation frameworks F_2, F_3, F_4 and F_5 are expansions of F_1, but F_2 is not a normal one. F_4 is a weak expansion, while F_5 is a strong one.

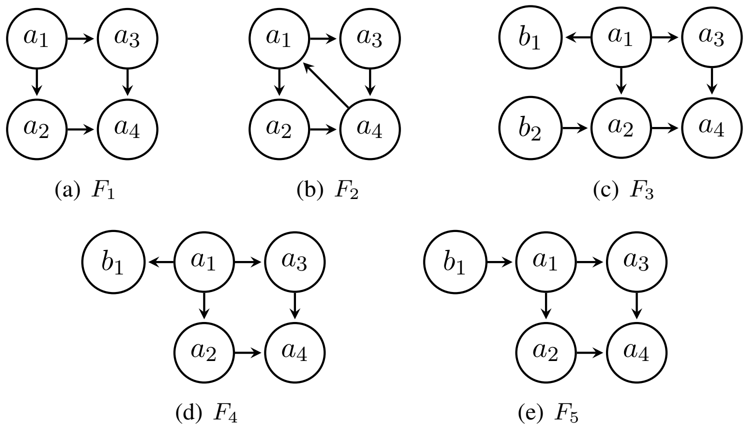


Figure 3.1: The Argumentation Framework F_1 and Different Possible Expansions of it

F_2 is not a normal expansion, since there is a new attack from a_4 to a_1. F_3 is a normal expansion, but not a weak one (since b_2 attacks a_2) nor a strong one (since b_1 is attacked by a_1). In F_4, b_1 is a weak argument, while it is a strong argument in F_5, so these expansions are respectively weak and strong.

In [Bau12], Baumann also introduces *arbitrary modifications*, which are more permissive than the expansions, since they allow deleting attacks between the former arguments. Formally, F' is an arbitrary modification of F (denoted $F\mathcal{U}F'$) if and only if $A \subseteq A'$ and $R \neq R'$.

3.2.2 Using Expansion to Enforce a Set of Arguments

Beyond the nature of expansion, two additional parameters must be made precise in order to define enforcement operators. First, enforcement can be *strict* when the expected set of arguments has to be exactly an extension of the output argumentation framework or *non-strict* when the set of arguments has to be included in an extension of the output argumentation framework. Then, enforcement can be *conservative* when the semantics stays the same one or *liberal* when the semantics may change.

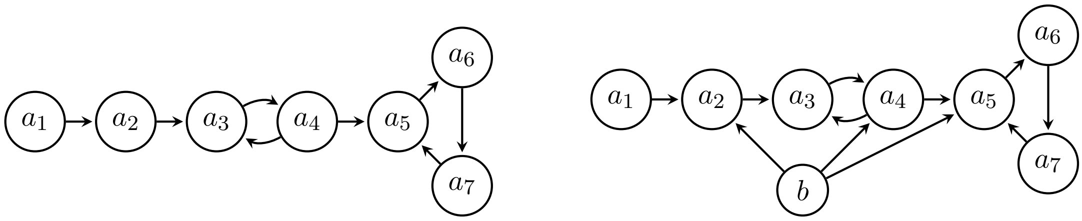
Definition 50.
Let $F = \langle A, R \rangle$ be an argumentation framework, σ an acceptability semantics, and $E \subseteq A$ a set of arguments. The *normal (respectively normal strict) enforcement operator* $+_\sigma^N$ (resp. $+_{\sigma,s}^N$) is defined as a mapping from F and E to an argumentation framework $F' = \langle A', R' \rangle$ such that F' is a normal expansion of F, and such that E is included in (respectively is exactly) an extension of F'. Moreover,

- if F' is a weak expansion of F, then $+_\sigma^{N,W}$ (respectively $+_{\sigma,s}^{N,W}$) is a *weak (respectively strict weak) enforcement operator*.
- if F' is a strong expansion of F, then $+_\sigma^{N,S}$ (respectively $+_{\sigma,s}^{N,S}$) is a *strong (respectively strict strong) enforcement operator*.

We use σ' to denote the semantics that the agent uses before performing an enforcement. The enforcement operator is called *conservative* if $\sigma = \sigma'$, and *liberal* otherwise.

We remark that, from a technical point of view, the difference between conservative and liberal enforcement does not influence the definition of the operator, as soon as the target semantics σ is fixed. But it is useful to know if an agent agrees to change her semantics (so, to change her way of reasoning), since enforcement may be impossible with respect to a given semantics, but possible with respect to another one. Let us illustrate the strong enforcement approach, and examplify the utility of liberal enforcement.

Example 14.
Let F_6 be the argumentation framework given in Figure 3.2(a). Its set of stable extensions is $Ext_{st}(F) = \{\{a_1, a_4, a_6\}\}$. The set of arguments expected to be enforced is $E = \{\{a_1, a_3\}\}$. A possible strong enforcement is presented in Figure 3.2(b): the stable extensions of F_7 are $Ext_{st}(F_7) = \{\{a_1, a_3, a_6, b\}\}$, and so E is enforced.



(a) F_6 (b) F_7

Figure 3.2: Strong enforcement process

Another possibility to enforce E is to switch the semantics from stable to preferred: the preferred extensions of F_6 are $Ext_{pr}(F_6) = \{\{a_1, a_4, a_6\}, \{a_1, a_3\}\}$, so E can be enforced just by applying liberal enforcement whose outcome is F_6 itself. In this case, liberal enforcement allows to obtain the expected result with less change than conservative enforcement.

3.2.3 Minimal Change Enforcement

We have explained previously that different kinds of enforcement may lead to the expected result with more or less required changes. In [Bau12], Baumann borrows minimal change principle from belief revision and studies the minimal change required to enforce a set of arguments, depending on the semantics and the kind of change which is permitted. A simple measure of difference between argumentation frameworks is used: the cardinality of the symmetric difference between the attack relations, which counts how many attacks must be added (or removed, if attacks removal are allowed) to a graph to make it identical to another one[10].

Definition 51 (Distance Between Argumentation Frameworks)**.**
Let $F = \langle A, R \rangle$ and $F' = \langle A', R' \rangle$ be two argumentation frameworks. The distance between F and F' is a natural number defined by the following function.

$$d(F, F') = |(R \backslash R') \cup (R' \backslash R)|$$

Then, Baumann introduces the notion of (σ, Φ)-characteristic of a set of arguments E, which is the minimal number of modifications to make in an argumentation framework F to enforce E in F. σ denotes the considered semantics, while Φ is a binary relation over argumentation frameworks which indicates which kinds of modification are allowed to enforce E (roughly speaking, which kind of expansion is used).

[10]This distance is obviously related to the Hamming distance. Indeed, if we consider interpretations as sets of variables which are assigned the value *true*, then the Hamming distance is equal to the cardinality of the symmetric difference between the interpretations.

Definition 52 ((σ, Φ)-characteristic)**.**
Given a semantics σ, a binary relation over argumentation frameworks Φ and an argumentation framework F, the (σ, Φ)-characteristic of E with respect to F is a natural number or infinity defined by the following function.

$$N^F_{\sigma,\Phi} = \begin{cases} 0 & \text{if} \quad \exists E' \in Ext_\sigma(F) \text{ such that } E \subseteq E' \\ \min(K, \leq) & \text{if} \quad K = \{d(F, F') \mid (F, F') \in \Phi \text{ and } N^{F'}_{\sigma,\Phi}(E) = 0\} \neq \emptyset \\ \infty & \text{otherwise} \end{cases}$$

Baumann defines so-called value functions, which are computable functions which leads to the (σ, Φ)-characteristic of a set of arguments.

Definition 53 ((σ, Φ)-value)**.**
Given an argumentation $F = \langle A, R\rangle$ and a set of arguments $E \subseteq A$, the (σ, Φ)-value of E with respect to F is defined by the following function. If $\sigma \in \{st, ad\}$ and $\Phi = \leq^N_W$, then

$$V^F_{\sigma,\Phi} = \begin{cases} 0 & \text{if} \quad \exists E' \in Ext_\sigma(F) \text{ such that } E \subseteq E' \\ \infty & \text{otherwise} \end{cases}$$

If $\sigma \in \{st, ad\}$ and $\Phi = \leq^N_S$, then

$$V^F_{\sigma,\Phi} = \min(\{|n_\sigma(F, E')| \mid E' \in cf(F) \text{ and } E \subseteq E'\} \cup \{\infty\})$$

where $n_{ad}(F, E') = \{a_i \in A \mid a_i \text{ attacks } E'\} \backslash \{a_i \in A \mid a_i \text{ is attacked by } E'\}$ and $n_{st}(F, E') = A \backslash \{a_i \in A \mid a_i \text{ is attacked by } E'\}$.
If $\sigma \in \{st, ad\}$ and $\Phi = \mathcal{U}$, then

$$V^F_{\sigma,\Phi} = \min(\{|R_{\downarrow E'}| + |\sigma(F, E')| \mid E \subseteq E' \subseteq A\})$$

where $R_{\downarrow E'} = R \cap (E' \times E')$.

Intuitively, the $|n_\sigma(F, E')|$ which is used in the previous definition counts the number of arguments which must be attacked in F to make E' a σ-extension of the result. In the case of admissible semantics, this corresponds to the number of arguments which attack E' such that E' does not defend itself against them. For the stable semantics, it is the number of arguments which are not attacked by E', since E' should attack every other argument to be a stable extension.

Table 3.1 sums up the different (σ, Φ)-characteristics.

$N^F_{\sigma,\Phi}$	$\leq^N_W$	$\leq^N_S, \leq^N, \leq$	$\mathcal{U}$
st	$V^F_{st,\leq^N_W}$	$V^F_{st,\leq^N_S}$	$V^F_{st,\mathcal{U}}$
pr, co, ad	$V^F_{ad,\leq^N_W}$	$V^F_{ad,\leq^N_S}$	$V^F_{ad,\mathcal{U}}$

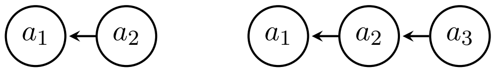
Table 3.1: (σ, Φ)-characteristics

Baumann also establishes that weak enforcement may be "harder" in term of attacks to add, than strong enforcement.

Proposition 18.
For any argumentation framework F and any semantics $\sigma \in \{st, ad\}$, $V_{\sigma,\leq_W^N} \geq V_{\sigma,\leq_S^N}$.

Among other reasons, a possible explanation is the fact that some enforcements may be impossible with a weak expansion, while they are simple to realize with a strong expansion.

Example 15.
In the argumentation framework F_8, whatever the semantics, a_1 is rejected. It is impossible to enforce the set $\{a_1\}$ by adding only weak arguments, while a single change on the attack relation proves enough if strong arguments can be added. So, $\{a_1\}$ is enforced in F_9.



(a) F_8 (b) F_9

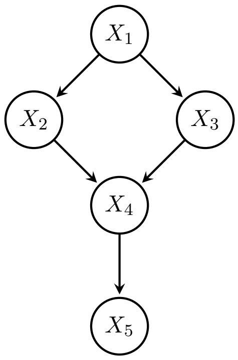
Figure 3.3: Enforcing $\{a_1\}$ in F_8 is impossible with weak expansion, while F_9 is a succesful strong enforcement.

3.3 Intervention and Observation in Argumentation

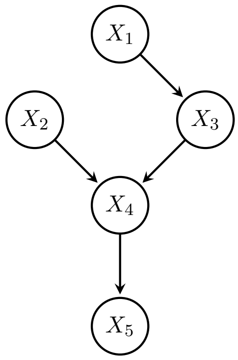
An interesting idea studied recently about change in argumentation frameworks is the use of the concepts of observation and intervention, very familiar to the researchers working on Causal Bayesian Networks. Let us briefly recall what is a Causal Bayesian Network. Such a network is a structure based on two components: random variables which are linked to each others through a causality relation (the variable X is a parent of Y means that X being true is a possible cause for Y being true), and some conditional probabilities associated with the variables. We will not give an in-depth presentation of Causal Bayesian Networks, and we refer the reader to [Pea09] for more details.
In this setting, it is admitted that an information about a variable can have two reasons: either the agent has observed that some variable X is in a particular state, which can be explained by the fact that some of the possible causes Y has been affected to some value which is the origin of the state of X, or the agent has perform an action on the system which explains the state of X. This kind of situation does not allow to deduce any information about Y. Let us illustrate this on an example borrowed from [Pea09].

Example 16.
Figure 3.4(a) shows a Causal Bayesian Network representing the relationships between the season ($X_1 \in \{wet, dry\}$), the state of the sprinkler ($X_2 \in \{on, off\}$), the rain ($X_3 \in \{yes, no\}$), the state of the pavement ($X_4 \in \{wet, dry\}$) and the fact the the pavement is slippery or not ($X_5 \in \{yes, no\}$).

If any observation about the value of a variable occurs, we can deduce some information about the children of the variable in the graph (for instance, if we observe that the sprinkler is switched on, then the pavement is more probably wet, and so it is more probably slippery), but also about the parents of the variable (if the sprinkler has been switched on, the season is very likely to be dry).
If the state of the variable is changed by an intervention of the agent, we cannot deduce anything about the parents of the variable (the agent can choose to switch on the sprinkler even if the season is wet). To represent the fact that we know that the season is not a cause of the sprinkler being turned on, we disconnect the variable from its parents. Figure 3.4(b) shows the result of this intervention in the previous Causal Bayesian Network.



(a) The Original Causal Bayesian Network



(b) The Causal Bayesian Network after the Intervention

Figure 3.4: The Result of an Intervention on the Causal Bayesian Network

In his thesis, Tjitze Rienstra [Rie14] borrows these ideas to adapt them to the case of abtract argumentation. When an agent learns a new piece of information about the acceptance status of an argument, it can be the case that some new argument has been added to the argumentation framework, changing the evaluation of the already known arguments. Let us remark that this framework supposes that the attacks between the previously known arguments are fixed. Similarly to the inference in Causal Bayesian Networks, this change can be seen as an intervention (for instance, the agent adds an attacker to the argument a_1 to make it rejected) or an observation (so the agent knows that some argument has been added in the graph, changing the status of a_1 and possibly the status of some parents of a_1).

Let us borrow Rienstra's example to illustrate how intervention and observation differ in argumentation frameworks.

Example 17.
Let $F = \langle A, R \rangle$ be the argumentation framework described in Figure 3.5. It is obvious that the accepted arguments are a_1 and a_3. Now, let us suppose that the expected change of status is that a_1 has to

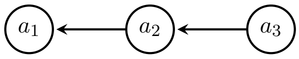


Figure 3.5: The Initial Argumentation Framework

be rejected. If this change of status is due to an intervention, Rienstra explains that the only possible option is to add an attack to a_1 (as described on Figure 3.6): it is the agent herself who performed the intervention which explains the new status of a_1. With this kind of process, only the status of a_1 and its possible descendant in the graph are affected. We see that the statuses of a_2 and a_3 have not changed.

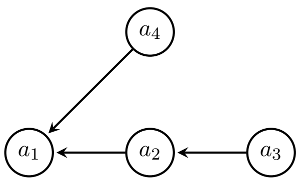
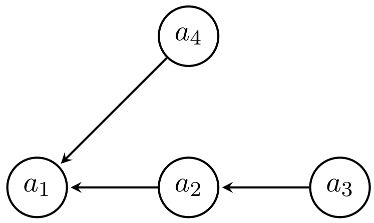
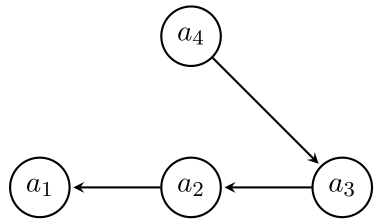


Figure 3.6: The Argumentation Framework after an Intervention

If the agent does not perform the intervention to change a_1 status, but makes an observation that a_1 status has changed, there are two possible changes in the argumentation framework to explain this new status: either a_1 is attacked by some new argument (which is the same change as the one required by the intervention, described on Figure 3.7(a)), or some new argument attacks a_3, as described on Figure 3.7(b).



(a) The New Argument Attacks a_1



(b) The New Argument Attacks a_3

Figure 3.7: Two Possible Explanations for the Observation

With an observation of a change of status for the argument a_1, there is still some possibility of status change of a_1 descendant in the graph, but unlike the case of the intervention, it is also possible to have some change of status on a_1 ascendants: in the argumentation framework given on Figure 3.7(b), the statuses of a_2 and a_3 have changed.

So, Rienstra defines two types of entailment relations: *intervention-based entailment* is concerned with the consequences of an hypothetical action performed in the argumentation framework (in particular, the addition of an argument which attacks some existing ones), while *observation-based entailment* is concerned with the consequences of a new piece of information about the acceptance of some argument.

Briefly, Rienstra defines these entailment relations which answer the question "Given an argumentation framework F, an intervention I (respectively an observation O) and a piece of information about the arguments statuses φ, does φ hold in F with the intervention I (respectively the observation O)?". He studies the properties of such entailment relations, in particular with respect to the Kraus, Lehmann and Magidor properties [KLM90, LM92].

3.4 Goal-Oriented Change in Argumentation

In [KBM+13], the authors tackle the problem, in a debate between several agents, of changing an argumentation framework which represents the current state of the debate, to ensure that a given goal (concerning the acceptance of an argument) is reached. The changes which are allowed are addition and removal of attacks between the arguments. First, let us describe the representation of the state of the debate.

Definition 54 (State of a Debate)**.**
A state of a debate is represented by a tuple $\langle F, R^+, R^- \rangle$, with $F = \langle A, R \rangle$ an argumentation framework $R^+ \subseteq (A \times A) \backslash R$ the set of attacks which can be added to the argumentation framework, and $R^- \subseteq R$ the set of attacks which can be removed from the argumentation framework.

Example 18 (Borrowed from [KBM+13])**.**
Let F be the state of the debate given at Figure 3.8. The dotted attacks represent the ones which can be removed from the debate, while the dashed ones can be added.

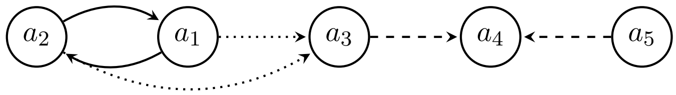


Figure 3.8: The State of the Debate S

The authors consider that the attacks are the core component of the argumentation framework, so they choose to work with *attacks semantics*, introduced by [VBvdT11]. Rather than associating each argument with an acceptance status, this kind of semantics associates each attack in the argumentation framework with a "success" status: an attack is labelled 1 when the first argument of the pair is in with respect to the considered complete labelling, ? if the first argument is $undec$, and 0 in the remaining case. For instance, with the simple argumentation framework containing two arguments a_1 and a_2 attacking each other, the complete labellings are $\{(a_1, in), (a_2, out)\}$, $\{(a_1, out), (a_2, in)\}$ and $\{(a_1, undec), (a_2, undec)\}$. Each of them can be associated, respectively, with the following evaluations of attacks: $\{((a_1, a_2), 1), ((a_2, a_1), 0\}$, $\{((a_1, a_2), 0), ((a_2, a_1), 1\}$ and $\{((a_1, a_2), ?), ((a_2, a_1), ?\}$. The attack is called *unsuccessful* when it is labelled 0 (since the attacker does not succeed in making the other argument rejected). It is called *successful* in the remaining cases.

Then, the authors define a language composed of atoms concerning actions to perform on the state of the debate.

Definition 55 (Atom)**.**
For $x \in A \times A$ an attack, and a_i an argument,

- the atom $(x, +, \#)$ (respectively $(x, -, \#)$) means that the attack x must be added to (respectively removed from) the argumentation framework);
- the atom $(x, X, *)$ (with $X \in \{1, 0, ?\}$) means that the agent must find a way to change the argumentation framework to ensure that the attack x is labelled X;
- the atom $(x, X, \#)$ (with $X \in \{1, 0, ?\}$) means that the attack x is actually labelled X in the argumentation framework;
- the atom $PRO(a_i)$ (respectively $CON(a_i)$) means that the agent wants the argument a_i to be accepted (respectively rejected);
- the atom $\top$ means success of the procedure, while the atom $\bot$ means failure.

A move is a set of $(x, +, \#)$ and $(x, -, \#)$ atoms, which indicates the attacks to be respectively added and removed from the argumentation framework.

Definition 56 (Move)**.**
The set $m = \{(x, s, \#) \mid x \in (A \times A)$ and $s \in \{+, -\}\}$ is called a move on the state of the debate if and only if for each $(x, +, \#) \in m$, $x \in R^+$, and for each $(x, -, \#) \in m$, $x \in R^-$.

Example 18 Continued.
The move $m = \{((a_1, a_3), +, \#), ((a_5, a_4), -, \#)\}$ from S leads to the state of the debate S' given at Figure 3.9: (a_5, a_4) is not anymore in the state of the debate, while (a_1, a_3) has been added to the debate.

These moves make possible for the debate to satisfy a given goal. A *positive goal* is a property, expressed by a set of atoms, which is expected to be satisfied by the debate. On the opposite, a *negative goal* is a property which is expected not to be satisfied. Then, the best moves to perform to satisfy the goal are identified.

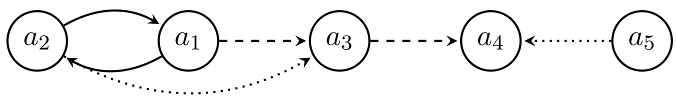


Figure 3.9: The State of the Debate S'

Definition 57 (Successful Move, Target Set)**.**
m is called a *successful move* for the goal g in the state of the debate S if and only if the application of m on S leads to a new state of the debate S' such that g is satisfied.
m is a *target set* for the goal g in the state of the debate S if and only if m is minimal with respect to the inclusion among all the successful moves for g in S.

Example 18 Continued.
The move m given previously is a successful move for the goal g "make a_4 credulously accepted for the preferred semantics", since $\{a_1, a_4, a_5\}$ is a preferred extension of the argumentation framework after the move.
But it is not successful for the goal g' ”make a_4 skeptically accepted for the preferred semantics", since $\{a_2, a_3, a_5\}$ is also a preferred semantics of the argumentation framework after the move.

Then, the authors focus on two types of goals:

- positive goals which consist in requiring that a particular argument a_i is credulously (respectively skeptically) accepted by the argumentation framework with respect to the semantics σ, noted $\sigma_\exists(a_i)$ (respectively $\sigma_\forall(a_i)$);
- negative goals which consist in requiring that a particular argument a_i is not credulously (respectively skeptically) accepted by the argumentation framework, noted $\neg\sigma_\exists(a_i)$ (respectively $\neg\sigma_\forall(a_i)$).

For both kinds of goals, the semantics is the complete semantics, the preferred semantics, or the admissible semantics. The computation of the target sets for these goals is then performed through a Maude program [CDE+99]. Since we do not present Maude in this thesis, the reader may refer to the original publication for technical details [KBM+13].
Let us still remark that the authors have proven that their procedure, called *RP* (for *Rewritting Procedure*) always terminates. They have studied its correctness and completeness. Formally:

Proposition 19 (Correctness and Completeness of *RP*)**.**
The procedure RP *is correct for successful moves with respect to a given goal g if and only if every move it returns is actually a successful move for g.*
The procedure RP *is complete for target sets with respect to a given goal g if and only if it returns each target set for g.*
Table 3.2 gives the goals for which the procedure is correct, complete, or none.

We notice that the procedure cannot guarantee to lead only to correct result, so it could return some moves which are not successful for the given goal. In a similar way, the procedure do not guarantee to return each target set. It depends, of course, on the goal. For some goals, like making an argument credulously accepted with respect to the admissible semantics, the procedure is both correct and complete. But the opposite may also happen, as it is the case for the preferred semantics: the procedure is neither correct nor complete if the goal is to make an argument credulously accepted with respect to the preferred semantics.

Goal	Correctness for Successful Moves	Completeness for Target Sets
$ad_\exists$	Yes	Yes
$pr_\exists$	No	No
$co_\exists$	No	Yes
$\neg ad_\exists$	No	No
$\neg pr_\exists$	No	?
$\neg co_\exists$	Yes	Yes

Table 3.2: Correctness and Completeness Results for *RP*

3.5 Change in Argumentation through Belief Update

Both works presented in this section use some ideas from belief change theory, in particular belief update, to perform some change in an argumentation framework. They perform the update of an argumentation framework through a logical encoding, and then use propositional update operators to reach their goal.

3.5.1 Updating an Argumentation Framework through Propositional Encoding

The approach proposed in [BCdSL13] shows how to take advantage of belief update in propositional logic to define update operators suited to argumentation frameworks. The authors suppose that the agent has at her disposal a propositional language $\mathcal{L}_A$ which is able to describe an argumentation framework and its set of accepted arguments (the notion of acceptance can be tuned by a semantics, and an inference policy among credulous and skeptical). For any formula $\varphi \in \mathcal{L}_A$, the models of φ are the argumentation frameworks for which φ is true. This notion is not clearly defined, since the propositional language $\mathcal{L}_A$ is not fixed by the authors, but depends of the agent. If F is an argumentation framework, $F \models \varphi$ means that $F \in \text{Mod}(\varphi)$, and $\varphi \models \psi$ means that $\text{Mod}(\varphi) \subseteq \text{Mod}(\psi)$. A formula which admits several models expresses the uncertainty of the agent about the state of the world (each model being a possible argumentation framework representing the world).

Then, any kind of update cannot be performed. In the framework described by the authors, it is supposed that the agents have in their common knowledge a fixed set of arguments Arg and a fixed set of attacks $Rel \subseteq Arg \times Arg$. Each argumentation framework $F = \langle A, R \rangle$ is composed of $A \subseteq Arg$ and $R \subseteq Rel \cap (A \times A)$. So, an authorized change from F_1 is an addition or removal of arguments (and the corresponding attacks) such that the resulting argumentation framework F_2 is legal with respect to Arg and Rel. These changes are called *executable operations*. An *executable program* is a sequence $(o_1, \ldots, o_n)$ of operations such that o_1 is executable from the current argumentation framework, and for each i ($1 < i \leq n$), o_i is executable on the outcome of o_{i-1}. Since it may be the case that each executable operation (or program) is not permitted for an agent, the authors introduce the notion of *allowed transitions*, which are a subset of all the pairs (F_1, F_2) such that F_2 can be reached from F_1 through an executable program.

Before introducing the rationality postulates, we introduce a last required notation: $\forall \varphi, \psi \in \mathcal{L}_A$, a transition in the set T is possible between a set of models of φ and a set of models of ψ, denoted $(\varphi, \psi) \models T$, if and only if $\text{Mod}(\varphi) \neq \emptyset$ and $\forall F \in \text{Mod}(\varphi)$, $\exists F' \in \text{Mod}(\psi)$ such that $(F, F') \in T$.

Now, let us give the set of rationality postulates, adapted from the KM postulates for belief update. Here, the authors change some postulates to take into account the constraints on the allowed transitions for the agent.

(U1) $\varphi \diamond_T \alpha \models \alpha$

(U2) If $\varphi \models \alpha$, then $\varphi \diamond_T \alpha \equiv \varphi$

(E3) $\text{Mod}(\varphi \diamond_T \alpha) \neq \emptyset$ if and only if $(\varphi, \alpha) \models T$

(U4) If $\varphi \equiv \psi$ and $\alpha \equiv \beta$, then $\varphi \diamond_T \alpha \equiv \psi \diamond_T \beta$

(E5) If $|\text{Mod}(\varphi)| = 1$, then $(\varphi \diamond_T \alpha) \wedge \beta \models \varphi \diamond_T (\alpha\beta)$

(E8) If ($\text{Mod}(\varphi) \neq \emptyset$ and $\text{Mod}(\varphi \diamond_T \alpha) = \emptyset$) or ($\text{Mod}(\psi) \neq \emptyset$ and $\text{Mod}(\psi \diamond_T \alpha) = \emptyset$) then $\text{Mod}((\varphi \vee \psi) \diamond_T \alpha) = \emptyset$, else $\text{Mod}((\varphi \vee \psi) \diamond_T \alpha) = \text{Mod}((\varphi \diamond_T \alpha) \vee (\psi \vee \alpha))$

(U9) If $|\text{Mod}(\varphi)| = 1$, then $\text{Mod}((\varphi \diamond_T \alpha) \wedge \beta) \neq \emptyset$ implies $\varphi \diamond_T (\alpha \wedge \beta) \models (\varphi \diamond_T \alpha) \wedge \beta$

As usual, the representation theorem uses pre-orders between interpretations.

Definition 58 (Assignment Respecting T)**.**
Given T a set of allowed transitions, an *assignment respecting* T is a function that associates with each argumentation framework F a total pre-order $\leq_F$ such that for each argumentation frameworks F_1, F_2, if $(F, F_1) \in T$ and $(F, F_2) \notin T$, then $F_2 \not\leq_F F1$.
Moreover, the assignment is *faithful* if and only if $\forall F_1 \neq F$, $F <_F F_1$.

Proposition 20 (Representation Theorem)**.**
*Given T a set of allowed transitions, there is an update operator $\diamond_T$ satisfying **(U1)**, **(E3)**, **(U4)**, **(E5)**, **(E8)** and **(U9)** if and only if there is an assignment respecting T such that for each argumentation framework F, for each formulae φ, α,*

$$\text{Mod}(f(F) \diamond_T \alpha) = \{F_1 \in \text{Mod}(\alpha) \mid (F, F_1) \in T \textit{ and } \forall F_2 \in \text{Mod}(\alpha) \textit{ such that } (F, F_2) \in T, F_1 \leq_F F_2\}$$

with $f(F)$ the formula whose single model is F, and

$$\text{Mod}(\varphi \diamond_T \alpha) = \begin{cases} \emptyset & \textit{if } \exists F \in \text{Mod}(\varphi) \textit{ such that } \text{Mod}(f(F) \diamond_T \alpha) = \emptyset \\ \bigcup_{F \in \text{Mod}(\varphi)} \text{Mod}(f(F) \diamond_T \alpha) & \textit{otherwise} \end{cases}$$

*Moreover, the operator satisfies **(U2)** if and only if the assignment is faithful.*

3.5.2 Updating Argumentation Frameworks through DLPA programs

We already presented the Dynamic Logic of Propositional Assignments (DL-PA) and its application for belief change in Section 2.2.4. In [DHP14], the authors use this setting to encode argumentation frameworks and make them evolve using update operators. These encodings are based on a propositional language $\mathcal{L}_{att,in}$ which is built on the set of variables

$$ATT_A \cup IN_A$$

where $ATT_A = \{att_{a_i,a_j} \mid a_i, a_j \in A\}$ and $IN_A = \{in_{a_i} \mid a \in A\}$, with the usual connectives. att_{a_i,a_j} means that there is an attack from the argument a_i to the argument a_j in the considered argumentation framework, while in_{a_i} means that the argument a_i belongs to an extension of the argumentation framework. $\mathcal{L}_{att}$ is the restriction of the language to att_{a_i,a_j} variables, and $\mathcal{L}_{in}$ is the restriction to in_{a_i} variables. Then, encodings of the most usual semantics borrowed from [BD04] are defined in this propositional language. We do not give these encodings in details here, since similar ones are presented in

Chapter 5.

DL-PA can be used to compute the σ-extensions of an argumentation framework $F = \langle A, R\rangle$ through the program $\texttt{makeExt}^{\sigma}_{A} = \texttt{flip}^{\geq 0}(\{in_{a_i} \mid a_{\in} A\}); \Phi^{\sigma}_{A}?$ where the formula Φ^{σ}_{A} is the propositional encoding of the semantics. A model of this program leads to a valuation of the in_{a_i} variables which satisfies the formula Φ^{σ}_{A}, that is a σ-extension of F.
Then, DL-PA programs are defined to reach the main purpose of the paper: modifying an argumentation framework. Modifications of the structure of the framework are very simple to implement in DL-PA. If we want to incorporate a piece of information such that $(\bigwedge_{a_i,a_j} att_{a_i,a_j}) \wedge (\bigwedge_{a_k,a_l} \neg att_{a_k,a_l})$ in the framework, a simple DL-PA program which assigns the value *true* to the att_{a_i,a_j} variables and *false* to the att_{a_k,a_l} proves enough.

Now, let us explain how DL-PA programs are used to update the extensions of the argumentation framework. Two kinds of update operators are defined. The piece of information to incorporate into the framework is a formula φ about arguments statuses (which means, a formula built on the in_{a_i} variables). So the update operator can modify the argumentation framework to ensure that φ is satisfied by at least one extension of the result, which is called credulous enforcement of φ, or by each extension of the result, which is called skeptical enforcement of φ.
Both programs are adaptations of Forbus' update operator described in Section 2.2.4. The difference is that the credulous (respectively skeptical) enforcement operator only flips att_{a_i,a_j} variables, rather than flipping any variable of the vocabulary, and then it tests if formula φ is satisfied after at least one (respectively after each) execution of the program $\texttt{makeExt}^{\sigma}_{A}$.

Proposition 21 (Credulous and Skeptical Enforcement)**.**
Given an argumentation framework $F = \langle A, R\rangle$ and a formula $\varphi \in \mathcal{L}_{in}$, the following DL-PA *program modifies F to ensure that φ is satisfied by at least one extension of the resulting argumentation framework:*

$$\left(\bigcup_{0\leq m\leq |ATT_A|} H(\langle\texttt{makeExt}^{\sigma}_{A}\rangle\varphi, ATT_A, \geq m)?; \texttt{flip}(ATT_A)^m\right); \langle\texttt{makeExt}^{\sigma}_{A}\rangle\varphi?$$

Given an argumentation framework $F = \langle A, R\rangle$ and a formula $\varphi \in \mathcal{L}_{in}$, the following DL-PA *program modifies F to ensure that φ is satisfied by each extension of the resulting argumentation framework:*

$$\left(\bigcup_{0\leq m\leq |ATT_A|} H([\texttt{makeExt}^{\sigma}_{A}]\varphi, ATT_A, \geq m)?; \texttt{flip}(ATT_A)^m\right); [\texttt{makeExt}^{\sigma}_{A}]\varphi?$$

3.6 Change in Argumentation through Belief Revision

Now we focus on the works that adapt belief revision theory to abstract argumentation. In particular, [BKRvdT13] and [BB15] use the AGM framework and adapt it to abstract argumentation[11]. We also describe in part 3.6.2 a work which adapts a syntax-based setting for belief revision.

[11]There exists a third similar approach, which is described in Chapter 4, since it is directly related to the contribution of this Chapter.

3.6.1 A Labelling-based Integrity Constraint

Adding an integrity constraint to an argumentation framework is not completely new. Indeed, [CDM06] uses a propositional formula built on the set of arguments as a constraint to be satisfied by the outcome of the argumentation framework. The approach proposed by [BKRvdT13] is different. There, the integrity constraint does not impact the outcome of the argumentation framework, which is computed using the usual Dung's semantics. The proposal is to modify the argumentation framework to ensure that it satisfies the integrity constraint. The difference with usual revision scenarios is that the integrity constraint is not supposed to be a new piece of information received by that agent. It exists in the belief state of the agent as a fixed knowledge about the world.

First, this work defines the agent's belief state through an extension of Dung's framework, which uses a propositional formula on labellings as an integrity constraint, and considers that the agent's beliefs are the complete labellings of the argumentation framework which satisfy this integrity constraint.

Definition 59 (Labelling-Based Propositional Language)**.**
Given an argumentation framework $F = \langle A, R\rangle$, the language $\mathcal{L}_A^F$ is generated by the following grammar in Backus-Naur Form:

$$\varphi ::= in_{a_i} \mid out_{a_i} \mid u_{a_i} \mid \neg\varphi \mid \varphi \vee \varphi \mid \top \mid \bot$$

where $a_i \in A$. The satisfaction of such a formula by a labelling L, noted $\mid\!\sim_F$, is defined by:

- $L \mid\!\sim_F in_{a_i}$ if and only if $L(a_i) = in$;
- $L \mid\!\sim_F out_{a_i}$ if and only if $L(a_i) = out$;
- $L \mid\!\sim_F u_{a_i}$ if and only if $L(a_i) = undec$;
- $L \mid\!\sim_F \varphi_1 \vee \varphi_2$ if and only if $L \mid\!\sim_F \varphi_1$ or $L \mid\!\sim_F \varphi_2$;
- $L \mid\!\sim_F \neg\varphi$ if and only if $L \not\mid\!\sim \varphi$;
- $L \mid\!\sim_F \top$ and $L \not\mid\!\sim \bot$.

A labelling L is called a model of the formula φ if L satisfies φ. $\mathrm{Mod}_F(\varphi)$ is the set of all the models of φ. For any set of labellings $\mathcal{L}$, $\varphi_{\mathcal{L}}$ is the formula such that $\mathrm{Mod}_F(\varphi_{\mathcal{L}}) = \mathcal{L}$.

In particular, the complete labellings Co_F of the argumentation framework F are exactly the models of the formula:

$$\bigwedge_{a_i \in a} [(in_{a_i} \Leftrightarrow (\bigwedge_{(a_j,i) \in R} out_{a_j}) \wedge (out_{a_j} \Leftrightarrow (\bigvee_{(a_j,a_i) \in R} in_{a_j}))]$$

while the conflict-free labellings Cf_F are the models the formula:

$$\bigwedge_{a_i \in A} (in_{a_i} \Rightarrow ((\bigwedge_{(a_j,a_i) \in R} out_{a_j}) \wedge (\bigwedge_{(a_i,a_j) \in R} out_{a_j})))$$

A formula φ is said to be conflict-free if and only if there is a conflict-free labelling L such that $L \not\mid\!\sim \neg\varphi$.

Definition 60 (Belief State)**.**
A belief state is a pair $S = \langle F, K\rangle$ where $F = \langle A, R\rangle$ is an argumentation framework, and $K \in \mathcal{L}_A^F$ is the agent's integrity constraint. The agent's beliefs $Bel(S)$ is the formula whose models are the complete labellings of F which satisfy the integrity constraint K.

Booth *et al.* propose two approaches to restore the agent's consistency in the case when the agent's beliefs are empty, meaning that there is no complete labelling of F which satisfies K. The first one directly uses the expansion method defined by Baumann and Brewka (see Section 3.2), while the second one uses first belief revision techniques before performing an expansion.

Proposition 22 (Restoring Consistency through Normal Expansion)**.**
Let $S = \langle F, K\rangle$ be an inconsistent belief state with K a conflict-free formula. There exists F' a normal expansion of F such that $S' = \langle F', K\rangle$ is consistent.

This first method to restore consistency through a normal expansion ensures to satisfy the integrity constraint, but there is no relation between the agent's complete labellings before and after the restoration. The second approach uses belief revision techniques to compute what is called the *fallback beliefs*, which are represented by a formula such that the set of its models is the consistent subset of the current's agent beliefs which are the most plausible. Then a framework expansion is performed to match these fallback beliefs. The underlying idea is that these beliefs are the "best" outcome of the current argumentation framework, given the integrity constraint.
First, we have to introduce a counterpart to Katsuno and Mendelzon's faithful assignments.

Definition 61 (Faithful Assignment)**.**
A *faithful assignment* is a mapping from each argumentation framework F to a pre-order $\leq_F$ between conflict-free labellings of F such that L is a minimal element among the conflict-free labellings of F with respect to $\leq_F$ if and only if L is a complete labelling of F.

Then, a set of rationality postulates adapted from Katsuno and Mendelzon are stated, with a representation theorem.

In the following postulates, $S = \langle F, K\rangle$ is a belief state, and $Bel^*(S)$ are the *fallback beliefs* of S.

(P1) $Bel^*(S) \models K \wedge \varphi_{Cf_F}$

(P2) If S is consistent, then $Bel^*(S) \equiv Bel(S)$

(P3) If K is conflict-free, then $Bel^*(S)$ is conflict-free

(P4) If $F_1 = F_2$ and $K_1 \equiv K_2$, then $Bel^*(\langle F_1, K_1\rangle) \equiv Bel^*(\langle F_2, K_2\rangle)$

(P5) $Bel^*(S) \wedge \varphi \models Bel^*(\langle F, K \wedge \varphi\rangle$

(P6) If $Bel^*(S) \wedge \varphi$ is conflict-free, then $Bel^*(\langle F, K \wedge \varphi\rangle) \models Bel^*(S) \wedge \varphi$

Proposition 23.
*There exists a faithful assignment mapping each argumentation framework F to a total pre-order $\leq_F$ such that $\mathrm{Mod}_F(Bel^*(\langle F, K\rangle)) = \min(\mathrm{Mod}_F(K) \cap Cf_F, \leq_F)$ if and only if Bel^* satisfies the postulates **(P1)-(P6)**.*

3.6.2 Removed Set-Based Revision

The work presented in [NW14] uses a belief revision framework different from the classical AGM framework. It is based on the work from Falappa and colleagues [FKS02], who suggest to revise a belief base by a set of formulae rather than a single formula. This operation is call *multiple revision*. [NW14] revise an argumentation framework by another one. This is why they consider using an adaptation of multiple revision: an argumentation framework can be interpreted as a set of attacks, which is associated with the set of formulae from Falappa and colleagues.

In a nutshell, the revision approach considers an input framework $F = \langle A, R\rangle$ and a new piece of information represented by the framework $F' = \langle A', R'\rangle$. Each of them are supposed to be consistent, meaning that they admit at least one stable extension. If their union $F \sqcup F' = \langle A \cup A', R \cup R'\rangle$ is also consistent, then it is kept as the result of the revision. It is the counterpart of the equivalence between belief revision and belief expansion when the input belief base and the new piece of information are consistent with each other. In the remaining case, some attacks must be dropped from R to obtain a consistent argumentation famework. This is inspired by removed set revision developed in [Pap92, WJP00].

This revision procedure needs to identify some sets of arguments that attack their complement but are not conflict-free. Each of these sets would be a stable extension of the argumentation framework if it was conflict-free.

Definition 62 (Pseudo-Stable Set)**.**
Given an argumentation framework $F = \langle A, R\rangle$. A set of arguments $S \subseteq A$ is called a *pseudo-stable set* of F if and only if $\forall a_i \in A\backslash S$, $\exists a_j \in S$ such that $(a_j, a_i) \in R$.

To transform such a set of arguments into a stable extension, dropping each attack between arguments from S proves enough. Now, let us identify the sets of attacks which must be dropped to restore consistency in an argumentation framework.

Definition 63 (Potential Removed Set)**.**
Given an inconsistent argumentation framework $F = \langle A, R\rangle$, $R_r \subseteq R$ is a *potential removed set* of F if and only if $\langle A, R\backslash R - r\rangle$ is a consistent argumentation framework.

Example 19.
Let $F = \langle A, R\rangle$ be the argumentation framework given at Figure 3.10. F does not admit a stable extension. Each singleton of R is a potential removed set, since it would lead to a consistent argumentation

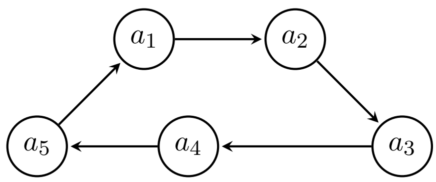


Figure 3.10: The Argumentation Framework F

framework. For instance, removing the set $\{(a_1, a_2)\}$ leads to an argumentation framework which admits the stable extension $\{a_1, a_2, a_4\}$.

Now, each potential removed set of F is not a correct removed set for the revision.

Definition 64 (Removed Set)**.**
Given two consistent argumentation frameworks $F = \langle A, R\rangle$ and $F' = \langle A', R'\rangle$ such that $F \sqcup F'$ is inconsistent, $R_r \subseteq R$ is a *removed set* of $F \sqcup F'$ if and only if:

1. $R_r \cap R' = \emptyset$;

2. R_r is a potential removed set of $F \sqcup F'$;

3. R_r is minimal with respect to cardinality among the subsets of R which satisfy conditions 1 and 2.

In other words, a removed set of $F \sqcup F'$ is a set of n attacks from F which can be dropped to restore consistency in $F \sqcup F'$, such that it is impossible to restore consistency dropping only $m < n$ attacks from F. The set of all removed sets of $F \sqcup F'$ is denoted $\mathcal{R}(F \sqcup F')$. The function f is a selection function, which maps a collection of removed sets to a single one.
Now we can give the removed set-based revision operator:

Definition 65 (Removed Set-Based Revision)**.**
Let $F = \langle A, R\rangle$ and $F' = \langle A', R'\rangle$ be two consistent argumentation frameworks. The *removed set-based revision* of F by F', denoted by $F * F'$, is defined by:

- if $F'' = F \sqcup F'$ is consistent, then $F * F' = F''$;
- else $F * F' = \langle A \cup A', (R \backslash f(\mathcal{R}(F \sqcup F'))) \cup R'\rangle$.

Example 19 Continued.
Let F_1 and F_2 be the argumentation frameworks given at Figure 3.15. $F = F_1 \sqcup F_2$ does not admit a stable extension. As explained previously, any attack can be removed from F to obtain a consistent

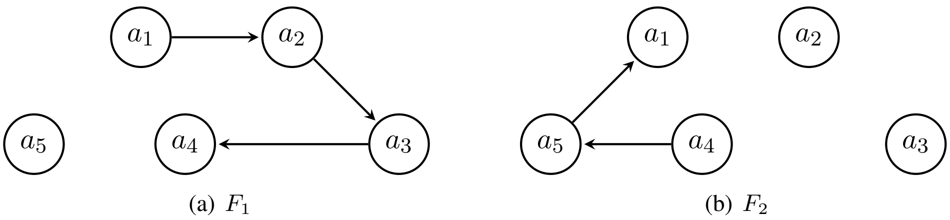


Figure 3.11: The Input Framework F_1 and the Revising Framework F_2

argumentation framework. But the first condition in the definition of a removed set clearly states that neither $\{(a_4, a_5)\}$ nor $\{(a_5, a_1)\}$ can be removed in this particular case, since they belong to F_2. So one the the three other attacks must be removed. Continuing the previous example, we consider that the selection function choses (because of lexicographic order for instance) to remove the set $\{(a_1, a_2)\}$. This leads to the argumentation framework given at Figure 3.12.

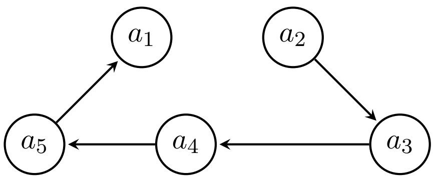


Figure 3.12: The Result of the Removed Set-based Revision

The authors also give an adaption of the rationality postulates for multiple revision, and state which ones are satisfied by the revision operator. Then they propose an algorithm which computes the removed sets.
Let us mention that this kind of revision approach is not generic, since the definition of consistency by the existence of an extension is not relevant for most of the usual semantics, which always lead to at least one extension for each argumentation framework.

3.6.3 Structure-Based AGM Revision

A last contribution from other authors that we want to present in this chapter is a work from Baumann and Brewka [BB15]. This work combines the AGM framework with abstract argumentation. In this work, the new piece of information which explains the revision is an argumentation framework F' such that the result of the revision is an argumentation framework which is a possible consequence of F' in a classical scenario of dialogue; more concretely, this means that the result of the revision is an expansion of F', which has to be as plausible as possible given the original argumentation framework.
To define such revision, the authors define so-called Dung-logic, which is a language composed of the set of every argumentation frameworks which can be built on a given universe of arguments, associated with a consequence relation between argumentation frameworks. We call this set of argumentation frameworks the universe of argumentation frameworks. The consequence relation between frameworks is based on the notion of kernel of an argumentation framework :

Definition 66 (σ-kernel of an Argumentation Framework)**.**
Given an argumentation framework $F = \langle A, R \rangle$, the σ-kernels of F are defined by $F^{k(\sigma)} = \langle A, R^{k(\sigma)} \rangle$, where:

- $R^{k(st)} = R \backslash \{(a_i, a_j) \mid a_i \neq a_j, (a_i, a_i) \in R\}$
- $R^{k(ad)} = R \backslash \{(a_i, a_j) \mid a_i \neq a_j, (a_i, a_i) \in R, \{(a_j, a_i), (a_j, a_j)\} \cap R \neq \emptyset\}$
- $R^{k(gr)} = R \backslash \{(a_i, a_j) \mid a_i \neq a_j, (a_j, a_j) \in R, \{(a_i, a_i), (a_j, a_i)\} \cap R \neq \emptyset\}$
- $R^{k(co)} = R \backslash \{(a_i, a_j) \mid a_i \neq a_j, (a_i, a_i), (a_j, a_j) \in R\}$

F is called *k-r-free* if and only if $F = F^{k(\sigma)}$.

Example 20.
Let us illustrate the notion of stable kernel. Consider the argumentation framework F given at Figure 3.13. To obtain the stable kernel of F, one must remove the attacks such that the attacker is a

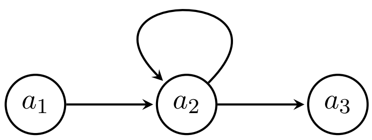


Figure 3.13: The Argumentation Framework F

self-attacking argument. Here, it concerns the attack from a_2 to a_3, since a_2 is self-attacking. So, the stable kernel of F is the argumentation framework given at Figure 3.14.

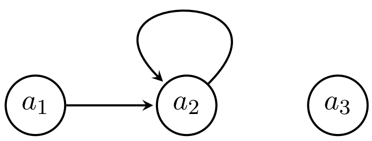


Figure 3.14: $F^{k(st)}$, the Stable Kernel of F

As explained at Example 20, such a kernel is obtained from the argumentation framework F by deleting some redundant attacks. They are used to define k-models of an argumentation framework. Quoting the authors, "*a k-model of F is any dynamic argumentation scenario respecting the information of F modulo redundancy, as encoded by k*". This means that F' is k-model of F when, after the removal of the redundant information in F, the kernel of F' can be obtained by adding arguments and/or (non redundant) attacks.

Definition 67 (k-model)**.**
Given an argumentation framework F and a kernel k, the set of k-models of F is defined as

$$\mathrm{Mod}^k(F) = \{F' \mid F^k \subseteq F'^k\}$$

and given a set of argumentation frameworks $\mathcal{F}$, the set of k-models of $\mathcal{F}$ is defined as

$$\mathrm{Mod}^k(\mathcal{F}) = \bigcap_{F \in \mathcal{F}} \mathrm{Mod}^k(F).$$

Example 20 Continued.
Let F' be the argumentation framework given at Figure 3.15(a). F' is a $k(st)$-model of F, since the stable kernel of F is included in the stable kernel of F', given at Figure 3.15(b).

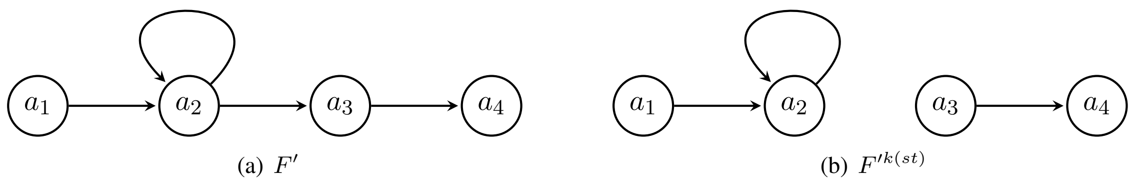


Figure 3.15: The $k(st)$-model F' of F and its Stable Kernel

Similarly to the usual notion of satisfiability (respectively tautology), a set of argumentation frameworks is said to be k-satisfiable (respectively k-tautological) if and only if its set of k-models is empty (respectively equal to the universe of argumentation frameworks). The consequence relation is also defined in a usual way:

Definition 68 (k-consequence Relation)**.**
The *k-consequence relation* $\models^k$ is defined, for each argumentation frameworks F and F', by

$$F \models^k F' \text{ if and only if } \mathrm{Mod}^k(F) \subseteq \mathrm{Mod}^k(F')$$

Then, two argumentation frameworks F and F' are k-equivalent, denoted by $F \equiv^k F'$, if and only if $F \models^k F'$ and $F' \models^k F$.

A Dung-logic $\mathcal{L}^k_{Dung}$ is then a universe of argumentation frameworks associated with the k-consequence relation.

The last concept that we present before describing the AGM-like framework by Baumann and Brewka is the realizability of a set of argumentation frameworks:

Definition 69 (k-Realizability)**.**
A set $\mathcal{M}$ of argumentation frameworks is k-realizable if and only if there exists a set of argumentation frameworks $\mathcal{F}$ such that $\mathrm{Mod}^k(\mathcal{F}) = \mathcal{M}$.

Now, a belief expansion operation for Dung-logics can be defined.

Definition 70 (Belief Expansion for Dung-logics)**.**
A function $+^k$ which maps two argumentation frameworks F and F' to a third one denoted by $F +^k F'$ is a k-*expansion* if and only if

$$\text{Mod}^k(F +^k F') = \text{Mod}^k(F) \cap \text{Mod}^k(F')$$

In the case where $F^k \sqcup F'^k$ is k-r-free, then this is the result of the expansion. Otherwise, the result is the inconsistent constant denoted $\bot$.

This expansion operation is then used, like in the usual AGM framework, in the postulates which charaterize the revision operation.

Definition 71 (Belief Revision for Dung-logics)**.**
A function $\star^k$ which maps two argumentation frameworks F and F' to a third one denoted by $F \star^k F'$ is a k-*revision* if and only if the following postulates are satisfied:

(R1) $F \star^k F'$ is an argumentation framework

(R2) $F \star^k F' \models^k F'$

(R3) $F +^k F' \models^k F \star^k F'$

(R4) if $F +^k F'$ is k-satisfiable, then $F \star^k F' \models^k F +^k F$

(R5) $F \star^k F'$ is k-satisfiable if and only if F' is k-satisfiable

(R6) if $F \equiv^k G$ and $F' \equiv^k G'$, then $F \star^k F' \equiv^k G \star^k G'$

(R7) $(F \star^k F') +^k F'' \models^k F \star^k (F' +^k F'')$

(R8) if $(F \star^k F') +^k F''$ is k-satisfiable, then $F \star^k (F' +^k F'') \models (F \star^k F') +^k F''$

Baumann and Brewka have proven that classical distance-based approaches, such that Dalal's revision, do not satisfy the rationality postulates for Dung-logics. They present another approach, which is based on the kernel. The result of the revision is constructed syntactically. In particular, it can be deduced from the postulates that the result of the revision $F \star^k F'$ must be equal to $F' \sqcup X$, where X is an argumentation framework which maintains as much information as possible from F.
The definition of the operator is based on the notion of maximal k-r-free frameworks.

Definition 72.
Given two argumentation frameworks F and F', the set of *maximal* k-r-*free sets* with respect to F and F' is defined by

$$\mathcal{M}^k_{F-F'} = \max(\{F'^k \sqcup F \mid G \sqsubseteq F^k, F'^k \sqcup G \text{ is } k - r - \text{free}\}, \subseteq)$$

When the considered kernel is the stable one, this set is a singleton. It is then simple to define the $k(st)$-revision operator, and to prove that it satisfies the postulates.

Proposition 24 ($k(st)$-Revision)**.**
Given two argumentation frameworks F and F', the $k(st)$-revision operator is defined by $F \circ^{k(st)} F' = G$, where $\mathcal{M}^{k(st)}_{F-F'} = G$.
*This operator satisfies the postulates for Dung-logics **(R1)**-**(R8)**.*

(a) F_1 (b) F_2

Figure 3.16: The Argumentation Frameworks F_1 and F_2

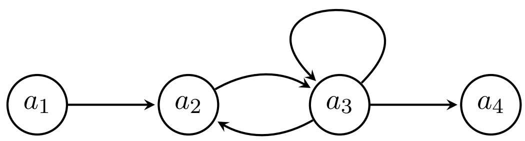


Figure 3.17: The $k(gr)$-expansion $F_1 +^{k(gr)} F_2$

Example 21 (Example of Expansion and Revision, Borrowed from [BB15])**.**
Let us now consider the argumentation frameworks F_1 and F_2 given at Figure 3.16. In the case of $k = k(gr)$, $F_1^k \sqcup F_2^k$ is k-r-free, so the $k(gr)$-expansion $F_1 +^{k(gr)} F_2$ is the argumentation framework given at Figure 3.17. For the other semantics $\sigma \in \{st, co, ad\}$, $F_1^{k(\sigma)} \sqcup F_2^{k(\sigma)}$ is not $k(\sigma)$-r-free, so $F_1 +^{k(\sigma)} F_2 = \bot$. However, it is possible to obtain a meaningful result for $k(st)$-revision, which is present at Figure 3.18. Concretely, we observe that $F_1 \star^{k(str)} F_2$ is a normal expansion of F_2, as close as

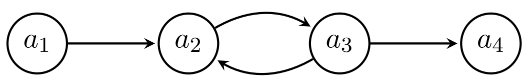


Figure 3.18: The $k(st)$-revision $F_1 \star^{k(str)} F_2$

possible to the initial argumentation framework F_1.

For other kernels, the set of maximal k-r-free set is not necessarily a singleton. But using a selection function to choose a single argumentation framework as the result allow to define revision operators which satisfy the postulates **(R1)**-**(R6)**.

3.7 On Minimal Change of Arguments Statuses and Change of the Attack Relation

When we started the study leading to this thesis, most of the existing works on the dynamics of argumentation framework only considered some simple debate-like scenarios, where the existing arguments and the attacks between them are fixed [BKvdT09b, BKvdT09a, CdSCLS10, BB10, BCdSCLS11, Bau12]. The only kind of change allowed in these work is the addition of a new argument, similarly to what happens in a discussion, when an agent A adds a counter-argument to words of her opponent. Moreover, except for the works on extension enforcement, the existing approaches only study the consequences of the extensions of a given change on the structure of the argumentation framework. Only extension enforcement tackles the question "how to change an argumentation framework to reach a given goal concerning the extensions?".

The only existing work considering some kind of minimal change (enforcement described in [Bau12]) consider that the minimization must deal with the number of attacks which can be added during the change process. Contrastingly, if we consider that the meaning of an argumentation framework is given by the set of arguments that are accepted or rejected with respect to a given semantics, then minimal change means to minimize the change on the statuses of arguments. This option makes sense when the aim of the agent is to decide which arguments are accepted, and to take her decisions with respect to these acceptance statuses. Then, the acceptance statuses of arguments can be associated to the beliefs of the agents, and we make the parallel with the classical AGM framework: when revising an agent's beliefs, the minimal change applies on the models of the formulae, through the definition of rankins on the interpretations. This is why we consider as an interesting reserarch question the study of minimal change on the arguments statuses.
Giving the priority on the minimization of the attack relation or on the minimization of the arguments status is really a question of considering the status of the arguments as first-class citizen of argumentation frameworks or only as a by-product of the graph. Whereas this last option received considerable attention before the begining of this thesis, there has been no work before the ones presented here which concentrates on the first view. The distinction between both kinds of approaches is reminiscent to the duality between coherentist and foundational approaches to belief revision in logical settings [Doy92, dV97].

In addition, previous works also suppose that one can add as many arguments as one wants in order to modify the status of some arguments. In some cases this is perfectly sensible, but in other cases it is difficult to assume that such arguments are available. Consider for instance the case of big society debates, where political parties, economists, journalistes, and other specialists have already put forward all the arguments in favor of or against some decision (for example whether the state has to increase or decrease individual taxation). If a political leader wants to change the current decision, then he will need to be very brilliant in order to find an argument that has not been already pointed out by experts. More probably she will rather try to change the beliefs (or preferences) of the people on the fact that some arguments do or do not attack other ones.

We can motivate this kind of change with the toy examples we presented in Section 1.3. In their debate about the movie screening, John and Yoko use every argument they have to decide which movie they will watch. We explained that their debate may lead to watch Yoko's movie. Since he does not have any new argument to add in the debate, John has to convince Yoko that some of the attacks are not actually present, or that some other ones are missing, to change the outcome of the argumentation framework and obtain a result which satisfies him more.
In the case of John and Paul debating about the place of their next concert, the conclusion of the discussion is that since a night in New-York City is more expensive than a night in Chicago (argument a_4), it is better to play in Chicago (argument a_2), which is Paul's preferred place, than in New-York City (argument a_1). Since John does not know any argument which would allow to change that, he may try to convince that the price of the hotel rooms is not a relevant information to choose a concert place, and so that the attack from a_4 to a_1 must be removed.
Our last example is the resource allocation of toys between the children John, Paul and George. In this case, the only way for Paul to play the board game with the other children is to convince his parents that he is not too young to understand the rules, which would lead to remove the self-attack on the argument (g, P).

Let us now mention some other contexts where allowing only changes of the attack relation is very natural.

A first example of application for the revision process without adding new arguments is the reception of a unjustified but trustworthy information (which is a particular case of argument from authority). This scenario is frequent in applications of argumentation on social network debates [GT13]. When an agent A initiates a debate about an argument α, if another agent B does not agree with A about α but considers that A is trustworthy, B has to revise her beliefs to accept α. In this case, agent B can change her beliefs even if agent A has not introduced a new argument in the debate. So, B has to reconsider the attacks between some arguments, but not the set of arguments itself.

A second example concerns applications of argumentation on public opinion, and is related to the society debates motivation we discussed above: suppose that an argumentation framework represents the opinion of some groups of agents, where an attack between arguments exists if the majority of the group supports it. If a group leader wants to modify the statuses of arguments, then she can perform a revision of the input framework even without the introduction of some new arguments. The resulting argumentation frameworks may help her to determine the attacks she has to focus on so as to modify the majoritarian opinion.

A third context concerns preference-based argumentation (see [AC02a]). In such argumentation frameworks some arguments attack each other (in particular if the arguments are based on logical formulae and rebuttal, attack is symmetric), and the preference relation determines if an attack succeeds or not. So it is possible to modify the attack relation just by modifying the preferences of the agent. A similar case of revision can occur with value-based argumentation frameworks [BC02]: each argument is mapped to a value, and a value can be "stronger" than another. Comparison of values can lead an attack to fail. In this case, a change of values leads to a change of the (succeeding) attacks.

For these reasons, the following approaches for revising an argumentation framework (see Chapter 4, Chapter 5) and enforcing a set of arguments in an argumentation framework (see Chapter 6) are defined as to be able to take account for such situations.
For a matter of generality, we also present some extensions of our change operators which allow to add arguments and to take into account some integrity constraints. These generalized operators can tackle the situations which are described by the previously existing change approaches.
Finally, in Chapter 7 we present a typology of the different kinds of constraint and change that can be considered in the dynamics of argumentation frameworks, and we show how the existing approaches fit in this typology, and which questions are still open challenges.

Part II

Contributions to the Dynamics of Argumentation Frameworks

Chapter 4

Adapting the AGM Framework for Abstract Argumentation

Since we cannot change reality, let us change the eyes which see reality.
Níkos Kazantzákis – *Report to Greco*

In this chapter, we investigate the revision of argumentation frameworks *à la* Dung. We focus on revision as minimal change of the arguments statuses. Contrarily to most of the previous works on the topic, the addition of new arguments is not allowed in the revision process, so that the revised framework has to be obtained by modifying the attack relation, only. We introduce a language of revision formulae which is expressive enough for enabling the representation of complex conditions on the acceptability of arguments in the revised framework. We show how AGM belief revision postulates can be translated to the case of extension-based argumentation semantics. We provide a corresponding representation theorem in terms of minimal change of the arguments statuses. Several distance-based revision operators satisfying the postulates are also pointed out, along with some methods to build revised argumentation frameworks. We also discuss some computational aspects of those methods. Then, we describe several enrichements of our basic approach. The first one is a reformulation of our rationality postulates in terms of labellings, which allow to revise an argumentation framework by some more complex pieces of information. We also discuss the possibility to incorporate some integrity constraints to the revision operator, and to add new arguments during the revision process. Last, we present some approaches to provide a single argumentation framework as the outcome of the revision step.

Contents

4.1 Revision at the Extension Level

4.1.1 On Revision Formulae

We want to define a revision setting for Dung's argumentation frameworks in which sophisticated revision formulae can be taken into account, and not only the fact that a given argument should be accepted or refused. To this end, we consider a logical language $\mathcal{L}_A$, where negation is used to denote the fact that a given argument should be refused, and atomic formulae can be connected using conjunction and disjunction.

Definition 73.
Given a set of arguments $A = \{\alpha_1, \ldots, \alpha_k\}$, $\mathcal{L}_A$ is the language generated by the following context-free grammar in BNF:

$$\begin{array}{lll} arg & ::= & \alpha_1 | \ldots | \alpha_k \\ \Phi & ::= & arg | \neg\Phi | (\Phi \wedge \Phi) | (\Phi \vee \Phi) \end{array}$$

For instance, $\varphi_1 = (a_1 \wedge a_2 \wedge a_3) \vee (a_1 \wedge \neg a_2 \wedge \neg a_3)$ is a formula of $\mathcal{L}_A$ when $\{a_1, a_2, a_3\} \subseteq A$. Such a formula expresses that a_1 must be accepted and a_2 and a_3 must be both accepted or both refused. The epistemic status of such a formula φ from $\mathcal{L}_A$ in an argumentation framework $F \in \mathsf{AFs}_A$ for a given semantics σ is given by:

Definition 74.
Let $\varepsilon \subseteq A$ and $\varphi \in \mathcal{L}_A$. The concept of *satisfaction* of φ by ε, noted $\varepsilon \mid\!\sim \varphi$, is defined inductively as follows:

- If $\varphi = a_i \in A$, then $\varepsilon \mid\!\sim \varphi$ if and only if $a \in \varepsilon$,
- If $\varphi = (\varphi_1 \wedge \varphi_2)$, $\varepsilon \mid\!\sim \varphi$ if and only if $\varepsilon \mid\!\sim \varphi_1$ and $\varepsilon \mid\!\sim \varphi_2$,
- If $\varphi = (\varphi_1 \vee \varphi_2)$, $\varepsilon \mid\!\sim \varphi$ if and only if $\varepsilon \mid\!\sim \varphi_1$ or $\varepsilon \mid\!\sim \varphi_2$,
- If $\varphi = \neg\psi$, $\varepsilon \mid\!\sim \varphi$ if and only if $\varepsilon \not\mid\!\sim \psi$.

Then for any F in AFs_A, and any semantics σ, we say that:

- φ is *accepted* with respect to F, noted $F \mid\!\sim_\sigma \varphi$, if $\varepsilon \mid\!\sim \varphi$ for every $\varepsilon \in Ext_\sigma(F)$,
- φ is *refused* with respect to F, noted $F \mid\!\sim_\sigma \neg\varphi$, if $\varepsilon \mid\!\sim \varphi$ for no $\varepsilon \in Ext_\sigma(F)$,
- φ is *undefined* with respect to F in the remaining case.

Inference $\mid\!\sim_\sigma$ can be extended to the case of a set S of argumentation frameworks by considering $Ext_\sigma(S) = \bigcup_{F \in S} Ext_\sigma(F)$.

The language $\mathcal{L}_A$, used in next sections to define the revision of an argumentation framework, allows to change the status of an information φ to accepted (revise by φ) or rejected (revise by $\neg\varphi$), but not to undetermined. This is normal: Gärdenfors defines the change of status of a belief from accepted or rejected to undetermined as a *contraction*, as explained in Chapter 2 (see Fig. 2.1).

As introduced in Chapter 1, from now on, we call *candidate* any subset ε of A. Continuing the previous example, if $A = \{a_1, a_2, a_3\}$, then φ_1 is satisfied by the candidates from $\{\{a_1\}, \{a_1, a_2, a_3\}\}$. Thus, for the grounded semantics, φ_1 is accepted with respect to $F_1 = \langle A, R_1 \rangle$ with $R_1 = \{(a_2, a_3), (a_3, a_2)\}$ but is refused with respect to $F_2 = \langle A, R_2 \rangle$ with $R_2 = \{(a_1, a_2), (a_2, a_1)\}$.

We define consistency in a classical way:

Definition 75.
Given a formula φ, $\mathcal{A}_\varphi$ denotes the set of candidates satisfying φ. φ is said to be *consistent* if and only if $\mathcal{A}_\varphi \neq \emptyset$.

In the general case, $\mathcal{A}_\varphi$ is not the set of all σ-extensions of an F in AFs_A. Consider for instance, $A = \{a_1, a_2, a_3\}$ and $\varphi_1 = (a_1 \wedge a_2 \wedge a_3) \vee (a_1 \wedge \neg a_2 \wedge \neg a_3)$. $\mathcal{A}_{\varphi_1} = \{\{a_1\}, \{a_1, a_2, a_3\}\}$, and there is no F in AFs_A such that $Ext_\sigma(F) = \mathcal{A}_{\varphi_1}$ for $\sigma =$ grounded, $\sigma =$ preferred or $\sigma =$ stable. It is obvious for the grounded semantics (since an argumentation frameworks admits only one grounded extension) and preferred semantics (since $\mathcal{A}_{\varphi_1}$ does not satisfy the maximality principle: $\{a_1\} \subseteq \{a_1, a_2, a_3\}$). For the stable semantics, we notice that if $\{a_1\}$ is an extension, then a_1 must attack both the other arguments, which prevents $\{a_1, a_2, a_3\}$ to be an extension.

In this case, it is enough to consider two argumentation frameworks to cover the extensions $\{\{a_1\}, \{a_1, a_2, a_3\}\}$ (for instance, in the first argumentation framework a_1 attacks a_2 and a_3, and in the second one the attack relation is empty). Note that in the general case, increasing the number of frameworks is not enough to capture the expected extensions. In order to characterize formulae that can be associated with a set of frameworks and a semantics, a concept of σ-representability can be defined as follows:

Definition 76.
A set $\mathcal{C}$ of candidates is *σ-representable* if and only if there exists a set $\mathcal{S}$ of argumentation frameworks in AFs_A such that $\mathcal{C} = Ext_\sigma(S)$.

From σ-representability, we define a notion of model which takes the semantics into account:

Definition 77.
Given a formula $\varphi \in \mathcal{L}_A$ and a semantics σ, the set of *models* of φ is defined by

$$\mathcal{A}_\varphi^\sigma = \{\varepsilon \in \mathcal{A}_\varphi | \{\varepsilon\} \text{ is } \sigma\text{-representable}\}.$$

A formula $\varphi \in \mathcal{L}_A$ is *σ-representable* if and only if $\mathcal{A}_\varphi^\sigma$ is *σ-representable*.

A form of consistency can be defined to take account for the semantics:

Definition 78.
Given a semantics σ, a formula $\varphi \in \mathcal{L}_A$ is σ-consistent if and only if φ is consistent and σ-representable.

We remark that σ-representability is closed under (non-empty) subsumption: each non-empty subset of a σ-representable set of candidates is σ-representable. When $A = \{a_1, a_2, a_3\}$, $\varphi_1 = (a_1 \wedge a_2 \wedge a_3) \vee (a_1 \wedge \neg a_2 \wedge \neg a_3)$ is σ-representable for $\sigma =$ grounded, $\sigma =$ preferred or $\sigma =$ stable since $\{\{a_1\}, \{a_1, a_2, a_3\}\} = Ext_\sigma(F_3) \cup Ext_\sigma(F_4)$ where $R_3 = \{(a_1, a_2), (a_1, a_3)\}$ and $R_4 = \emptyset$. Contrastingly, $\varphi_2 = \neg a_1 \wedge \neg a_2 \wedge \neg a_3$ is grounded-representable and preferred-representable but not stable-representable. $\varphi_3 = a_1 \wedge \neg a_1$ neither is grounded-representable nor preferred-representable, but is stable-representable (consider F_5 such that $R_5 = \{(a_1, a_2), (a_2, a_3), (a_3, a_1)\}$).

A last point about formulae is the definition of equivalence. Two formulae $\varphi, \psi \in \mathcal{L}_A$ are said to be σ-equivalent, noted $\varphi \equiv_\sigma \psi$, if and only if $\mathcal{A}_\varphi^\sigma = \mathcal{A}_\psi^\sigma$.

4.1.2 Extension-Based Revision Operators

In order to define revision operators, we follow a two-step process. Intuitively, the process first selects from models of φ those as close as possible to the σ-extensions of F. This selection has to ensure the minimal change of arguments statuses, through an adaptation of KM revision operators, suited to classical logic propositional formulae, suited to the formulae the we defined previously. Then, the second step generates the argumentation frameworks such that the union of their σ-extensions precisely coincides with the selected candidates. This method allows to guarantee the minimal change of arguments statuses, and then to apply any generation function to obtain the resulting argumentation framework, both steps being independant from each other.

We define a revision operator on argumentation frameworks as a mapping associating a set of argumentation frameworks with the input argumentation framework and the input revision formula:

Definition 79.
Given any set of arguments A, a *revision operator* on argumentation frameworks $\star$ is a mapping from $\mathsf{AFs}_A \times \mathcal{L}_A$ to 2^{AFs_A}.

Clearly, the result of the revision of an argumentation framework is not a unique argumentation framework in the general case, but a set of argumentation frameworks. The reason is quite simple: there can be several possible results which have exactly the same maximum attractivity. So in this case there is no reason to select just one of them (we will return to this point later on). If this is problematic for a particular application, a selection function can be used as a tie-break rule for ensuring the unicity of the result (just like, for instance, the maxichoice selection function considered in AGM belief revision [Gär88]). We come back on this topic in Section 4.5.

Of course, each mapping from $\mathsf{AFs}_A \times \mathcal{L}_A$ to 2^{AFs_A} is not a reasonable revision operator. For instance, the constant, yet trivial operator defined by $F \star \varphi = \emptyset$ should be discarded.

In order to identify interesting revision operators, we have to identify the logical properties which guarantee a rational behaviour. Such an axiomatic approach is standard in logic, and the AGM postulates [AGM85, KM91] have been pointed out for characterizing valuable revision operators in a logical setting. As in [QLB06], we can revisit these postulates in a set-theoretic style, here suited to the argumentation case.

Let $\mathcal{S}$ be a set of argumentation frameworks F in AFs_A, $Ext_\sigma(\mathcal{S}) = \cup_{F \in \mathcal{S}} Ext_\sigma(F)$. The counterpart of AGM postulates in the argumentation case is given by:

Definition 80.
$\star$ is an extension-based AGM revision operator on argumentation frameworks if and only if $\star$ satisfies the following postulates. For any argumentation framework F, any formulae φ and ψ, and any semantics σ:

(AE1) $Ext_\sigma(F \star \varphi) \subseteq \mathcal{A}_\varphi^\sigma$

(AE2) If $Ext_\sigma(F) \cap \mathcal{A}_\varphi^\sigma \neq \emptyset$, then $Ext_\sigma(F \star \varphi) = Ext_\sigma(F) \cap \mathcal{A}_\varphi^\sigma$

(AE3) If φ is σ-consistent, then $Ext_\sigma(F \star \varphi) \neq \emptyset$

(AE4) If $\varphi \equiv_\sigma \psi$, then $Ext_\sigma(F \star \varphi) = Ext_\sigma(F \star \psi)$

(AE5) $Ext_\sigma(F \star \varphi) \cap \mathcal{A}_\psi^\sigma \subseteq Ext_\sigma(F \star (\varphi \wedge \psi))$

(AE6) If $Ext_\sigma(F \star \varphi) \cap \mathcal{A}^\sigma_\psi \neq \emptyset$, then $Ext_\sigma(F \star (\varphi \wedge \psi)) \subseteq Ext_\sigma(F \star \varphi) \cap \mathcal{A}^\sigma_\psi$

(AE1) states that the σ-extensions of the resulting set of argumentation frameworks must be among the models of φ. **(AE2)** demands that if there are σ-extensions of the input framework satisfying φ, then the resulting σ-extensions must coincide with them. **(AE3)** requires the resulting set of σ-extensions to be non-empty as soon as φ is σ-consistent. **(AE4)** guarantees the irrelevance of syntax: the revision by two formulae must be identical if the formulae are equivalent. The last two postulates **(AE5)** and **(AE6)** express a minimal change principle with respect to the arguments statuses: changes of the statuses of the arguments are expected to be minimal with respect to the input framework. In particular, these postulates give the expected behaviour of the operator when an argumentation framework is revised by a conjunction of formulae.

Interestingly, as in the logical case, we can derive a representation theorem which characterizes exactly the revision operators satisfying the postulates in a constructive way. To this end, we first need to present a counterpart of the notion of faithful assignment [KM91] in the argumentation setting:

Definition 81.
A *faithful assignment* is a mapping associating any argumentation framework $F = \langle A, R \rangle$ (under a semantics σ) with a total pre-order $\leq^\sigma_F$ on the set of candidates such that:

1. if $\varepsilon_1 \in Ext_\sigma(F)$ and $\varepsilon_2 \in Ext_\sigma(F)$, then $\varepsilon_1 \simeq^\sigma_F \varepsilon_2$,
2. if $\varepsilon_1 \in Ext_\sigma(F)$ and $\varepsilon_2 \notin Ext_\sigma(F)$, then $\varepsilon_1 <^\sigma_F \varepsilon_2$.

The representation theorem can then be stated as follows:

Proposition 25 (Representation Theorem)**.**
*Given a semantics σ, a revision operator $\star$ satisfies the rationality postulates **(AE1)** - **(AE6)** if and only if there exists a faithful assignment which maps every framework $F = \langle A, R \rangle$ to a total pre-order $\leq^\sigma_F$ so that*

$$Ext_\sigma(F \star \varphi) = \min(\mathcal{A}^\sigma_\varphi, \leq^\sigma_F).$$

This theorem is important for defining operators satisfying the rationality postulates, as the ones presented in the next section.

4.1.3 Distance-Based Revision

Let us now present some (pseudo-)distance-based revision operators satisfying the rationality postulates **(AE1) - (AE6)**.

Let d be any pseudo-distance on 2^A, for instance, the Hamming distance given by $d_H(\varepsilon_1, \varepsilon_2) = |(\varepsilon_1 \setminus \varepsilon_2) \cup (\varepsilon_2 \setminus \varepsilon_1)|$. Given $\varepsilon \in 2^A$ and $\mathcal{E} \subseteq 2^A$, d can be extended to a "distance" between ε and $\mathcal{E}$, by stating that $d(\varepsilon, \mathcal{E}) = \min_{\varepsilon' \in \mathcal{E}} d(\varepsilon, \varepsilon')$. For any argumentation framework $F \in \mathsf{AFs}_A$, this distance induces a total pre-order between candidates $\varepsilon_1, \varepsilon_2 \in 2^A$ given by

$$\varepsilon_1 \leq^{\sigma,d}_F \varepsilon_2 \text{ iff } d(\varepsilon_1, Ext_\sigma(F)) \leq d(\varepsilon_2, Ext_\sigma(F)).$$

On this ground, revision operators can be defined by:

Definition 82.
Let σ be any given semantics. A *pseudo-distance-based revision operator* $\star^d$ is any revision operator for which there exists a pseudo-distance d on 2^A such that for every F and every φ, we have $Ext_\sigma(F \star^d \varphi) = \min(\mathcal{A}^\sigma_\varphi, \leq^{\sigma,d}_F)$.

Proposition 26.
*Let σ be any semantics. Any pseudo-distance-based revision operator $\star^d$ satisfies the rationality postulates **(AE1) - (AE6)**.*

Let us now define another family of pseudo-distance-based operators, which take advantage of labellings. We introduce the notation $Labs^{\sigma}_{\varphi}$ to denote the set of labellings L such that $E(L) \in \mathcal{A}^{\sigma}_{\varphi}$.

Labellings, which bring richer information than extensions, can be used to define interesting pseudo-distance-based revision operators. Consider the following notion of edition pseudo-distance:

Definition 83.
Let m, n, o be three integers and let L_1 and L_2 be two labellings.
An *edition pseudo-distance* $d_{(m,n,o)}$ between labellings is defined as:

$$d_{(m,n,o)}(L_1, L_2) = \sum_{a \in A} ad(L_1(a), L_2(a)),$$

where

- $ad(in, in) = ad(out, out) = ad(undec, undec) = 0$
- $ad(in, out) = ad(out, in) = m$
- $ad(in, undec) = ad(undec, in) = n$
- $ad(out, undec) = ad(undec, out) = o$

Proposition 27.
Let m, n, o be three integers. $d_{(m,n,o)}$ is a pseudo-distance.

Interestingly, these edition pseudo-distances are not necessarily neutral or in/out balanced. We call neutral an edition pseudo-distance such that $ad(in, undec) + ad(undec, out) = ad(in, out)$ and in/out balanced a pseudo-distance such that $ad(in, undec) = ad(undec, out)$. Defining in/out non-balanced edition pseudo-distances is a way for instance to favor acceptance of arguments over rejection (see Example 22).

For any pseudo-distance $d_{\mathcal{L}}$ between labellings, we can define a pre-order $\leq^{\sigma,d_{\mathcal{L}}}_F$ between labellings as we did it for candidates:

$$L_1 \leq^{\sigma,d_{\mathcal{L}}}_F L_2 \text{ iff } d_{\mathcal{L}}(L_1, Labs_{\sigma}(F)) \leq d_{\mathcal{L}}(L_2, Labs_{\sigma}(F)).$$

Definition 84.
Let σ be any given semantics. A *labelling-pseudo-distance-based revision operator* $\star_{d^{\mathcal{L}}}$ is any revision operator for which there exists a pseudo-distance $d^{\mathcal{L}} = d_{(m,n,o)}$ on 2^A such that for every F and every φ, we have $Labs_{\sigma}(F \star_{d^{\mathcal{L}}} \varphi) = \min(Labs^{\sigma}_{\varphi}, \leq^{\sigma,d_{\mathcal{L}}}_F)$.

The following example illustrates the impact of the chosen pseudo-distance on the revised framework:

Example 22.
Let σ be the stable semantics. We revise the framework F_6 below by the formula $\varphi_4 = (\neg d \wedge \neg e)$.

The labellings associated with F_6 are $Labs_{\sigma}(F_6) = \{\{(a, in), (b, out), (c, out), (d, in), (e, in)\}, \{(a, out), (b, in), (c, out), (d, in), (e, in)\}\}$. When we revise F_6 by φ_4 using the pseudo-distance-based operator induced by the pseudo-distance $d_{(1,9,10)}$ on labellings, the obtained result is a framework with

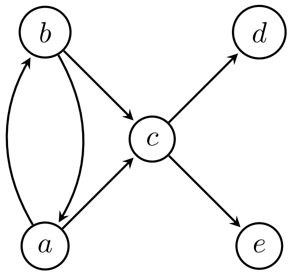


Figure 4.1: The Framework F_6

the following labellings: $\{(a, in), (b, out), (c, out), (d, out), (e, out)\}$ and $\{(a, out), (b, in), (c, out), (d, out), (e, out)\}$. When the pseudo-distance $d_{(9,1,10)}$ is used, we get $\{(a, in), (b, out), (c, out), (d, undec), (e, undec)\}$ and $\{(a, out), (b, in), (c, out), (d, undec), (e, undec)\}$ as labellings of the result frameworks.
If the second step of the process, *i.e.* the generation of the resulting argumentation frameworks, as expected, takes account for the labellings, the structure of the resulting graphs will be different: when the refused arguments are out, it means that there exists an attack from an accepted argument to a refused argument. When the arguments are $undec$, those attacks do not exist.
With the first pseudo-distance $d_{(1,9,10)}$, it is cheaper to change an argument from in to out than to $undec$. Such a pseudo-distance allows for choosing candidates which refuse arguments. Contrarily, the pseudo-distance $d_{(9,1,10)}$ allows for choosing candidates which accept more arguments.

Like operators based on extensions, pseudo-distance-based operators using labellings exhibit good logical properties:

Proposition 28.
Let σ be any semantics. Any labelling-pseudo-distance-based revision operator $\star_{d^{\mathcal{L}}}$ satisfies the rationality postulates ***(AE1) - (AE6)****.*

4.2 Revision at the System Level

The operators defined in the previous sections focus on the candidates that are as close as possible to the extensions of the input framework. This is the actual AGM-like revision of the argumentation framework. However, they do not indicate how to generate the corresponding argumentation frameworks, *i.e.*, the argumentation frameworks such that the union of their extensions coincides with the selected candidates. This task is the second step in the definition of the revision operator. The whole process is schematically described at Figure 4.2. This section presents the generation step. First, we focus on the extension-based generation operators, defining different approaches to generate argumentation frameworks corresponding to the selected candidates. Then, we present some complexity results about these approaches.

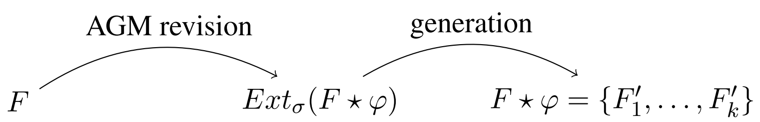


Figure 4.2: Schematic Explanation of the Revision Process

4.2.1 Extension-Based Generation

In order to achieve this task, we consider a mapping $\mathcal{AF}_\sigma$ from 2^{2^A} to 2^{AFs_A}, called *generation operator*, that associates with any set $\mathcal{C}$ of candidates a set of argumentation framework such that $Ext_\sigma(\mathcal{AF}_\sigma(\mathcal{C})) = \mathcal{C}$.

An important point we would like to discuss is the fact that a revision operator $\star$ outputs a set of argumentation frameworks, and not a single argumentation framework in the general case. Actually, this is a consequence of the expressiveness of the language of revision formulae we want to consider. In order to illustrate it, consider $A = \{a, b, c, d\}$, and F_7 as represented in Figure 4.3.

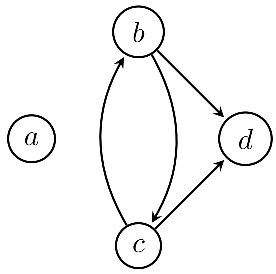
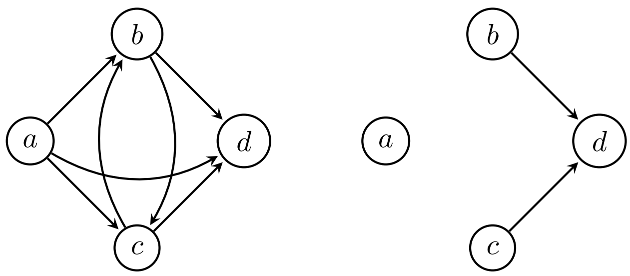


Figure 4.3: The Framework F_7



(a) F_8 (b) F_9

Figure 4.4: Revision of F_7

The extensions of F_7 are the same ones for the stable and preferred semantics, $Ext_\sigma(F_7) = \{\{a, b\}, \{a, c\}\}$. Let $\varphi_5 = (\neg b \vee c) \wedge (\neg c \vee b)$. When computing the result of the revision with the revision operator based on Hamming distance between candidates, we obtain two candidates $\{a\}$ and $\{a, b, c\}$. We present in the following generation operators leading to the two corresponding argumentation frameworks F_8 (corresponding to candidate $\{a\}$) and F_9 (corresponding to candidate $\{a, b, c\}$). Clearly, choosing one of these systems would require to accept some arbitrariness and to lose one of the candidates, although both of them are equaly plausible with respect to notion of minimal change used in this example. And it is obviously impossible to obtain a single argumentation framework which corresponds to these candidates, since $\{a\} \subseteq \{a, b, c\}$.

More generally, we want to recall that obtaining a set as result of a revision process is just usual in most belief revision settings. It is important to note that the canonical representation of AGM contraction/revision operators by use of relational partial-meet functions [AGM85] defines the result of the process as a set of minimal theories. It turns out that the language used makes it possible to produce a single theory from this set using intersection (conjunction). But for languages where this conjunction is not possible it seems natural to keep a set as result. For instance [FKUV86] defines *flocks*, that are the set of logical databases which result from the revision of a single logical database. Flocks have also been used as sets of possible results for combination/merging operators [BKM91, BKMS92, Kon00]. If a particular application clearly needs a single argumentation framework as the result of the revision, a simple possibility is to use a tie-break rule to obtain this result. We come back on this possibility and we present some other ones in Section 4.5

As explained in Section 4.1, the property of σ-representability is closed under non-empty subsumption. Since the candidates $\mathcal{C}$ for the revision are chosen among the models A_{φ}^{σ} of the revision formula φ, it is guaranteed that $\mathcal{C}$ is σ-representable, and so, whatever the semantics, the input argumentation framework and the revision formula, it is possible to generate a set of argumentation frameworks corresponding to the candidates.

So now we can define revision operators:

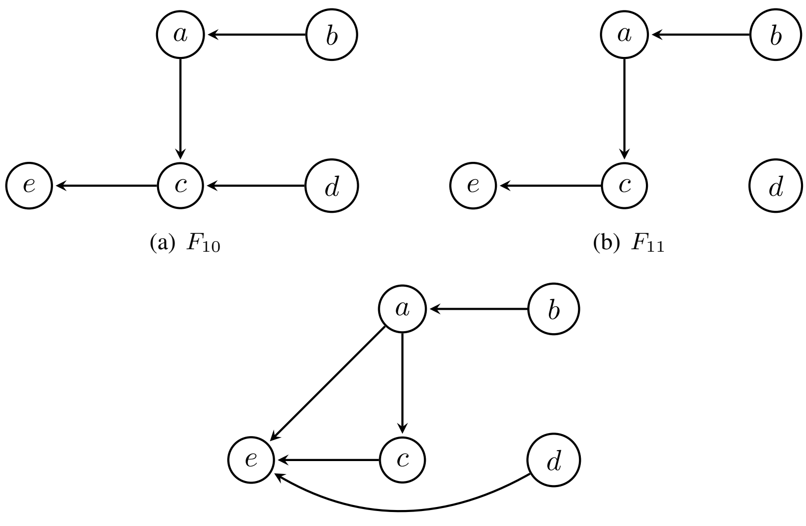
Definition 85.
Given a semantics σ, a faithful assignment that matches every argumentation system to a total pre-order $\leq_F^{\sigma}$, and a generation operator $\mathcal{F}_{\sigma}$, the corresponding revision operator $\star$ is defined by:

$$F \star \varphi = \mathcal{AF}_{\sigma}(\min(\mathcal{A}_{\varphi}^{\sigma}, \leq_F^{\sigma})).$$

One of the key results of the chapter is that:

Proposition 29.
*Every revision operator $\star$ defined following Definition 85 satisfies the postulates **(AE1)-(AE6)**.*

By construction, these revision operators are ensured to deal with minimality of change of arguments statuses, but not with minimality of change of the attack relation. Indeed, the rationality postulates ask for preserving as much as possible the statuses of arguments in the input system: doing so while ensuring that the revision formula is satisfied does not usually imply a minimal change of the attack relation, and vice-versa. As a matter of illustration, consider the argumentation framework F_{10}, F_{11} and F_{12}.



(a) F_{10} (b) F_{11}

(c) F_{12}

Figure 4.5: Minimal Change

Suppose that our goal is to reject e, that is to get a framework so that e does not appear in any extension. So we consider the revision formula $\varphi_6 = \neg e$. A minimal change on the attack relation of F_{10} leads to F_{11}: they differ on a single attack. This contrasts with F_{12} since the change on the attack relation required to go from F_{10} to F_{12} is strictly greater than the change on the attack relation required to go from F_{10} to F_{11}. Each of these three frameworks has a unique extension for the usual semantics:[12] $\{b, d, e\}$ for F_{10}, $\{b, c, d\}$ for F_{11}, and $\{b, d\}$ for F_{12}. Hence, the change on the statuses of arguments achieved when going from F_{10} to F_{12} is strictly smaller than the change on the statuses of arguments achieved when going from F_{10} to F_{11}.

During the generation process, minimization can actually be considered in at least two ways: either minimizing change on the attack relation, or minimizing the number of output systems. In fact, these ways can be combined, either with a more important role to minimal change of the attack relation, or with a more important consideration for minimization of the cardinality of the set of argumentation frameworks which are generated.

Thus, a notion of minimal change on the attack relation can be defined through a notion of pseudo-distance dg on the attack relation. Such a pseudo-distance can be for instance the Hamming distance, given by $dg_H(F_1, F_2) = |(\mathcal{R}_1 \setminus \mathcal{R}_2) \cup (\mathcal{R}_2 \setminus \mathcal{R}_1)|$. The dg_H distance between two argumentation systems corresponds to the number of attacks that must be added or removed to make them identical. But we can also consider more elaborated edition pseudo-distances such as those given in [CMDK+07]. Each pseudo-distance dg induces a pre-order between argumentation systems, defined by $F_1 \leq_F^{dg} F_2$ if and only if $dg(F_1, F) \leq dg(F_2, F)$.
As usual, we can easily extend this notion to a distance between a system F and a set of systems AFs by $dg(F, AFs) = \min_{F_i \in AFs}(dg(F, F_i))$.

In order to give priority to minimal change on the attack relation, we define a generation operator that builds sets of argumentation systems which cover the candidates; then one chooses the ones which minimize a function of the pseudo-distance dg; and finally one retains the sets which are minimal in terms of cardinality.

Definition 86.
Given $\mathcal{C}$ a set of candidates, σ a semantics, dg a pseudo-distance between graphs and F an argumentation system, $\mathcal{AF}_\sigma^{dg,F}$ is defined as:

$$\mathcal{AF}_\sigma^{dg,F}(\mathcal{C}) = \bigcup\{AFs \in sets_\sigma^{dg,F}(\mathcal{C}) \mid card(AFs) \text{ is minimal}\}$$

with

$$sets_\sigma^{dg,F}(\mathcal{C}) = \{AFs \mid Ext_\sigma(AFs) = \mathcal{C} \text{ and } \sum_{F_i \in AFs} dg(F, F_i) \text{ is minimal}\}.$$

A second approach consists in giving priority to the minimality of the output cardinality. It builds first sets of systems that cover the set of candidates with a minimal number of systems, and then chooses the sets which minimize the change on the attack relation.

Definition 87.
Given $\mathcal{C}$ a set of candidates, σ a semantics, dg a pseudo-distance between graphs and F an argumentation system, $\mathcal{AF}_\sigma^{card,F}$ is defined by:

$$\mathcal{AF}_\sigma^{card,F}(\mathcal{C}) = \bigcup\{AFs \in sets_\sigma^{dg,F}(\mathcal{C}) \mid \sum_{F_i \in AFs} dg(F, F_i) \text{ is minimal}\}$$

[12] Especially, for the complete, the preferred, the stable and the grounded semantics.

with

$$sets_\sigma^{dg,F}(\mathcal{C}) = \{AFs \mid Ext_\sigma(AFs) = \mathcal{C} \text{ and } card(AFs) \text{ is minimal}\}.$$

Let us define formally the revision operators corresponding to these generation operators.

Definition 88.
The revision operator $\star_{dg}$ is the mapping from an argumentation framework and a formula to a set of argumentation frameworks such that

$$F \star_{dg} \varphi = \mathcal{AF}_\sigma^{dg,F}(\min(\mathcal{A}_\varphi^\sigma, \leq_F^{\sigma,d_H}))$$

The revision operator $\star_{card}$ is the mapping from an argumentation framework and a formula to a set of argumentation frameworks such that

$$F \star_{card} \varphi = \mathcal{AF}_\sigma^{card,F}(\min(\mathcal{A}_\varphi^\sigma, \leq_F^{\sigma,d_H}))$$

As seen on the following example, these two approaches are not equivalent:

Example 23.
Let us now give an example of revision with the previously defined approaches. The input system F_{13} is given on Figure 4.6.

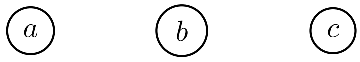


Figure 4.6: The Framework F_{13}

Its unique stable extension is $\{a, b, c\}$. The revision formula is $\varphi_7 = (a \lor b) \land (\neg a \lor \neg b)$, the revision operators are $\star_{dg}$ and $\star_{card}$, both based on the Hamming distance on candidates and the Hamming distance on attack relations. Each one uses one of the previously defined generation operators, the first one gives priority to the minimization of the attack relation, while the second focuses on the minimization on the cardinality of the set of generated argumentation frameworks.
Let us first compute the revised candidates. It is easy to show that $\{a, c\}$ and $\{b, c\}$ are the minimal models of φ_7 with respect to the Hamming distance and the stable extension of the input system.
Now we present the result for the two revision operators. When minimizing the change on attack relation using $\star_{dg}$, the generation step produces two argumentation systems, F_{14} and F_{15}, each one with a single difference from the input graph.

(a) F_{14} (b) F_{15}

Figure 4.7: $F_{13} \star_{dg} \varphi_7$

Contrastingly, the revision of F_{13} with operator $\star_{card}$ gives as output a unique argumentation system F_{16} with two differences with respect to the Hamming distance on the attack relation.

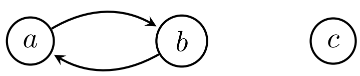


Figure 4.8: $F_{13} \star_{card} \varphi_7$

While the two approaches exemplified here use sum to aggregate the pseudo-distances, any aggregation function can be used instead. For instance, $\min$, $\max$, or any OWA (Ordered Weighted Average [Yag88]). These kinds of aggregation functions allow to combine distance and cardinality without giving priority to one of them. For instance, a specific ordered weighted average OWA_b is given by:

- $v(S) = (dg(F_1, F), \ldots, dg(F_k, F))$, such that $\forall i \in \{1, \ldots, k-1\}, dg(F_i, F) \geq dg(F_{i+1}, F)$, with $S = \{F_1, \ldots, F_k\}$ and k the cardinality of the largest set (the vectors corresponding to smaller sets are normalized by adding the appropriate number of zeroes in front of the vector).

- $w_i = 2^{i-1}$

- $\text{OWA}_b(E) = \sum_{i=1}^{k} w_i v(E)[i]$

With the OWA_b function, a set of frameworks E_1 such that the vector of pseudo-distances is $v_1 = (1, 1, 4)$ is less preferred than a set E_2 with $v_2 = (1, 2, 3)$, because $\text{OWA}_b(E_1) = 19 > \text{OWA}_b(E_2) = 17$.
A set E_2, with three argumentation frameworks, can also be preferred to a set E_3 with only two systems, if those two frameworks are too far from the input framework. For instance, if the vector of pseudo-distances is $(1, 4)$, once normalized to $v_3 = (0, 1, 4)$, $\text{OWA}_b(E_3) = 18$, and so E_2 is still preferred in spite of its larger cardinality.
Giving the priority to one of the kinds of minimality may lead to a result which is good with respect to the first criterion (for instance, a small cardinality of the set of argumentation frameworks) but bad with respect to the second criterion (for instance, with a high distance between the graphs). An aggregation function such as OWA_b allows to have a compromise between the two kinds of minimalities, and to obtain a result which is a more balanced option.

It is worthwhile to note that aggregation functions can also be used alone to define a generation operator. Given an aggregation function f, we define a pre-order $\leq_f$ such that $E_1 \leq_f E_2$ iff $f(E_1) \leq f(E_2)$. For any aggregation function on sets of argumentation systems, a generation operator is AF_σ^f given by

$$\mathcal{AF}_\sigma^f(\mathcal{C}) \in \min(\{E = \{F_i\} | Ext_\sigma(E) = \mathcal{C}\}, \leq_f\}.$$

All proposed approaches cannot guarantee to produce a single set of argumentation systems. For instance, two sets may have the same cardinality and the same pseudo-distance from the input system with respect to the pseudo-distance dg.
The result can then be defined following one of the two options below:

- The result is defined as the union of all the sets. The reason is that these sets represent the uncertain result of the revision, so we keep all of them to avoid an arbitrary choice. The main default of this method is that the size of the result may increase.

- A tie-break rule is used to select a single set of AFs. The agent is obliged to do an arbitrary choice. Note that it does not prevent the revision operator from satisfying rationality postulates (because the postulates deal with extensions, not with attack relations, and all the extensions are conserved when selecting one of the sets of argumentation frameworks).

4.2.2 Some Computational Aspects

A first interesting question concerns the size of the output of revision operators. The first step of the revision can lead to an exponential number of candidates, in terms of the number n of arguments in the input system. This number is directly related to the revision formula. Given a pseudo-distance-based revision operator, the size of the output depends also on the generation operator which is used at the second step of the process. In the worst case, the number of argumentation systems which are generated is exponential in n:

Proposition 30.
Let $\star$ be a revision operator based on the generation operator $\mathcal{AF}_{\sigma}^{dg,F}$. The size of $F \star \varphi$ can be exponential in $|A|$.

The complexity of the inference problem has also to be identified. Given a revision operator $\star$ for argumentation systems and a semantics σ, the *inference problem* from a revised argumentation system is the following decision problem:

- **Input:** An argumentation system F on A, and two formulae $\varphi, \psi \in \mathcal{L}_A$.
- **Query:** Does $F \star \varphi \mid\!\sim_{\sigma} \psi$ hold?

Unsurprisingly, provided that σ ensures the existence of an extension for every F, the inference problem from a revised argumentation system $F \star \varphi$ is at least as hard as the inference problem from F. In formal terms:

Proposition 31.
*Let C be a complexity class which is closed under polynomial-time reductions. Suppose that $\star$ satisfies **(AE1)** to **(AE6)**, and that the semantics σ ensures the existence of an extension for every F. If the inference problem from an argumentation system is C-hard, then the inference problem from a revised argumentation system is C-hard as well.*

Clearly enough, it can be the case that the inference problem from a revised argumentation system $F \star \varphi$ is strictly harder than the inference problem from F (unless $\mathsf{P} = \mathsf{NP}$). For instance, under the restriction when the queries ψ are restricted to arguments (or more generally, CNF formulae on A), it is easy to show that the inference problem from F with respect to the grounded semantics can be solved in polynomial time. Contrastingly:

Proposition 32.
*Suppose that $\star$ satisfies **(AE1)** and **(AE3)**. The inference problem from a revised argumentation system with respect to the grounded semantics is coNP-hard, even under the restriction when the queries ψ are restricted to CNF formulae on A.*

Our results show that the revision of argumentation systems is comparable to the revision of propositional formulae from a computational point of view. Especially, it may lead to harder computational problems: on the one hand, the revision of an argumentation system may require exponentially many systems for being represented (this is reminiscent to the non-compilability of some belief revision operators [CDLS99]); on the other hand, inference may also become harder [Neb98].

4.3 Labelling-Based Revision of Argumentation Frameworks

4.3.1 Labelling-based Postulates

In the Section 4.1.3, we showed that it is possible to use labelling-based distances to define revision operators which satisfy the rationality postulates. Now we refine our postulates to express constraints on the expected labellings of the revised argumentation frameworks rather than the expected extensions. The aim of this new framework is to be able to express more specific revision constraints, such as "the argument a_i must be out" or "the argument a_i must be $undec$", which are stronger conditions than "the argument a_i must not be accepted". Let us formalize this.

Definition 89.
Given a set of arguments $A = \{a_1, \ldots, a_n\}$, a revision formula on labellings is an element of the language $\mathcal{L}_A^{Labs}$ generated by the context-free grammar in BNF:

$$\begin{array}{ll} arg & ::= a_1 | \ldots | a_n \\ \Phi & ::= in(arg) | out(arg) | undec(arg) | \neg\Phi | \Phi \wedge \Phi | \Phi \vee \Phi \end{array}$$

For instance, with $A = \{a_1, a_2, a_3\}$, the formula $\varphi = (\neg in(a_1) \vee out(a_2)) \wedge undec(a_3)$ intuitively means that if a_1 is accepted then a_2 is rejected, and a_3 has to be undecided.
Formally, the satisfaction of such a formula is defined with respect to a labelling, and can be extended to a notion of satisfaction with respect to an argumentation framework and a semantics[13].

Definition 90.
Given $A = \{a_1, \ldots, a_n\}$ a set of arguments, L a labelling on A, and a formula φ, then:

- if $\varphi = X(a)$ with $a \in A$ and $X \in \{in, out, undec\}$, then $L \mid\!\sim \varphi$ if and only if $L(a) = X$,
- if $\varphi = \psi_1 \wedge \psi_2$, then $L \mid\!\sim \varphi$ if and only if $L \mid\!\sim \psi_1$ and $L \mid\!\sim \psi_2$,
- if $\varphi = \psi_1 \vee \psi_2$, then $L \mid\!\sim \varphi$ if and only if $L \mid\!\sim \psi_1$ or $L \mid\!\sim \psi_2$,
- if $\varphi = \neg\psi$, then $L \mid\!\sim \varphi$ if and only if $L \not\mid\!\sim \psi$.

$L_\varphi = \{L | L \mid\!\sim \varphi\}$ is the set of labellings which satisfy the formula φ.
Given an argumentation framework $F = \langle A, R \rangle$ and a semantics σ, $F \mid\!\sim_\sigma \varphi$ if and only if $\forall L \in Labs_\sigma(F), L \mid\!\sim \varphi$.

Similarly to what we have explained for extensions, it is not enough to require a labelling to satisfy a formula. Given a semantics σ, it is important that this labelling satisfies the conditions to be a σ-labelling. For instance, if L is a labelling satisfying a formula φ, and $\exists a \in A$ such that $L(a) = undec$, then L cannot be a stable labelling (a stable labelling can be defined as a complete labelling with an empty set of $undec$ labels), and so L cannot be used as a result of a revision by φ under the stable semantics. This leads to define the counterpart of the σ-representability and σ-consistency in the case of labellings:

Definition 91.

- Given a formula φ, L_φ denotes the set of labellings satisfying φ. φ is said to be *consistent* if and only if $L_\varphi \neq \emptyset$.
- A set $\mathcal{L}$ of labellings is *σ-representable* if and only if there exists a set $\mathcal{S}$ of argumentation frameworks in AFs_A such that $\mathcal{L} = Labs_\sigma(S)$.
- Given a formula $\varphi \in \mathcal{L}_A^{Labs}$ and a semantics σ, the set of *models* of φ is defined by

  $$L_\varphi^\sigma = \{L \in L_\varphi | \{L\} \text{ is } \sigma\text{-representable}\}.$$

- A formula $\varphi \in \mathcal{L}_A^{Labs}$ is *σ-representable* if and only if L_φ^σ is *σ-representable*.
- Given a semantics σ, a formula $\varphi \in \mathcal{L}_A^{Labs}$ is σ-consistent if and only if φ is consistent and σ-representable.

[13] We notice that the language defined here and its semantics are similar to what have been proposed by [BKRvdT13].

Now we can state the set of AGM-like rationality postulates for labelling-based revision. These are the counterparts to the ones introduced in Definition 80, expressing this time some constraints on the expected labellings of the outcome of the revision, rather than constraints on the expected extensions:

Definition 92.
$\star$ is a labelling-based AGM revision operator on argumentation frameworks if and only if $\star$ satisfies the following postulates. For any argumentation framework F, any formulae φ and ψ, any semantics σ:

(AL1) $Labs_\sigma(F \star \varphi) \subseteq L^\sigma_\varphi$

(AL2) If $Labs_\sigma(F) \cap L^\sigma_\varphi \neq \emptyset$, then $Labs_\sigma(F \star \varphi) = Labs_\sigma(F) \cap L^\sigma_\varphi$

(AL3) If φ is σ-consistent, then $Labs_\sigma(F \star \varphi) \neq \emptyset$

(AL4) If $\varphi \equiv \psi$, then $Labs_\sigma(F \star \varphi) = Labs_\sigma(F \star \psi)$

(AL5) $Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi \subseteq Labs_\sigma(F \star \varphi \wedge \psi)$

(AL6) If $Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi \neq \emptyset$, then $Labs_\sigma(F \star \varphi \wedge \psi) \subseteq Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi$

Let us now introduce a counterpart to Katsuno and Mendelzon's faithful assignment for labellings.

Definition 93.
Given a semantics σ, a faithful assignment is a function which maps every argumentation framework $F = \langle A, R\rangle$ to a total pre-order $\leq^\sigma_F$ on the set of σ-representable labellings such that:

- if $L_1 \in Labs_\sigma(F)$ and $L_2 \in Labs_\sigma(F)$, then $L_1 \approx^\sigma_F L_2$;
- if $L_1 \in Labs_\sigma(F)$ and $L_2 \notin Labs_\sigma(F)$, then $L_1 <^\sigma_F L_2$;

This notion is useful to adapt Katsuno and Mendelzon's representation theorem:

Proposition 33.
The revision operator $\star$ satisfies **(AL1)-(AL6)** *if and only if for every semantics σ there exists a faithful assignment which maps every argumentation framework F to a total pre-order $\leq^\sigma_F$ such that for every formula $\varphi \in \mathcal{L}^{Labs}_A$:*

$$Labs_\sigma(F \star \varphi) = \min(L^\sigma_\varphi, \leq^\sigma_F)$$

To define a particular family of operators which satisfy the postulates, let us present how a total pre-order between labellings can be built from a distance, an argumentation framework and a semantics:

Definition 94.
Let d be a pseudo-distance between labellings. Given a labelling L and a set of labellings $Labs$, we define $d(L, Labs) = \min_{L' \in Labs}(d(L, L'))$.
Given an argumentation framework F and a semantics σ, we define the total pre-order $\leq^\sigma_F$ between σ-representable labellings as :

$$L_1 \leq^\sigma_F L_2 \text{ if and only if } d(L_1, Labs_\sigma(F)) \leq d(L_2, Labs_\sigma(F))$$

Such distance-based pre-orders can be defined through the distances presented in Section 4.1.3 or the ones defined in [BCPR12].

Now, let us exhibit a family of revision operators satisfying the postulates, based on the previous definition of pre-orders between labellings:

Proposition 34.
Let d be a pseudo-distance between labellings. The labelling-distance-based revision operator $\star_d$ *such that, for every argumentation framework F, every semantics σ and every formula σ*

$$Labs_\sigma(F \star_d \varphi) = \min(L^\sigma_\varphi, \leq^d_F)$$

*satisfies the rationality postulates **(AL1)-(AL6)**.*

This family of revision operators obviously includes the one defined in Definition 84, but it is a wider family since these new revision operators allow to distinguish between out and $undec$ statuses in the revision formulae. However, we prove that the complete family of labelling-based revision operators, when the revision formulae are limited to the in language, satisfies the rationality postulates **(AE1)-(AE6)**.

We introduce first a useful lemma to prove the following proposition.

Lemma 1.
For each formula $\varphi \in \mathcal{L}_A^{Labs}$ such that φ only contains in variables, there is a formula $\varphi' \in \mathcal{L}_A$ such that $\forall c \in \mathcal{A}^\sigma_{\varphi'}$, $\exists L \in L^\sigma_\varphi$ such that $in(L) = c$; and $\forall L \in L^\sigma_{\varphi'}$, $\exists c \in \mathcal{A}^\sigma_\varphi$ such that $in(L) = c$.
We call φ' the extension-based formula equivalent to φ.

Proposition 35.
*Each operator satisfying **(AL1)-(AL6)**, restricted to formulae built on the in variables, satisfies **(AE1)-(AE6)**.*

We conclude the section on labelling-based revision by a remark on the expressiveness of labellings. We notice that labelling-based revision operators, when they are restricted to the part of the language corresponding to the extensions (the in variables) are rational in the point of view of extension-based revision. They still allow to revise an argumentation framework by a more expressive revision formula, as it is explained in Section 4.1.3, since the underlying (pseudo-)distance takes advantage of the difference between $undec$ and out to define the notion of minimality.

4.3.2 Labelling-Based Generation

To conclude about the generation of argumentation frameworks, we remark that each generation method has its counterpart for the labelling-based revision operators: given a semantics σ, a labelling-based generation operator is a mapping $\mathcal{AF}_\sigma$ from a set of labellings $\mathcal{L}$ to a set of argumentation frameworks $\mathcal{F}$ such that $Labs_\sigma(F) = \mathcal{L}$. Let us define formally the counterparts of the generation operators that we have defined for extension-based generation.

Definition 95.
Given $\mathcal{L}$ a set of labellings, σ a semantics, dg a pseudo-distance between graphs and F an argumentation framework, $\mathcal{AF}^{dg,F}_{\sigma,Labs}$ is defined as:

$$\mathcal{AF}^{dg,F}_{\sigma,Labs}(\mathcal{L}) = \bigcup \{AFs \in sets^{dg,F}_\sigma(\mathcal{L}) \mid card(AFs) \text{ is minimal}\}$$

with

$$sets^{dg,F}_\sigma(\mathcal{L}) = \{AFs \mid Labs_\sigma(AFs) = \mathcal{L} \text{ and } \sum_{F_i \in AFs} dg(F, F_i) \text{ is minimal}\}.$$

Given $\mathcal{L}$ a set of labellings, σ a semantics, dg a pseudo-distance between graphs and F an argumentation framework, $\mathcal{AF}^{card,F}_{\sigma,Labs}$ is defined by:

$$\mathcal{AF}^{card,F}_{\sigma}(\mathcal{L}) = \bigcup\{AFs \in sets^{dg,F}_{\sigma}(\mathcal{L}) \mid \sum_{F_i \in AFs} dg(F, F_i) \text{ is minimal}\}$$

with

$$sets^{dg,F}_{\sigma}(\mathcal{L}) = \{AFs \mid Ext_{\sigma}(AFs) = \mathcal{L} \text{ and } card(AFs) \text{ is minimal}\}.$$

Now, we exhibit two specific labelling pseudo-distance-based revision operators. Both of them are induced by the pseudo-distance $d_{(1,9,10)}$ which was used in the Example 22, and each of them takes advantage of one of the generation operators defined above.

Definition 96.
The revision operator $\star^{Labs}_{dg}$ is the mapping from an argumentation framework and a formula to a set of argumentation frameworks such that

$$F \star^{Labs}_{dg} \varphi = \mathcal{AF}^{dg,F}_{\sigma,Labs}(\min(L^{\sigma}_{\varphi}, \leq^{\sigma,d_{(1,9,10)}}_{F}))$$

The revision operator $\star^{Labs}_{card}$ is the mapping from an argumentation framework and a formula to a set of argumentation frameworks such that

$$F \star^{Labs}_{card} \varphi = \mathcal{AF}^{card,F}_{\sigma,Labs}(\min(L^{\sigma}_{\varphi}, \leq^{\sigma,d_{(1,9,10)}}_{F}))$$

Example 24.
We come back to the argumentation framework F_6 that we described previously. Its complete labellings

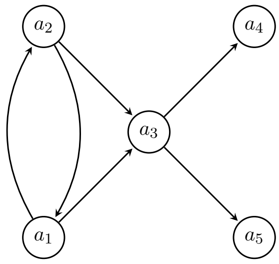
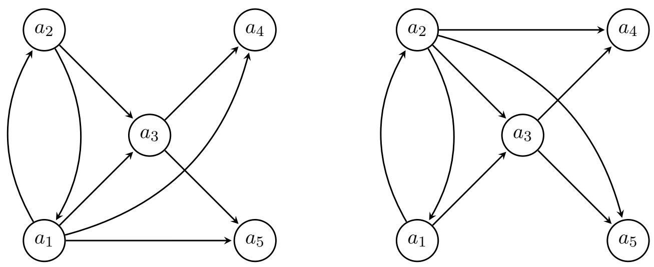


Figure 4.9: The Framework F_6

are $Labs_{co}(F_6) = \{\{(a_1, in), (a_2, out), (a_3, out), (a_4, in), (a_5, in)\}, \{(a_1, out), (a_2, in), (a_3, out), (a_4, in), (a_5, in)\}, \{(a_1, undec), (a_2, undec), (a_3, undec), (a_4, undec), (a_5, undec)\}\}$.
We revise it by $\varphi_8 = \neg in(a_4) \wedge \neg in(a_5) \wedge (\neg undec(a_4) \vee \neg undec(a_5))$, which expresses that a_4 and a_5 must not be accepted in any labelling, and that at least one of them must not be undecided. Using a distance-based revision with the distance $d_{(1,9,10)}$, the candidates selected at the first step of the revision are the labellings $\{(a_1, in), (a_2, out), (a_3, out), (a_4, out), (a_5, out)\}$ and $\{(a_1, out), (a_2, in), (a_3, out), (a_4, out), (a_5, out)\}$. Now, we illustrate the behaviour of two labelling-based generation operators.
$\mathcal{AF}^{dg,F}_{\sigma,Labs}$ minimizes the Hamming distance on graphs first, and then minimizes the number of argumentation frameworks in the outcome as a second criterion. Such a generation operator leads the argumentation frameworks F_{17} and F_{18} as the result of the revision. On the contrary, the generation operator $\mathcal{AF}^{card,F}_{\sigma,Labs}$, which is the labelling-based counter-part of $\mathcal{AF}^{card,F}_{\sigma}$ gives a single argumentation framework F_{19}, which is more distant from the input argumentation framework than F_{17} and F_{18}.



(a) F_{17} (b) F_{18}

Figure 4.10: A Possible Result for $\mathcal{AF}^{dg,F}_{\sigma,Labs}$

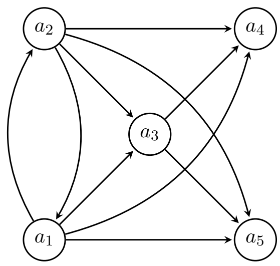


Figure 4.11: F_{19}, a Possible result for $\mathcal{AF}^{card,F}_{\sigma,Labs}$

4.4 Constrained Open World Revision

In the previous section, we presented how to generate argumentation frameworks from the set of candidates selected at the first step of the revision. The generation operators were expected to satisfy the condition that the set of arguments involved in the output argumentation frameworks is identical to the set of arguments of the input framework, and so some changes on the attack relation are required. But in some cases, it makes sense to suppose that some constraints are given about the attack relation: some particular attacks may be a certain information, and so must be preserved during the revision process. This kind of constraint has a very natural side-effect: if it is not possible to change arguments statuses by changing the attacks between them, then new arguments have to be added to perform this status change. More generally, combining integrity constraints, change of attacks and addition of arguments can make sense in some contexts, and it gives us a more general framework for revising argumentation frameworks.

We suppose now the existence of a set of arguments Ω called the *universe of arguments*, and we define an argumentation framework F as a pair $\langle A, R\rangle$ such that $A \subseteq \Omega$. $\mathsf{AFs}_\Omega = \{F = \langle A, R\rangle | A \subseteq \Omega$ and $R \subseteq A \times A\}$ is the set of all the argumentation frameworks built on arguments from Ω.

Definition 97. Structural Constraint on Ω
For every pair of arguments $(a_1, a_2) \in \Omega \times \Omega$,

- $\mathrm{att}(a_1, a_2)$ expresses that a_1 is known to be an attacker of a_2;
- $\neg\,\mathrm{att}(a_1, a_2)$ expresses that a_1 is known not to be an attacker of a_2.

An argumentation framework $F = \langle A, R\rangle$ satisfies a constraint $\mathrm{att}(a_1, a_2)$ (respectively $\neg\,\mathrm{att}(a_1, a_2)$) if and only if $a_1, a_2 \in A$ and $(a_1, a_2) \in R$ (respectively $(a_1, a_2) \notin R$). F satisfies a set of constraints $\mathfrak{C}$ if and only if F satisfies every constraint in $\mathfrak{C}$[14].

[14]Note that partial argumentation frameworks could be used to represent this kind of information [CMDK$^+$07].

The notion of generation operator can then be extended to allow constrained open world revision.

Definition 98. Constrained Open World Generation Operator
A constrained open world (COW) generation operator is a mapping from a semantics σ, a set of candidates $\mathcal{C}$ and a set of structural constraints on Ω $\mathfrak{C}$ to a set of arguments framework $\mathcal{AF}_\sigma(\mathcal{C}, \mathfrak{C})$ such that

- $Ext_\sigma(\mathcal{AF}_\sigma(\mathcal{C}, \mathfrak{C})) = \mathcal{C}$;
- every argumentation framework in $\mathcal{AF}_\sigma(\mathcal{C}, \mathfrak{C})$ satisfies $\mathfrak{C}$.

Now we can define constrained open world revision operators:

Definition 99.
Given a semantics σ, a faithful assignment that matches every argumentation system to a total pre-order $\leq_F^\sigma$, a set of structural constraints $\mathfrak{C}$ and a COW generation operator $\mathcal{AF}_\sigma$, the corresponding constrained open world revision operator $\star_{\mathfrak{C}}$ is defined by:

$$F \star_{\mathfrak{C}} \varphi = \mathcal{AF}_\sigma(\min(\mathcal{A}_\varphi^\sigma, \leq_F^\sigma), \mathfrak{C}).$$

Of course, we can obtain a revision operator as defined in Section 4.2 by defining a constrained open world generation with $\Omega = A$ and $\mathfrak{C} = \emptyset$.
This kind of generation operators may fail to generate a result: the constraint $\mathfrak{C}$ may be too strong, and in particular it may be conflicting with the revision formula φ. For instance, if the constraint is $\mathfrak{C} = att(a_1, a_2)$ and the revision formula is $\varphi_9 = a_1 \wedge a_2$, then it is obvious that the revision operator cannot give a result.

Similarly to the revision operators based on generation operators defined in Section 4.2, the revision operators based on constrained open world generation operators satisfy the postulates **(AE1)**-**(AE6)**, as soon as $\mathfrak{C}$ is not conflicting with the revision formula φ.

Proposition 36.
*Every revision operator $\star$ defined following Definition 99 satisfies the postulates **(AE1)-(AE6)** under the assumption that $\mathfrak{C}$ is not conflicting with the revision formula.*

We explained in the previous section that minimality can be considered in two different ways during the generation process: minimal change on the attack relation, and minimal cardinality. With open world generation, a third kind of minimality can be taken into account: minimal change of the set of arguments. This third kind of minimality can be combined with the first ones, with a more or less important level of priority. For instance, some applications may require to add as few arguments as possible, and consider that minimal change on the attack relation and minimal cardinality are less important, while some other applications may be more permissive on the addition of new arguments but prefer to minimize in priority the change on the attack relation.

4.5 On the Unicity of the Outcome

In the previous sections, we have presented several ways to define argumentation framework revision operators satisfying our adaptations of Katsuno and Mendelzon's postulates. These approaches have the specificity to compute a set of revised argumentation frameworks. We explained in Section 4.2 why it is very natural to keep such a set of argumentation frameworks as the result.

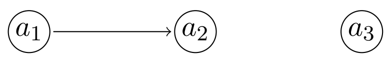
Nevertheless, we are aware of the fact that some applications may require the outcome of the revision to be a single argumentation framework. In general, it is not possible to ensure that a single AF can be computed such that it satisfies the rationality postulates. This implies that some arbirary choice has to be done to select the result of the revision, at the cost of some loss on the properties of this result.

When an agent is obliged to keep a single argumentation framework to represent her knowledge, the first possibility is to use a tie-break rule on the outcome of the revision operator. We can illustrate this simple approach and point out its weakness:

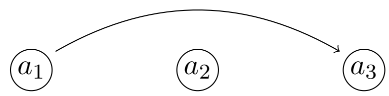
Example 25.
Let F be the argumentation framework given at Figure 4.12, and $\varphi_{10} = (a_1 \vee a_2 \vee \neg a_3) \wedge (a_1 \vee \neg a_2 \vee a_3) \wedge (\neg a_1 \vee a_2 \vee a_3)$. The single stable extension of F is the set $\{a_1, a_2, a_3\}$. We quantify the proximity between candidates with the Hamming distance on sets of arguments, and we quantify the proximity between argumentation frameworks with the Hamming distance between attack relations. The expected canditates when revising F by φ_{10} are $\mathcal{C} = \{\{a_1, a_2\}, \{a_1, a_3\}, \{a_2, a_3\}\}$. The generation

Figure 4.12: The Argumentation Framework F_{20}

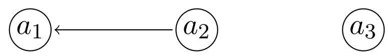
of argumentation frameworks corresponding to $\mathcal{C}$ which minimizes the distance between the resulting argumentation frameworks and F leads to the frameworks F_1, F_2, F_3 given at Figure 4.13. Now, using a



(a) The Argumentation Framework F_{21}



(b) The Argumentation Framework F_{22}



(c) The Argumentation Framework F_{23}

Figure 4.13: Result of the Revision $F_{20} \star \varphi_{10}$

tie-break rule for obtaining a single argumentation framework to represent the agent's kowledge leads to conserving one of these three argumentation frameworks. So, the set of stable extensions of the outcome is a singleton containing one of the candidates from $\mathcal{C}$.

To ensure a unique argumentation framework as the result, it is mandatory to make some compromise on the set of extensions, since it is well-known that, given a semantics, every set of candidates cannot be represented by a single argumentation framework [DDLW14]. As illustrated in the previous example, the risk of applying a tie-break rule on the set of revised argumentation frameworks is to lose almost all the candidates which are expected to be the extensions of the outcome. We present a more subtle method to define a reasonable single outcome for the revision. The idea is to keep a subset $\mathcal{C}'$ of the candidates $\mathcal{C}$, such that $\mathcal{C}'$ is σ-realizable and $\mathcal{C}'$ is as close as possible to $\mathcal{C}$, which ensures to lose as few information as possible from the original set $\mathcal{C}$. To formalize such an operator, we define the notion of σ-approximation:

Definition 100.
A σ-approximation γ_σ is a mapping from a set of candidates $\mathcal{C}$ to a set of candidates $\mathcal{C}'$ such that $\mathcal{C}' \subseteq \mathcal{C}$ and $\mathcal{C}'$ is σ-realizable.

Of course, using any σ-approximation does not keep a better result than the naive method of the tie-break rule. But we define a particular σ-approximation which allows to guarantee that the selection of the candidates implies as few loss of information as possible.

Definition 101.
γ_σ^{MAX} is the σ-approximation such that, for each set of candidates $\mathcal{C}$, $\mathcal{C}' = \gamma_\sigma^{MAX}(\mathcal{C})$ satisfies

1. $\mathcal{C}' \subseteq \mathcal{C}$;

2. $\mathcal{C}'$ is σ-realizable;

3. $\forall \mathcal{C}''$ which satisfies both 1. and 2., $|\mathcal{C}'| \geq |\mathcal{C}''|$.

In the worst case, γ_σ^{MAX} selects a singleton, since we know that each singleton is σ-realizable whatever the semantics σ[15]. This method guarantees to select a set of candidates as close as possible to the set $\mathcal{C}$. Of course, if more than one σ-realizable subset of $\mathcal{C}'$ is maximal with respect to the cardinality, a choice has to be done amongst them, but it is more preferable to have some arbitrary choice at this step of the revision operator than to have some arbitrary choice at the final step, as we illustrate on Example 26.

Example 26.
Let us continue Example 25. Now we apply the σ-approximation γ_σ^{MAX} to avoid using a tie-break rule on the set of argumentation frameworks. Three sets of candidates are the possible outcome of the approximation: $\mathcal{C}_1 = \{\{a_1, a_2\}, \{a_1, a_3\}\}$, $\mathcal{C}_2 = \{\{a_1, a_2\}, \{a_2, a_3\}\}$, and $\mathcal{C}_3 = \{\{a_1, a_3\}, \{a_2, a_3\}\}$. Let us suppose that the σ-approximation leads to conserving $\mathcal{C}_1$. Now it is possible to generate a single argumentation framework to represent the agent's knowledge, without losing too much information after the revision, since its stable extensions are $\{a_1, a_2\}, \{a_1, a_3\}$. A possible result is the argumentation framework F_4 given at Figure 4.14.

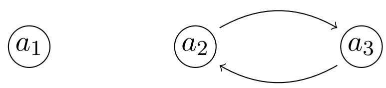


Figure 4.14: The Argumentation Framework F_{24}

Of course, it is possible to define additional criteria to distinguish between the possible results of γ_σ^{MAX}. For instance, the agent may want to keep the beliefs represented by the set of skeptically accepted arguments as close as possible to the set of arguments skeptically accepted by the set of candidates $\mathcal{C}$.

Definition 102.
The distance d_H^{sk} between two sets of candidates $\mathcal{C}, \mathcal{C}'$ is defined by

$$d_H^{sk}(\mathcal{C}, \mathcal{C}') = d_H(\bigcap_{c_i \in \mathcal{C}} c_i, \bigcap_{c'_i \in \mathcal{C}} c'_i)$$

with d_H the Hamming distance between sets of arguments.

$\gamma_\sigma^{MAX,sk}$ is the σ-approximation such that, for each set of candidates $\mathcal{C}$, $\mathcal{C}' = \gamma_\sigma^{MAX}(\mathcal{C})$ satisfies

1. $\mathcal{C}' \subseteq \mathcal{C}$;

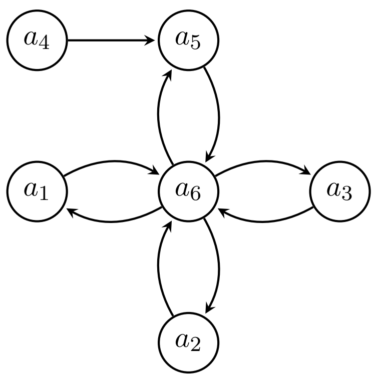
[15]Considering the simple argumentation framework such that each argument in the candidate attacks each other argument proves enough.

2. $\mathcal{C}'$ is σ-realizable;

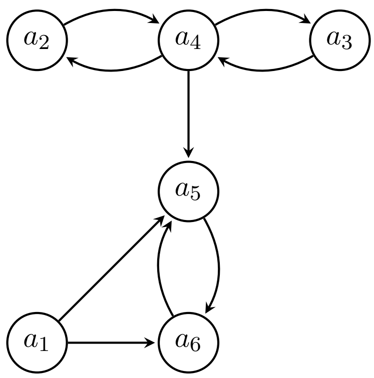
3. $\forall \mathcal{C}''$ which satisfies both 1. and 2., $|\mathcal{C}'| \geq |\mathcal{C}''|$;

4. $\forall \mathcal{C}''$ which satisfies 1., 2. and 3., $d_H^{sk}(\mathcal{C}, \mathcal{C}') \leq d_H^{sk}(\mathcal{C}, \mathcal{C}'')$.

Example 27.
To illustrate the behaviour of $\gamma_\sigma^{MAX,sk}$, we consider the revision of the argumentation framework F_{25} given at Figure 4.15(a). Its complete extensions are $\{a_4\}$, $\{a_4, a_6\}$ and $\{a_1, a_2, a_3, a_4\}$. Revising it with a classical Hamming distance-based revision operator by the formula $\varphi_{11} = (a_1 \wedge (a_1 \vee \neg a_3)) \wedge \neg(a_1 \wedge a_2 \wedge a_3 \wedge a_4) \wedge (\neg a_5 \wedge \neg a_6)$, we obtain the set of candidates $\mathcal{C} = \{c_1, c_2, c_3\}$ with $c_1 = \{a_1, a_2, a_4\}$, $c_2 = \{a_1, a_2, a_3\}$ and $c_3 = \{a_1, a_4\}$. While γ_σ^{MAX} would lead to two possibles sets of candidates $\mathcal{C}' = \{c_1, c_2\}$ and $\mathcal{C}'' = \{c_2, c_3\}$, $\gamma_\sigma^{MAX,sk}$ gives only one possible set of candidates, which is $\mathcal{C}''$. Indeed, $d_H^{sk}(\mathcal{C}, \mathcal{C}') = 1$ and $d_H^{sk}(\mathcal{C}, \mathcal{C}'') = 0$. The result of the revision depends of the generation operator used to associated an argumentation framework with $\mathcal{C}''$. A possible one is F_{26} given at Figure 4.15(b).



(a) F_{25}



(b) F_{26}

Figure 4.15: The Argumentation Framework F and a Possible Revised Framework

Of course, such a σ-approximation can be defined based on any underlying distance.

4.6 Related Work

Now, let us briefly present the work from [DHL+15]. This is a follow-up of our own work [CMKMM14b], presented in this chapter. We essentially point out the difference with our contribution.

Similarly to ourselves, Diller and colleagues consider extensions of an argumentation as the "models" of the argumentation framework, and they adapt the KM belief revision, but they expect that the output of the revision operator is a single argumentation framework. The idea is similar to the work from [DP11]: in this one, it is expected that the revision of a Horn propositional formula gives another Horn propositional formula. For Diller *et al.*, the aim is to ensure that the revision of the σ-extensions of an argumentation framework (which are by definition σ-realizable) gives a new set of σ-realizable extensions. To achieve this goal, they add a condition to be satisfied by the pre-order which is associated with each argumentation framework.

Definition 103 (σ-compliant Pre-Order)**.**
A pre-order $\leq$ is σ-compliant if and only if for each consistent formula φ, $\min(\mathcal{A}_\varphi, \leq) \in \Sigma_\sigma$.

A second kind of revision operators is defined in this paper. This time, the new piece of information which justifies the revision is an argumentation framework. But contrary to the works presented at Section 3.6.2 and Section 3.6.3, the piece of information which has to be incorporated in the outcome of the revision is not the structure of the argumentation framework, but its set of extensions: the revision of F by F' is supposed to be an argumentation framework such that its σ-extensions are included in the σ-extensions of F', and which are the most plausible ones with respect to a given pre-order associated with F.

For both approaches, the authors have proven the existence of a revision operator which satisfies an adaptation of the KM rationality postulates. Any faithful assignment-based revision operator proves enough in the case of revision of an argumentation framework by another one. In the case of revision by a formula, the condition of σ-compliance of the pre-order is required. A possible pre-order which satisfies both conditions (being faithful and σ-compliant) is $\leq_F$ such that its minimal elements are the σ-extensions of F, and the other candidates are sorted in a strict order.

4.7 Conclusion

In this chapter, we investigated the revision problem for abstract argumentation systems *à la* Dung. We focused on revision as minimal change of the arguments statuses. We introduced a language of revision formulae which is expressive enough for enabling the representation of complex conditions on the acceptability of arguments in the revised system. We showed how AGM belief revision postulates can be translated to the case of argumentation systems. We provided a corresponding representation theorem in terms of minimal change of the arguments statuses, and pointed out several pseudo-distance-based revision operators satisfying the postulates. We investigated some computational aspects of revision of argumentation systems.

We are currently encoding our revision operators by representing argumentation systems with logical constraints (in a similar way to [BD04]), so as to be able to benefit from the power of constraint solvers to compute revised systems. At the time of writing this chapter, the revision of the stable extensions of an AF by a logical constraint is encoded. Some future work is to define an encoding for other semantics, and to encode some generation operators. Information about this work are available here: http://www.cril.fr/DynArgs/revision.html.

Here are a couple of open issues.

First, we have explained that the constrained open world revision operators may fail to give a result when the constraint is too strong. This may lead to the violation of some of the rationality postulates. We want to identify the postulates which would be violated, and to give a better characterization of constrained open world revision.

We explained in this chapter that our revision approach does not allow to change an argumentation framework in order to ensure that the status of a given argument is undetermined. This kind of change is related to belief contraction. So, we want to check if the adaptation of belief contraction in propositional logic [CKM15] can be done in a similar way to the adaptation of belief revision which is done in this chapter.
Similarly, it is well-known that there exists a connection between belief merging and belief revision in propositional logic [KP99]. Since the aggregation of argumentation framework has been a dynamic topic

recently (see [DKV15] for an overview of the existing approaches), we want to investigate an adaptation of propositional belief merging for abstract argumentation.

Associating a minimal set of argumentation frameworks with a set of candidates is another important issue, not only for our revision purpose. It is related to the problem of realizability [DDLW14], where the question is to find a (unique) argumentation framework that corresponds to a set of candidates. This problem can also be studied in the case of labellings, and used for the generation of argumentation systems from a set of labellings, exploiting labelling-distance-based revision operators defined in this chapter.

Chapter 5

AGM Revision as a Tool to Revise Argumentation Frameworks

Without translation, I would be limited to the borders of my own country. The translator is my most important ally. He introduces me to the world.

Italo Calvino – *The New-York Times*

The previous chapter describes an approach to revise argumentation frameworks which adapts the AGM framework to abstract argumentation. In this chapter, we investigate the use of AGM revision operators as an underlying tool to revise argumentation frameworks.

Basically, given a semantics σ, we associate with an argumentation framework F a propositional formula $f_\sigma(F)$ which represents it; given the revision formula φ, we take advantage of usual belief revision operators $\circ$ in order to define the revision $F \star \varphi$ of F by φ. In a nutshell, the approach consists in revising using $\circ$ the representation of $f_\sigma(F)$ by a propositional formula induced by φ plus some additional constraints on the expected revision. The output is a propositional formula which characterizes the argumentation frameworks which can be interpreted as the revision of F by φ. This chapter only presents propositional encodings for Dung's complete and stable semantics, but our revision method can be used with any other acceptability semantics σ, as soon as there is a propositional encoding for arguments acceptance given σ.
We present some rationality postulates for the $\star$ operator, which are adapted from KM postulates, but are different from the ones presented in the previous chapter. Then we show that if the revision formulae are restricted to formulae about acceptance statuses, some $\star$ operators satisfy these postulates provided that the corresponding $\circ$ operator satisfies the KM postulates.
We conclude this chapter with a presentation of the connection with some other approaches concerning change of argumentation frameworks.

Contents

5.1 A Translation-Based Approach

In this section, we explain how to encode an argumentation framework into logical constraints, and which constraints must be added to take into account the main acceptability semantics. Then we show that classical belief revision operators can be used to revise an argumentation framework. This idea is reminiscent to the ones considered in [GRR98, CL12] for other purposes (revising modal or non-classical formulae, and case-based reasoning).

First, let us introduce the propositional language which is used, and how we translate an argumentation framework F and an acceptability semantics σ into propositional formulae. Then, we explain how KM revision operators can be used to revise the propositional encoding corresponding to F and σ.

5.1.1 A Propositional Encoding

Let us consider a finite set of arguments $A = \{a_1, \dots, a_n\}$ and an argumentation framework $F = \langle A, R \rangle$.

Definition 104 (Propositional language based on A)**.**

- for $a_i \in A$, acc_{a_i} is a propositional variable meaning "the argument a_i is skeptically accepted by the framework F".
- for $a_i, a_j \in A$, att_{a_i,a_j} is a propositional variable meaning "the argument a_i attacks the argument a_j in the framework F".
- for $a_i \in A$, x_{a_i} is a propositional variable meaning "the argument a_i belongs to the extension of the framework F which is taken in consideration".
- $Prop_A = \{acc_{a_i} \mid a_i \in A\} \cup \{att_{a_i,a_j} \mid a_i, a_j \in A\}$
- $\mathcal{L}_A$ is the propositional language built up from the set of variables $Prop_A$ and the connectives $\neg, \vee, \wedge$.
- $\mathcal{L}_A^+$ is the propositional languange built up from the the of variables $Prop_A \cup \{x_{a_i} \mid a_i \in A\}$ and the connectives $\neg, \vee, \wedge$.

The x_{a_i} variables are only introduced in a technical matter, they disappear in the final version of the encoding, and they cannot be used in the revision formulae, this is why they are not included in the vocabulary of $\mathcal{L}_A$. For this reason, and for a matter of readability, we will write a_i instead of x_{a_i} in the rest of this chapter.
An att-formula (respetively an acc-formula) is a formula from $\mathcal{L}_A$ which contains only variables from $\{att_{a_i,a_j} \mid a_i, a_j \in A\}$ (respectively $\{acc_{a_i} \mid a_j \in A\}$). The language composed of these formulae is denoted by $\mathcal{L}_A^{att}$ (respectively $\mathcal{L}_A^{acc}$).

Clearly enough, the set of models over $\{att_{a_i,a_j} \mid a_i, a_j \in A\}$ of an att-formula φ_{att} (called att-models) corresponds in a bijective way to a set of argumentation frameworks over A: (a_i, a_j) belongs to the attack relation R precisely when att_{a_i,a_j} is true in the model under consideration. It can be formalized through the definition of a mapping from a set of att litterals to an argumentation framework:

Definition 105 (Argumentation Framework Associated with a att-Model)**.**
Given a set A of arguments, any $m \subseteq \{att_{a_i,a_j} \mid a_i, a_j \in A\}$ can be associated with an argumentation framework $arg(m) = \langle A, \{(a_i, a_j) \in A \times A \mid att_{a_i,a_j} \in m\} \rangle$. This notion can be extended to the set of argumentation frameworks corresponding to a set of att-models: $arg(M) = \{arg(m) \mid m \in M\}$.

We also need the following notion of projection:

Definition 106 (att-Projection of Models and Formulae)**.**
Given a set A of arguments, any interpretation m over $\mathcal{L}_A$ can be projected on its att-part: $Proj_{att}(m) = m \cap \{att_{a_i,a_j} \mid a_i, a_j \in A\}$. This notion can be extended to the projection of a formula $\varphi \in \mathcal{L}_A$: $Proj_{att}(\varphi) = \{Proj_{att}(m) \mid m \in \mathrm{Mod}(\varphi)\}$.

Then, a formula φ representing argumentation frameworks can be associated with these frameworks by combining these two mappings: $arg(Proj_{att}(\varphi))$.

The other way around, at a shallow level, any $F = \langle A, R\rangle$ can be represented by the formula over $\{att_{a_i,a_j} \mid a_i, a_j \in A\}$

$$\bigwedge_{(a_i,a_j)\in R} att_{a_i,a_j} \wedge \bigwedge_{(a_i,a_j)\notin R} \neg att_{a_i,a_j}$$

but this translation does not take into account the semantics σ under which F must be interpreted. One clearly needs to consider σ in the encoding. We propose to do it as follows:

Definition 107 (σ-Formula of F)**.**
Given an argumentation framework $F = \langle A, R\rangle$ and a semantics σ, the σ-formula of F is

$$f_\sigma(F) = \bigwedge_{(a_i,a_j)\in R} att_{a_i,a_j} \wedge \bigwedge_{(a_i,a_j)\notin R} \neg att_{a_i,a_j} \wedge th_\sigma(A)$$

where $th_\sigma(A)$ is a logical formula (the σ-theory of A) that encodes the semantics σ.

Now, the question is how to define $th_\sigma(A)$ for usual semantics. To do so, we take advantage of the logical representation of σ-extensions as proposed in [BD04]. Let us begin with the stable semantics. It has been proved in [BD04] that the stable extensions of an argumentation framework $F = \langle\{a_1, \dots, a_n\}, R\rangle$ are exactly the models of the propositional formula:

$$\varepsilon \in Ext_{st}(F) \text{ if and only if } \varepsilon \models \bigwedge_{a_k\in A} (a_k \Leftrightarrow \bigwedge_{a_j:(a_j,a_k)\in R} \neg a_j)$$

It is interesting to note that an argument a_i is skeptically accepted by $F = \langle A, R\rangle$ if and only if every model of the previous formula contains a_i:

$$a_i \in Sc_{st}(F) \text{ if and only if } \forall a_1, \dots, a_n, \models [\bigwedge_{a_k\in A} (a_k \Leftrightarrow \bigwedge_{a_j:(a_j,a_k)\in R} \neg a_j) \Rightarrow a_i]$$

In this encoding, it is assumed that the argumentation framework is known. However, one can relax this assumption by taking advantage of the $att_{x,y}$ variables:

$$acc_{a_i} \Leftrightarrow \forall a_1, \dots, a_n, [\bigwedge_{a_k\in A} (a_k \Leftrightarrow \bigwedge_{a_j\in A} (att_{a_j,a_k} \Rightarrow \neg a_j)) \Rightarrow a_i]$$

This formula encodes a way to compute the skeptically accepted arguments of any argumentation framework built on A under the stable semantics (it proves enough to condition the formula by the literals att_{a_j,a_k} corresponding to the attack relation of the given argumentation framework to recover the encoding from [BD04]).

Altogether, we get:

$$th_{st}(A) = \bigwedge_{a_i\in A} (acc_{a_i} \Leftrightarrow \forall a_1, \dots, a_n, (\bigwedge_{a_k\in A} (a_k \Leftrightarrow \bigwedge_{a_j\in A} (att_{a_j,a_k} \Rightarrow \neg a_j)) \Rightarrow a_i))$$

It is well-known that a quantified Boolean formula (QBF) can be transformed into a classical propositional formula through the elimination of quantifications. We keep the notation of our encoding in QBF to keep a formula with reasonable size, but it does not prevent from using KM revision operators (Section 6.3). This explains why we said previously that the a_i variables disappear: the transformation from the given QBF into an equivalent propositional formula eliminates each occurence of the quantified variables.

Example 28.
Let us illustrate these notions on F_1, given on Fig.5.1.

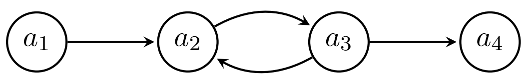


Figure 5.1: The Argumentation Framework F_1

The stable theory of the set of arguments $A = \{a_1, a_2, a_3, a_4\}$ is

$$th_{st}(A) = f_{a_1} \wedge f_{a_2} \wedge f_{a_3} \wedge f_{a_4}$$

with

$$\begin{aligned} f_X = acc_X \Leftrightarrow \quad & \forall a_1, a_2, a_3, a_4, [[(a \Leftrightarrow (att_{a_1,a_1} \Rightarrow \neg a_1) \wedge (att_{a_2,a_1} \Rightarrow \neg a_2)) \\ & \qquad \wedge (att_{a_3,a_1} \Rightarrow \neg a_3)) \wedge (att_{a_4,a_1} \Rightarrow \neg a_4)) \\ & \wedge (a_2 \Leftrightarrow (att_{a_1,a_2} \Rightarrow \neg a_1) \wedge (att_{a_2,a_2} \Rightarrow \neg a_2) \\ & \qquad \wedge (att_{a_3,a_2} \Rightarrow \neg a_3) \wedge (att_{a_4,a_2} \Rightarrow \neg a_4)) \\ & \wedge (a_3 \Leftrightarrow (att_{a_1,a_3} \Rightarrow \neg a_1) \wedge (att_{a_2,a_3} \Rightarrow \neg a_2) \\ & \qquad \wedge (att_{a_3,a_3} \Rightarrow \neg a_3) \wedge (att_{a_4,a_3} \Rightarrow \neg a_4)) \\ & \wedge (a_4 \Leftrightarrow (att_{a_1,a_4} \Rightarrow \neg a_1) \wedge (att_{a_2,a_4} \Rightarrow \neg a_2) \\ & \qquad \wedge (att_{a_3,a_4} \Rightarrow \neg a_3) \wedge (att_{a_4,a_4} \Rightarrow \neg a_4))] \Rightarrow X] \end{aligned}$$

So the stable formula of F_1 is given by

$$th_{st}(A) \wedge (att_{a_1,a_2} \wedge att_{a_2,a_3} \wedge att_{a_3,a_2} \wedge att_{a_3,a_4}) \wedge \bigwedge_{(a_i,a_j) \notin R} \neg att_{a_i,a_j}$$

Propagating the values of att-variables allows to deduce the values of acc-variables ($acc_{a_1} = acc_{a_3} = true$, and $acc_{a_2} = acc_{a_4} = false$), and so leads to the set of skeptically accepted arguments $\{a_1, a_3\}$.

The complete-theory $th_{co}(A)$ of A can be defined in a similar way. First, let us recall the encoding of the complete extensions given in [BD04]:

$$\bigwedge_{a_k \in A} [(a_k \Rightarrow \bigwedge_{a_j:(a_j,a_k) \in R} \neg a_j) \wedge (a_k \Leftrightarrow \bigwedge_{a_j:(a_j,a_k) \in R} (\bigvee_{a_l:(a_l,a_j) \in R} a_l))]$$

Using a similar reasoning scheme, we get that:

$$\begin{aligned} th_{co}(A) = \quad & \textstyle\bigwedge_{a_i \in A} [acc_{a_i} \Leftrightarrow [\forall a_1, \ldots, a_n, \\ & \textstyle\bigwedge_{a_k \in A} [(a_k \Rightarrow \bigwedge_{a_j \in A} (att_{a_j,a_k} \Rightarrow \neg a_j)) \\ & \textstyle\wedge (a_k \Leftrightarrow \bigwedge_{a_j \in A} (att_{a_j,a_k} \Rightarrow \bigvee_{a_l \in A} (att_{a_l,a_j} \Rightarrow a_l)))]] \Rightarrow a_i] \end{aligned}$$

Let us notice that the same scheme can be used to encode any semantics σ as soon as computing a σ-extension is (at most) NP-complete, since it is then possible to associate each argumentation framework with a propositional formula whose models coincide with the σ-extension. More complex semantics, such that the preferred semantics, require either a different encoding (in Quantified Boolean Formulae, for instance), or a propositional encoding whose size is exponential with respect to the size of the argumentation framework.

5.1.2 Encoding Revision Operators with Logical Constraints

One can take advantage of the encodings introduced in the previous section to define revision operators for argumentation frameworks, via the use of classical belief revision operators. In particular, the KM revision operators $\circ$ defined for propositional logic [KM91] are suited to the language $\mathcal{L}_A$.

At a first glance, one can consider to revise $f_\sigma(F)$ by the revision formula φ. However, this is not sufficient. Indeed, if the revision formula φ does not correspond to any argumentation framework interpreted under the semantics σ (for instance, when $\varphi = acc_a \wedge acc_b \wedge att_{a,b}$), then the revised formula will not correspond to any argumentation framework interpreted under σ. Indeed, the success postulate $f_\sigma(F) \circ \varphi \models \varphi$ would force φ to be the case.

Such pathological scenarios must be avoided. A way to ensure it consists in revising $f_\sigma(F)$ by $\varphi \wedge th_\sigma(A)$ since the latter formula is logically consistent precisely when there exists at least one argumentation framework interpreted under σ which is compatible with φ.

Finally, the models of the revised formula $f_\sigma(F) \circ (\varphi \wedge th_\sigma(A))$, projected onto the $att_{x,y}$ variables, characterize the revised argumentation frameworks.

Definition 108 (Translation-Based Revision)**.**
Let $\circ$ be a KM revision operator. For any semantics σ, any argumentation framework $F = \langle A, R \rangle$ and any formula $\varphi \in \mathcal{L}_A$, the associated *translation-based revision operator* $\star$ is given by:

$$F \star \varphi = arg(Proj_{att}(f_\sigma(F) \circ (\varphi \wedge th_\sigma(A))))$$

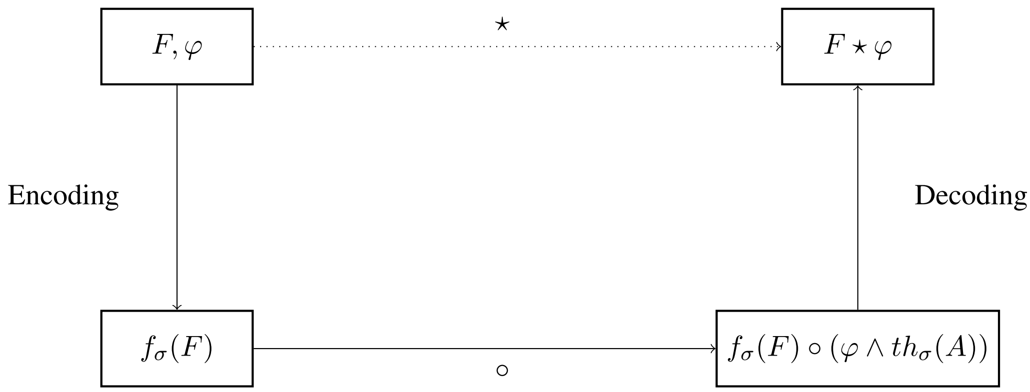


Figure 5.2: Schematic Explanation of the Revision Process

The decoding process is performed by the functions arg and $Proj_{att}$ defined previously (Definition 105, Definition 106).

Let us instantiate this general definition of translation-based revision operators, using distances between interpretations over $\mathcal{L}_A$.

Definition 109 (Distance-based revision)**.**
Let d be a distance between interpretations over $\mathcal{L}_A$, and $\circ_d$ the KM distance-based revision operator defined from d. Then, the *distance-based argumentation framework revision operator* $\star_d$ is defined by

$$F \star_d \varphi = arg(Proj_{att}(f_\sigma(F) \circ_d (\varphi \wedge th_\sigma(A))))$$

Depending on the revision operator $\circ$ used, the concept of minimal change in the argumentation framework can vary. As we have explained in the previous chapter, minimal change in the revision process has at least two meanings in argumentation: minimal change of the arguments statuses or minimal change of the attack relation. Using the Hamming distance between interpretations to define the revision operators (leading to the Dalal revision operator) does not distinguish between both kinds of minimalities, since changing the truth value of an att_{a_i,a_j} variable has the same "cost" than changing the truth value of an acc_{a_i} variable. But we can use other distances to be able to give some priority to one of the minimality criteria.

A first option is to consider minimal change on the arguments statuses more important than minimal change on the attack relation, as it is the case with the revision operators defined in Chapter 4.
To perform this kind of change, we can consider a weighted Dalal-like operator (see [Dal88, KM91]) which ensures minimal change on the acc variables. This kind of revision operator is a specific distance-based revision operator:

Definition 110 (Arguments Statuses Minimal Revision)**.**
Let A be a set of arguments, let $N = |A|^2 + 1$. The acceptance-weighted distance d^{acc} between interpretations is defined by

$$\begin{array}{rcl} d^{acc}(\omega_1, \omega_2) & = & N \times \sum_{a_i \in A}(\omega_1(acc_{a_I}) \oplus \omega_2(acc_{a_i})) \\ & & + \sum_{a_i,a_j \in A}(\omega_1(att_{a_i,a_j}) \oplus \omega_2(att_{a_i,a_j})) \end{array}$$

The *arguments statuses minimal revision operator* $\star_d^{acc}$ is the distance-based revision operator based on the distance d^{acc}.

The weight on acc_{a_i} variables is chosen in such a way that changing the value of every att_{a_i,a_j} variable is still cheaper than changing the value of a single acc_{a_i} variable.

Contrary to the revision approach introduced at Chapter 4, the translation-based revision allow to define a Dalal-like revision operator which requires minimal change on the attack relation. Here the weights are chosen to ensure that changing the value of every acc_{a_i} variable is cheaper than changing the value of a single att_{a_i,a_j} variable:

Definition 111 (Attacks minimal revision)**.**
Let A be a set of arguments, let $N = |A| + 1$. The attacks-weighted distance d^{att} between interpretations is defined by

$$\begin{array}{rl} d^{att}(\omega_1, \omega_2) = & \sum_{a_i \in A}(\omega_1(acc_{a_i}) \oplus \omega_2(acc_{a_i})) \\ & +N \times \sum_{a_i,a_j \in A}(\omega_1(att_{a_i,a_j}) \oplus \omega_2(att_{a_i,a_j})) \end{array}$$

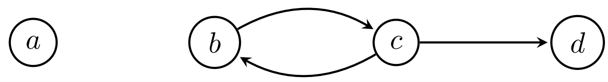
The *attacks minimal revision operator* $\star_d^{att}$ is the distance-based revision operator based on the distance d^{att}.

Let us now illustrate the previously defined revision operators.

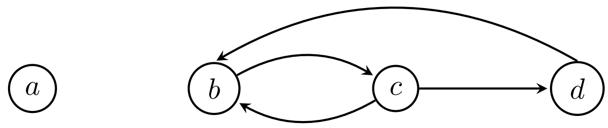
Example 29.
Let us revise the argumentation framework F_1, given on Fig.5.1, by the revision formula $\varphi = acc_a \land \neg att_{a,b}$, meaning that we want to change F_1 to have a skeptically accepted and without a attacking b.

F_1's single stable extension is $\{a, c\}$, so a is already skeptically accepted, but φ is not satisfied because a attacks b. The results of attack minimal revision and argument minimal revision are given respectively on Fig.5.3(a) and Fig.5.3(b). F_2's stable extensions are $\{\{a, c\}\{a, b, d\}\}$, so a is the only skeptically accepted argument. With respect to acceptance statuses, the difference between F_1 and F_2 is 1, and there is also 1 attack different between them ((a, b) is removed).

The single stable extension of F_3 is $\{a, c\}$, so there is no difference between F_1 and F_3 with respect to acceptance statuses. The difference only concerns the attack relation ((a, b) is removed and (d, b) is added).



(a) F_2: Attack Minimal Revision of F_1, $F_1 \star_d^{att} \varphi$



(b) F_3: Arguments Statuses Minimal Revision of F_1, $F_1 \star_d^{acc} \varphi$

Figure 5.3: Results of F_1 Revisions

Such weighted distances also allow to distinguish between the addition and the removal of attacks. For instance, let us imagine that an agent consider that it is easier to add an attack between two (previously unrelated) arguments than to question an existing attack. Then the part of the distance which concerns the attack relation can be adapted to meet the agent's requirement.

Definition 112 (Attacks Removal Distance).
Let A be a set of arguments, let $N' = |A| + 1$ and $N = N' \times |A|^2 + 1$. The attacks removal weighted distance d^{rem} between interpretations is defined by

$$\begin{aligned} d^{rem}(\omega_1, \omega_2) = \; & \sum_{a_i \in A}(\omega_1(acc_{a_i}) \oplus \omega_2(acc_{a_i})) \\ & +N^2 \times \sum_{a_i, a_j \in A}(\omega_1(att_{a_i,a_j}) \land \neg\omega_2(att_{a_i,a_j})) \\ & +N' \times \sum_{a_i, a_j \in A}(\neg\omega_1(att_{a_i,a_j}) \land \omega_2(att_{a_i,a_j})) \end{aligned}$$

Similarly to the weights chosen in the definitions of d^{att} and d^{acc}, the weights used in the definition of d^{rem} are chosen to ensure that it is more expensive for the agent to remove a single attack than to add every possible attack, and it is also more expensive to att a single attack than to change the status of an argument.

We use the word "distance" as a simplification, but d^{rem} is not formally a distance, not even a pseudo-distance, since it is not symmetric. But we can all the same define a pre-order from this "distance" which satisfies faithful assignment properties.

Proposition 37.
For each propositional formula $\varphi \in \mathcal{L}_A$, mapping φ to the total pre-order defined by

$$\omega_1 \leq_\varphi^{rem} \omega_2 \textit{ if and only if } \min_{\omega_3 \in \text{Mod}(\varphi)}(d^{rem}(\omega_1, \omega_3)) \leq \min_{\omega_3 \in \text{Mod}(\varphi)}(d^{rem}(\omega_2, \omega_3))$$

is a faithful assignment.

So we can define a KM revision operator $\circ_d^{rem}$ from this pre-order, and $\circ_d^{rem}$ allows to define a revision operator suited to argumentation frameworks.

Definition 113 (Attacks Removal Minimal Revision).
The *attacks-removal minimal revision operator* $\star_d^{rem}$ is the revision operator based on the KM revision operator $\circ_d^{rem}$ defined from the faithful assignment induced from d^{rem}.

Of course, the counterpart of d^{rem} for attacks addition can also be defined. We illustrate the difference between both approaches.

Example 30.
Let F_4 be the argumentation framework presented at Figure 5.4. The single stable extension of F_4 is $\{a_1\}$. We revise it by $\varphi_2 = acc_{a_2}$. Both F_5 and F_6 presented at Figure 5.5 are possible results, since $Ext_{st}(F_5) = \{\{a_1, a_2\}\}$ and $Ext_{st}(F_6) = \{\{a_2\}\}$. When we consider attacks-removal minimal

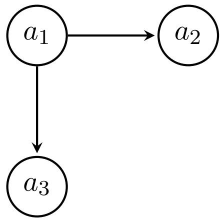
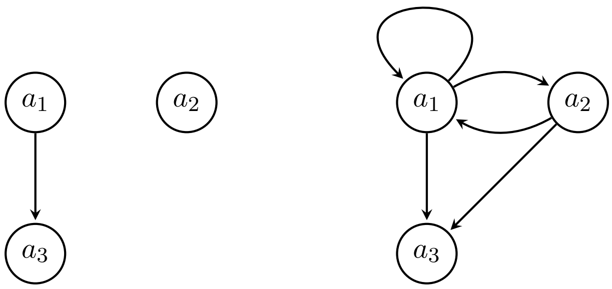


Figure 5.4: The Input Argumentation Framework F_4

revision, the addition of a single attack costs $N' = |A| + 1 = 4$ with respect to the distance d^{rem}, while removing an attack costs $N = N' \times |A|^2 + 1 = 4 \times 9 + 1 = 37$. Then, changing the status of an argument costs 1. So, the change from F_4 to F_5 is more expensive for the agent than the change from F_4 to F_6: removing an attack and changing the status of a_2 costs 38 (for F_5), while adding three attacks and changing the statuses of a_1 and a_2 costs 14 (for F_6). On the opposite, if we consider d^{add} the attacks-addition counterpart of d^{rem} to define the revision operator, the agent prefers to choose F_5 as the outcome of the revision: it costs then 5 to remove an attack and to change the status of a_2 (for F_5) while it costs 113 to add three attacks and to change the statuses of a_1 and a_2 (for F_6)..



(a) F_5 (b) F_6

Figure 5.5: Possible Outputs of the Revision

We have defined revision operators which allow to tackle some dynamic scenarios of argumentation where there is no new argument available, and the attack relation is completely subject to change. But similarly to our first contribution presented in Chapter 4, we can extend the family of revision operators to take into account some other scenarios. First, we define open world revision.

Definition 114 (Open World Revision)**.**
Given $F = \langle A, R \rangle$ an argumentation framework, B a non-empty set of arguments such that $A \cap B = \emptyset$, $\varphi \in \mathcal{L}_{A \cup B}$ a formula and $\circ$ a KM revision operator. The associated *open world revision operator* $\star_B$ is defined by:

$$F \star_B \varphi = arg(Proj_{att}(f_\sigma(F) \circ (\varphi \wedge th_\sigma(A \cup B))))$$

Here, new arguments and new attacks between them or between new and old arguments can be added.

One can also constrain the revision process: some integrity constraints can be required for a particular application (because a given attack is known to hold for sure or because a given argument has to be skeptically accepted, and so cannot change during the revision):

Definition 115 (Constrained revision)**.** Given $F = \langle A, R\rangle$ an argumentation framework, $\varphi, \mu \in \mathcal{L}_A$ formulae and $\circ$ a KM revision operator. The associated *μ-constrained revision operator* is

$$F \star_\mu \varphi = arg(Proj_{att}(f_\sigma(F) \circ (\varphi \wedge th_\sigma(A) \wedge \mu)))$$

Here are some examples of integrity constraints μ which can prove useful:

- $\bigwedge_{a\in A} \neg att_{a,a}$ is useful when self-attacking arguments are not allowed [CMKMM14b];
- $\bigwedge_{(a,b)\in R} att_{a,b} \wedge \bigwedge_{(a,b)\notin R} \neg att_{a,b}$ is useful when attacks between former arguments must be preserved but attacks involving new arguments can be added [CdSCLS10]. In this case, the combination of constrained revision and open-world revision is required to give a result.

Of course, the KM revision operator used to define $\star_B$, or $\star_\mu$ can take advantage of a weighted distance to ensure minimal change of arguments statuses, minimal change of the attack relation or the preference of attacks removal over attacks addition (or vice-versa). A more generalized family of revision operators combines both the use of integrity constraint and the addition of arguments.

Depending on the situation, it can also be useful to consider a single argumentation framework as result of the revision process. This amounts to select one model of the projected formula. Several methods can be used to do so; the simplest one is a tie-break rule which selects any model. We can also use other (more elaborate) criteria to chose the "best" result, like the minization of some distance over the σ-extensions (similarly to the minimal change approach proposed in Chapter 4). Even if we use such elaborate criteria, none of them can guarantee to give a single argumentation framework as the result, and using a tie-break rule may still be required.

5.2 Rationality Postulates in the acc Case

In this section, we focus on constraints expressing an information about skeptically accepted arguments. Let us recall that $Sc_\sigma(F)$ correspond to the *skeptical consequences* of the argumentation framework F with respect to the semantics σ. Formally, it is defined as $\{\bigcap_{\varepsilon\in Ext_\sigma(F)} \varepsilon\}$. We generalize this notion to $Sc_\sigma(S) = \bigcup_{F\in S} Sc_\sigma(F)$ where S is a set of argumentation frameworks. We call this set the skeptical consequences of S.

The satisfaction of acc-formulae can be defined with respect to a set of arguments. Let $\varepsilon \subseteq A$ and φ an acc-formula. The concept of *satisfaction* of φ by ε, noted $\varepsilon \mid\!\sim \varphi$, is defined inductively as follows:

- If $\varphi = acc_a$ with $a \in A$, then $\varepsilon \mid\!\sim \varphi$ if and only if $a \in \varepsilon$,
- If $\varphi = (\varphi_1 \wedge \varphi_2)$, $\varepsilon \mid\!\sim \varphi$ if and only if $\varepsilon \mid\!\sim \varphi_1$ and $\varepsilon \mid\!\sim \varphi_2$,
- If $\varphi = (\varphi_1 \vee \varphi_2)$, $\varepsilon \mid\!\sim \varphi$ if and only if $\varepsilon \mid\!\sim \varphi_1$ or $\varepsilon \mid\!\sim \varphi_2$,
- If $\varphi = \neg\psi$, $\varepsilon \mid\!\sim \varphi$ if and only if $\varepsilon \not\mid\!\sim \psi$.

Then for any argumentation framework F, any set S of argumentation frameworks on A, and any semantics σ, we say that:

- φ is *skeptically accepted* with respect to F, noted $F \mid\!\sim_\sigma \varphi$, if $\forall \varepsilon \in Sc_\sigma(F)$, $\varepsilon \mid\!\sim \varphi$.
- φ is *rejected* with respect to F in the remaining case.
- φ is *skeptically accepted* with respect to S, noted $S \mid\!\sim_\sigma \varphi$, if $\forall \varepsilon \in Sc_\sigma(S)$, $\varepsilon \mid\!\sim \varphi$.
- φ is *rejected* with respect to S in the remaining case.

Each ε in the set $\mathcal{S}(\varphi) = \{\varepsilon \subseteq A \mid \varepsilon \mid\!\sim \varphi\}$ is a possible set of skeptically accepted arguments with respect to a framework which accepts the formula φ. A formula φ is said to be acc-consistent if and only if $\mathcal{S}(\varphi) \neq \emptyset$. Two formulae φ and ψ are said to be acc-equivalent, noted $\varphi \equiv_{acc} \psi$, if and only if $\mathcal{S}(\varphi) = \mathcal{S}(\psi)$.

Let us now point out an adaptation of KM's postulates:

(AS1) $Sc_\sigma(F \star \varphi) \subseteq \mathcal{S}(\varphi)$

(AS2) If $Sc_\sigma(F) \cap \mathcal{S}(\varphi) \neq \emptyset$, then $Sc_\sigma(F \star \varphi) = Sc_\sigma(F) \cap \mathcal{S}(\varphi)$

(AS3) If φ is acc-consistent, then $Sc_\sigma(F \star \varphi) \neq \emptyset$

(AS4) If $\varphi \equiv_{acc} \psi$, then $Sc_\sigma(F \star \varphi) = Sc_\sigma(F \star \psi)$

(AS5) $Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi) \subseteq Sc_\sigma(F \star (\varphi \wedge \psi))$

(AS6) If $Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi) \neq \emptyset$, then $Sc_\sigma(F \star (\varphi \wedge \psi)) \subseteq Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi)$

The first postulate is the *success* postulate: the result of the revision must satisfy the formula φ. **(AS2)** requires the skeptical consequences to stay the same ones if the input framework already satisfies φ. **(AS3)** states that revising a framework by a consistent formula cannot lead to an inconsistent result (such an inconsistent result is identified by an empty set of skeptical consequences). **(AS4)** states that revising by equivalent formulae leads to the same result. The last two postulates constrain the behavior of the revision operator when revising by a conjunction of formulae.

We have proposed similar postulates in Chapter 4. The main difference concerns the semantics of revision formulae. In Chapter 4, argumentation frameworks are revised by propositional formulae the satisfaction of which is defined with respect to the extensions. For instance, $a_i \vee a_j$ means "a_i or a_j must be in every extension" (and so, this formula is satisfied for instance by a framework the extensions of which are $E = \{\{a_i\}, \{a_j\}\}$). Whereas here, formulae deal with the skeptical consequences of the framework, i.e., the intersection of the extensions. So the formula $acc_{a_i} \vee acc_{a_j}$ means "a_i must be in every extension or a_j must be in every extension", and is not satisfied by the set E of extensions.
More generally, the difference between our postulates and those expressed in Chapter 4 is the object of the constraints they give: in Chapter 4, the postulates give some constraints on the expected extensions (or labellings) of the output of the revision process, while the current postulates concern the set of skeptically accepted arguments.

The following proposition explains how to define a rational revision operator from any pseudo-distance between sets of arguments.

Proposition 38.
Given a pseudo-distance d between sets of arguments and an argumentation framework F, $\leq_F^d$ denotes

the total pre-order between sets of arguments defined by: $\varepsilon_1 \leq_F^d \varepsilon_2$ *iff* $d(\varepsilon_1, Sc_\sigma(F)) \leq d(\varepsilon_2, Sc_\sigma(F))$. *The* pseudo-distance based revision operator $\star_d$ *which satisfies*

$$Sc_\sigma(F \star_d \varphi) = \min(\mathcal{S}(\varphi), \leq_F^d)$$

satisfies the postulates ***(AS1) - (AS6)****.*

The previous proposition gives a sufficient condition to prove that a pseudo-distance based revision operator satisfies the rationality postulates. From this proposition, we prove that the arguments statuses minimal revision operator (restricted to the acc-case) satisfies the postulates, through a reduction of this operator to a pseudo-distance based revision operator as described in Proposition 38.

Proposition 39.
The arguments statuses minimal revision operator satisfies the postulates ***(AS1)-(AS6)****.*

Let us illustrate the behaviour of this restricted version of the arguments statuses minimal revision operator.

Example 31.
Let F_7 be the argumentation framework given at Figure 5.6. Its stable extensions are $Ext_{st}(F_7) = \{\{a_1, a_2, a_3\}\}$, so the skeptically accepted arguments are a_1, a_2 and a_3. Let us revise it by the formula $\varphi_3 = \neg acc_{a_1} \vee (\neg acc_{a_2} \wedge \neg acc_{a_3})$. Without taking minimal change into account, the result of the

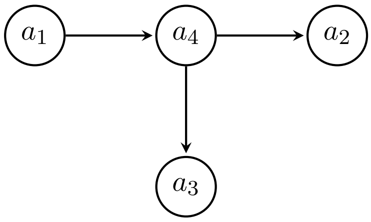


Figure 5.6: The Argumentation Framework F_7

revision could any argumentation framework such that either a_1 is not skeptically accepted, or both a_2 and a_3 are not skeptically accepted. Here, revising with $\star_d^{acc}$, the minimal change principle induced by the distance d^{acc} guarantees that the skeptically accepted arguments of the result must be a_2 and a_3 (since it is minimal to remove only a_1 from the skeptical consequences). Then, minimizing the change of the attack relation, a possible result is the argumentation framework F_8 given at Figure 5.7.

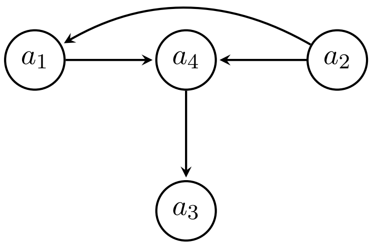


Figure 5.7: The Revised Argumentation Framework $F \star_d^{acc} \varphi$

5.3 Related Work

Among the different kinds of change of argumentation frameworks that we presented in Chapter 3, some of them can be implemented through a revision operator from the family defined here. In particular, the atomic changes studied by Boella and colleagues [BKvdT09b, BKvdT09a] and Cayrol and colleagues [CdSCLS10, BCdSCLS11] are easy to encode in our propositional language. Adding (respectively removing) an attack (a_i, a_j) can be done revising the argumentation framework by the formula $\varphi = att_{a_i,a_j}$ (respectively $\varphi = \neg att_{a_i,a_j}$), while adding an argument a_k can be performed through a constrained open world revision. The constraint

$$\mu = \bigwedge_{(a_i,a_j)\in R} att_{a_i,a_j} \wedge \bigwedge_{(a_i,a_j)\notin R} \neg att_{a_i,a_j}$$

ensures that the former attacks do not change, and the revision formula must be

$$\varphi = \bigwedge_{(a_i,a_j)\in R_{a_k}} att_{a_i,a_j} \wedge \bigwedge_{(a_i,a_j)\notin R_{a_k}, a_i=a_k \text{ or } a_j=a_k} \neg att_{a_i,a_j}$$

where R_{a_k} is the the of attacks which are related to a_k. If priority to recency if expected to be satisfied, then the revision formula is $\varphi' = \varphi \wedge acc_{a_k}$. Since there is not criterion of minimal change in these works, any revision operator can be used.

The same kind of translation of Rienstra's intervention and observation [Rie14] in a revision operation can be done. For instance, an intervention to reject an argument a_i is the addition of a new argument $a_j \notin A$ with an attack from a_j to a_i, which can be easily represented as a constrained open-world revision, with the constraint μ to ensure that the former attacks do not change[16].
Then, the observation that an argument a_i is accepted can also be performed by a constrained open world revision of the framework. The same constraint μ than previously is required, and the revision formula is $\varphi = acc_{a_i}$. The revision operator provides then a result such the the addition of the new argument ensures that a_i is accepted.

The goal-oriented change proposed in [KBM$^+$13] is not exactly a particular case of our revision operators, since the rewriting procedure which has been implemented concerns credulous acceptance. However, the theoretical framework also proposes a skeptical counterpart, with the goal "accept skeptically argument a_i". This is exactly is exactly a revision by formula $\varphi = acc_{a_i}$ with the attacks minimal revision operator $\star_d^{att}$.

It is well-known that when a propositional formula has a single model, revision and update collapse. For this reason, some revision operator described in this chapter may be equivalent to some update operators from the frameworks defined in [BCdSL13, DHP14]. But as soon as we consider the revision or update of a set of argumentation frameworks (which is encoded in a propositional formula which admits several models), both operations differ one from the other. For instance, [DHP14] uses the Forbus update operator to perform the change of an argumentation framework. So the update operator called skeptical enforcement by Doutre *et al.* corresponds in our framework to the revision with the Dalal operator.

[16]In fact, the minimal change of the attack relation can be used to ensure that the former attacks do not change, even if the integrity constraint μ is not used directly.

5.4 Conclusion

In this chapter, we studied a way to benefit from the well-known logical revision operators from Katsuno and Mendelzon's work to define revision operators for abstract argumentation frameworks.
This approach is particularly interesting due to the ability of our revision operators to enforce both *structural* and *acceptability* constraints. Depending on the underlying operator $\circ$, the operator $\star$ ensures minimal change on the acceptance statuses, or on the attack relation. Moreover, these operators can encode some change operators defined in some recent related works.
We have also stated some rationality postulates inspired by the classical AGM framework, and proved that under the assumption that revision formulae only deal with acceptability, any revision operator $\star$ based on an AGM operator $\circ$ satisfies our postulates.

Among the other existing approaches to change an argumentation framework, the ones that we have presented in Section 3.5 are related to the contribution we describe in this chapter. One of the main differences is that these works use *update* operators rather than *revision* operators, which means that they are useful if the change in the argumentation framework comes from a change in the state of the world, while our revision operators tackle the situation of an agent which has to re-evaluate her beliefs about the world, without having an evidence of a change of the world.

As a future work, several possibilities are opened. First, this chapter only presents the logical characterization of skeptical acceptance under the stable and complete semantics. It would be interesting to define a similar characterization of skeptical acceptance under other semantics, this can be done thanks to the encoding method defined in [BD04, EW06, NOC07, AC13, NAD14, BDH14], in particular we can take advantage of the QBF formalism to tackle semantics which have a higher complexity. Another interesting result would be to define the credulous σ-theory for these semantics σ. We are also interested in enforcing the result of the revision to belong to a particular subclass of argumentation frameworks, as the acyclic argumentation frameworks or non-controversial argumentation frameworks.
Another point for further studies is the axiomatic characterization of revision operators. We proved that arguments statuses minimal revision satisfies some rationality postulates in the case of acceptability revision constraints, but it would be interesting to know if some other kinds of operators satisfy these postulates, and to know if some other kinds of revision constraints can be characterized.
We also plan to encode our revision operators into a SAT-based software. The propositional setting of our operators is particularly well-suited to SAT solvers, so this approach is very promising from a computational point of view.

Finally, we think that using logical encodings of abstract argumentation is a powerful way to express other kinds of change operations for argumentation frameworks, and to compute them with some efficient satisfaction or optimization software. The next chapter presents such an approach suited to extension enforcement.

Chapter 6

Extension Enforcement

Each constraint is a gift.

Leonardo da Vinci

As explained in the introduction of this thesis, dynamics of argumentation is a very active topic. The previous chapters describe contributions related to belief change theory. But other kinds of change of an argumentation framework have been studied recently. Especially, the problem of enforcing a set E of arguments, i.e., ensuring that E is an extension (or a subset of an extension) of a given argumentation framework F, has received a particular attention in the recent years. We mentioned it in Section 3.2.

In this chapter, we study the existing approaches to enforce a set of arguments, and point out some of their weaknesses. Then, we define a new family of enforcement operators, for which enforcement can be achieved by adding new arguments (and attacks) to F (as in previous approaches to enforcement), but also by questioning some attacks (and non-attacks) of F. This family includes previous enforcement operators, but also new ones for which the success of the enforcement operation is guaranteed.
We show how the enforcement problem for the operators of the family can be modeled as a pseudo-Boolean optimization problem. Intensive experiments show that the method is practical and that it scales up well.

Contents

6.1 Weaknesses of the Existing Enforcement Approaches

First, we study the enforcement approaches through normal, strong and weak expansion that we presented previously (see Section 3.2 for details). Importantly, whatever the normal enforcement operator under consideration, it must be noted that enforcement may fail. As a simple example, let us consider $E = \{a_1, a_2\}$ in an argumentation framework $F = \langle A, R \rangle$ such that $(a_1, a_2) \in R$. It is obviously impossible to enforce E with any of the enforcement operators described in Section 3.2. Theorem 2 and Theorem 3 from [BB10] give some more elaborate impossibility results about strict enforcement. An interesting result from [BB10] states that for each argumentation framework F, it is possible to enforce any set of arguments E which is conflict-free in F with a non-strict strong enforcement, and it also guarantees that adding a single new argument is enough. It means that non-strict strong enforcement can be performed with any singleton A'. This will be useful to define logical encodings suited to enforcement (see Section 6.3).

Note that the presence of conflicts in the set E of arguments to be enforced is a sufficient, yet unnecessary condition for normal enforcement to fail. In order to make it more formal, let us first introduce the notion of non-trivial set of arguments with respect to a given semantics:

Definition 116.
Let $F = \langle A, R \rangle$ be an argumentation framework, and σ a semantics. $E \subseteq A$ is a σ non-trivial set of arguments in F if and only if E is conflict-free in F and $E \notin Ext_\sigma(F)$.

Assuming the set E of arguments to be enforced to be σ non-trivial is a way to avoid the trivial cases when enforcement is already satisfied because E is a σ-extension of F or impossible because of conflicts. However, it does not prove sufficient for preventing from failure for every semantics. Even if non-strict enforcement is possible, when the agent wants a set of arguments to be *exactly* an extension, enforcement through a normal (or strong, or weak) expansion may be impossible under certain conditions.

Proposition 40.
For every $F = \langle A, R \rangle$ and $E \subseteq A$ a stable non-trivial set in F, there is no strict enforcement of E in F with respect to the stable semantics.

This is a very strong result about the stable semantics, which states that an agent cannot enforce strictly a set of arguments with Baumann and Brewka's approaches, unless if it is already a stable extension of the argumentation framework.

Proposition 41.
For every $F = \langle A, R \rangle$, and $E \subseteq A$ a complete non-trivial set in F,

1. *if E is not admissible, then there is no strict enforcement of E in F with respect to the complete semantics.*
2. *else, if E defends some argument $a_i \in A \backslash E$, then*
   (a) *there is no strict weak enforcement of E in F with respect to the complete semantics.*
   (b) *if odd-length cycles are not allowed, then there is no strict strong enforcement of E in F with respect to the complete semantics.*

There are two different cases which explain that a set of arguments E is not already a complete extension. Either this set is not admissible, and then there is no possible way to enforce strictly E in the argumentation framework. Even if E is admissible, there are some restrictions to the possibility to enforce it.

Proposition 42.
For every $F = \langle A, R\rangle$ and $E \subseteq A$ a grounded non-trivial set in F, if $Ext_{gr}(F) = \{\emptyset\}$, then there is no strict enforcement of E in F with respect to the grounded semantics.

This last proposition states that a set of arguments cannot be enforced strictly with respect to the grounded semantics if there is no unattacked argument in the argumentation framework.

6.2 Argument-Fixed and General Enforcement

Now, we present some new approaches for extension enforcement, which guarantee the success of the operation.

6.2.1 Argument-Fixed Enforcement

In the previous approaches for enforcing a set of arguments, it is supposed that new arguments can be added, and that interactions between the existing arguments do not change. This method is particularly sensible when enforcement is supposed to be the result of a dialog: given an argumentation framework representing the state of a dialog, an agent adds new arguments if she wants to convince the other agent to accept a given set of arguments as an extension. Forbidding any change over the initial attacks of the framework is the reason of the above impossibility results. Interestingly, the converse case, i.e., considering situations where the set of arguments cannot change, but the attack relation is subject to evolutions, also makes sense. It is sensible, for instance, when a set of arguments is observed to be an extension in the output of an argumentation process, but does not correspond to the output of the own argumentation framework of an agent. In such a case, without the knowledge of some new arguments, the agent has to change her beliefs about the attack relation to be consistent with the observed set of arguments.

Definition 117.
Let $F = \langle A, R\rangle$ be an argumentation framework, σ an acceptability semantics, and $E \subseteq A$ a set of arguments. The *argument-fixed (respectively strict argument-fixed) enforcement operator* $+_{\sigma}^{A}$ (respectively $+_{\sigma,s}^{A}$) is defined as a mapping from F and E to an argumentation framework $F' = \langle A, R'\rangle$, with $R' \subseteq A \times A$, and such that E is included in (respectively is exactly) an extension of F'.

The argument-fixed operators guarantee the success of enforcement, even in the strict case:

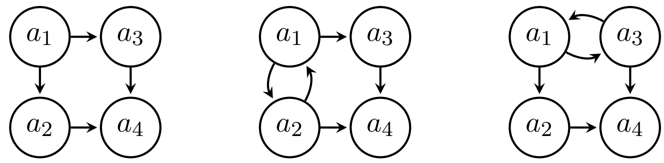
Proposition 43.
Let $F = \langle A, R\rangle$ be an argumentation framework, σ an acceptability semantics and $E \subseteq A$ a set of arguments. There is a strict enforcement F' of E.

Of course, both ideas (adding arguments, and changing the attacks) can be combined:

Definition 118.
Let $F = \langle A, R\rangle$ be an argumentation framework, σ an acceptability semantics, and $E \subseteq A$ a set of arguments. The *general (strict) enforcement operator* $+_{\sigma}$ (respectively $+_{\sigma,s}$) is defined as a mapping from F and E to an argumentation framework $F' = \langle A \cup A', R'\rangle$, with $R' \subseteq (A \cup A') \times (A \cup A')$, and such that E is included in (respectively is exactly) an extension of F'.

Since general enforcement allows any kind of change, it is obvious that it also ensures the success of the operation. Let us exemplify the behaviour of the argument-fixed enforcement operator:

Example 32.
Let us consider the argumentation framework F_1 described in Figure 6.1(a). We give an example of non-strict strong enforcement, but as shown by Proposition 40, it is impossible to perform a strict enforcement under the stable semantics using Baumann's approaches (normal, strong and weak). Let us use the argument-fixed enforcement operator to obtain a strict enforcement of the set of argument $E = \{\{a_2, a_3\}\}$. A possible result is the argumentation framework F_2 described in Figure 6.1(b),



(a) The Input Framework F_1

(b) The Output Framework F_2

(c) The Output Framework F_3

Figure 6.1: Two Possible Results of the Argument-Fixed Enforcement

whose stable extensions are $Ext_{st}(F_2) = \{\{a_1, a_4\}, \{a_2, a_3\}\}$, and so E is enforced as a stable extension of the result. Another one is F_3 given in Figure 6.1(c), whose set of extensions is the same one: $Ext_{st}(F_3) = \{\{a_1, a_4\}, \{a_2, a_3\}\}$.

6.2.2 Minimal Change

As explained in Proposition 43, the possibility to enforce a set of argument is ensured when changes on the attack relation are allowed. From a practical point of view, it offers a success guarantee, which is a valuable property for an enforcement operator. Another expected property is minimal change, borrowed from belief revision. Enforcement processes can lead to several results, and the enforcement operators defined previously have to select one of the possible argumentation frameworks as the result. Considering minimal change during the enforcement process means that the chosen argumentation framework has to be as close as possible to the initial argumentation framework. [Bau12] already studies such a notion of closeness for the normal enforcement approaches. He defines minimal change as minimization of the number of attacks which are added to the argumentation framework during the enforcement process. We generalize this notion of minimal change, using the Hamming distance to measure how much two argumentation frameworks are different.

Definition 119.
We recall that the Hamming distance d_h between two argumentation frameworks $F = \langle A, R \rangle$ and $F' = \langle A', R' \rangle$ is defined by $d_h(F, F') = |(R \backslash R') \cup (R' \backslash R)|$.
For every enforcement operator $+$, the minimal change version $\underset{\min}{+}$ is such that the selected output argumentation framework F' minimizes the Hamming distance from the input argumentation framework F amongst the argumentation frameworks which are possible outputs of $+$.

6.3 Enforcement as Satisfaction and Optimization Problems

6.3.1 Complexity of Enforcement

A first observation is that enforcing a set of arguments while limiting the number of allowed changes in the attack relation is computationally demanding in the general case:

Proposition 44.
Let $F = \langle A, R\rangle$ be an argumentation framework, $E \subseteq A$, and k be an integer. Determining whether it is possible to enforce E in F under the stable semantics with at most k changes (addition or removal) of attacks is NP*-hard.*

Proposition 44 ensures that (unless P = NP) there is no polynomial-time algorithm to perform minimal change enforcement in the general case. For this reason, it makes sense to tackle the enforcement (respectively minimal change enforcement) issue using algorithms developed for solving (respectively optimizing) NP-hard problems. This is what we do in the following: we reduce enforcement to a propositional satisfiability problem, and minimal change enforcement to a pseudo-Boolean optimization problem.

6.3.2 Enforcement as Boolean Satisfaction

Our translation-based approach is based on the possibility to associate an argumentation framework F and a semantics σ with a propositional formula such that the models of the formula correspond exactly to the σ-extensions of F.

Definition 120.
Given F an argumentation framework and σ a semantics, Φ_σ^F is a propositional formula built upon the set of Boolean variables $\{x_a \mid a \in A\}$, such that $\{x_{a_1}, \ldots, x_{a_k}\}$ is a model of Φ_σ^F if and only if $\{a_1, \ldots, a_k\}$ is a σ-extension of F.

In the following, for a matter of simplification and since no ambiguity is possible, we write the formulae using a_i symbols instead of x_{a_i}. We focus on the encoding Φ_{st} of the stable extension, as given in [BD04]. Given $F = \langle A, R\rangle$, Φ_{st} is defined by

$$\Phi_{st} = \bigwedge_{a_i \in A} [a_i \Leftrightarrow (\bigwedge_{a_j : (a_j, a_i) \in R} \neg a_j)]$$

Then, checking if a set of arguments E is a stable extension of F is equivalent to checking the satisfiability of the formula $\Phi_{st,s}^E = \Phi_{st} \wedge (\bigwedge_{a_k \in E} a_k) \wedge (\bigwedge_{a_l \notin E} \neg a_l)$. To perform non-strict enforcement, a way to determine whether E is included in an extension is required. Dropping the conjunct $(\bigwedge_{a_l \notin E} \neg a_l)$ from the formula $\Phi_{st,s}^E$ gives precisely the formula Φ_{st}^E we need.

In order to link the semantics with the structure of the graph in the models of the formula, we introduce Boolean variables att_{a_i,a_j} meaning that there is an attack from a_i to a_j in F. The previous formulae are generalized into:

$$\Phi_{st}^{A,E} = \bigwedge_{a_i \in A} [a_i \Leftrightarrow (\bigwedge_{a_j \in A} (att_{a_j,a_i} \Rightarrow \neg a_j)] \wedge (\bigwedge_{a_k \in E} a_k)$$

and

$$\Phi_{st,s}^{A,E} = \bigwedge_{a_i \in A} [a_i \Leftrightarrow (\bigwedge_{a_j \in A} (att_{a_j,a_i} \Rightarrow \neg a_j)] \wedge (\bigwedge_{a_k \in E} a_k) \wedge (\bigwedge_{a_l \notin E} \neg a_l)$$

Clearly, propagating the truth values of the variables att_{a_i,a_j} in those formulae is enough to recover the previous formula $\Phi_{st,s}^E$ and the non-strict counterpart. This formula is the basis of our propositional encoding of extension enforcement operators.

Example 33 (Example 32 Continued)**.**
Considering the argumentation framework F_1, and $E = \{a_2, a_3\}$, the formula $\Phi_{st}^{A,E}$ is instantiated in

$$\begin{array}{lll} & a_1 & \Leftrightarrow [(att_{a_1,a_1} \Rightarrow \neg a_1) \wedge \cdots \wedge (att_{a_4,a_1} \Rightarrow \neg a_4)] \\ \wedge & a_2 & \Leftrightarrow [(att_{a_1,a_2} \Rightarrow \neg a_1) \wedge \cdots \wedge (att_{a_4,a_2} \Rightarrow \neg a_4)] \\ \wedge & a_3 & \Leftrightarrow [(att_{a_1,a_3} \Rightarrow \neg a_1) \wedge \cdots \wedge (att_{a_4,a_3} \Rightarrow \neg a_4)] \\ \wedge & a_3 & \Leftrightarrow [(att_{a_1,a_4} \Rightarrow \neg a_1) \wedge \cdots \wedge (att_{a_4,a_4} \Rightarrow \neg a_4)] \\ & & \wedge a_2 \wedge a_3 \end{array}$$

When we consider a model ω of this formula, we can generate an argumentation framework F_ω which corresponds to it, such that

$$F_\omega = \langle A, R_\omega \rangle \text{ with } R_\omega = \{(a_i, a_j) \mid a_i, a_j \in A, \omega(att_{a_i,a_j}) = true\}$$

and E is included in at least one stable extension of F_ω. If we consider the strict counterpart of the encoding, the process leads to argumentation frameworks F_ω such that E is an extension, thanks to the conjunction $a_2 \wedge a_3 \wedge \neg a_1 \wedge \neg a_4$.

Let us introduce more formally the decoding process described at Example 33. This decoding is performed by two functions similar to the ones used in Chapter 5 to decode the models of the revised formulae and generate the revised argumentation frameworks.

- $Proj_{att}^{A}(\Phi) = \{\omega \cap \{att_{a_i,a_j} \mid a_i, a_j \in A\} \mid \omega \models \Phi\}$ is the set of models of the formula Φ projected onto the att_{a_i,a_j} variables.
- $arg^A(\omega) = \langle A, R \rangle$ such that $(a_i, a_j) \in R$ if and only if $att_{a_i,a_j} \in \omega$, with ω a model projected onto the att_{a_i,a_j} variables, is the argumentation framework corresponding to the assignment of the att_{a_i,a_j} variables. Then, with Ω a set of such models, $arg^A(\Omega) = \{arg^A(\omega) \mid \omega \in \Omega\}$.

We also need an encoding for the structure of an argumentation framework $F = \langle A, R \rangle$:

$$struct_{A'}(F) = (\bigwedge_{(a_i,a_j) \in R} att_{a_i,a_j}) \wedge (\bigwedge_{(a_i,a_j) \notin R} \neg att_{a_i,a_j})$$

where $a_i, a_j \in A \cup A'$. This encoding allows to consider the additional arguments A' involved in the enforcement process (for normal, strong, weak and general approaches). $struct(F)$ is a notation for $struct_\emptyset(F)$.

Finally, $\delta : \{F_1, \ldots, F_k\} \rightarrow F_j$ such that $F_j \in \{F_1, \ldots, F_k\}$ is a tie-break rule which selects a single argumentation framework from a set of argumentation frameworks.

Now, every enforcement operator defined in the previous section can be encoded as a satisfaction problem on a propositional formula. Indeed, by construction, every model of the formula $\Phi_\sigma^{A \cup A',E}$, when projected onto the att_{a_i,a_j} variables, gives an argumentation framework which is a normal enforcement of E. We only need to state the right constraints for ensuring that strong (respectively weak) enforcement operators are reached. In order to avoid the introduction of new arguments and get argument-fixed operators, considering the formula $\Phi_\sigma^{A,E}$ as the encoding proves enough. Similarly, the formulae $\Phi_{\sigma,s}^{A \cup A',E}$ and $\Phi_{\sigma,s}^{A,E}$ can be used to define the strict counterparts of the enforcement operators.

Definition 121.
For any argumentation framework $F = \langle A, R \rangle$, any set of arguments $E \subseteq A$, any semantics σ, and $X = \sigma$ or $X = \sigma, s$:

- $F +_X^N E = \delta(arg^{A \cup A'}(Proj_{att}^{A \cup A'}(\Phi_X^{A \cup A', E} \wedge struct(F)))$
- $F +_X^{N,W} E = \delta(arg^{A \cup A'}(Proj_{att}^{A \cup A'}(\Phi_X^{A \cup A', E} \wedge struct(F) \wedge (\bigwedge_{(a_i,a_j) \in A' \times A} \neg att_{a_i,a_j}))))$
- $F +_X^{N,S} E = \delta(arg^{A \cup A'}(Proj_{att}^{A \cup A'}(\Phi_X^{A \cup A', E} \wedge struct(F) \wedge (\bigwedge_{(a_i,a_j) \in A \times A'} \neg att_{a_i,a_j}))))$
- $F +_X^A E = \delta(arg^A(Proj_{att}^A(\Phi_X^{A,E})))$
- $F +_X E = \delta(arg^{A \cup A'}(Proj_{att}^{A \cup A'}(\Phi_X^{A \cup A', E})))$

For any of these enforcement operators $+$, any argumentation framework $F = \langle A, R \rangle$ and any set of arguments $E \subseteq A$, $Enc(F + E)$ denotes the corresponding propositional encoding. For instance, $Enc(F +_\sigma^N E)$ is the propositional formula $\Phi_\sigma^{A \cup A', E} \wedge struct(F)$. Using any SAT solver to find a model of $Enc(F + E)$ and then decoding the truth values of the att_{a_i,a_j} variables is a way to determine an enforcement of E in F.

6.3.3 Minimal Change Enforcement as Pseudo-Boolean Optimization

As explained previously, [Bau12] considers a notion of minimal change enforcement. In his work, minimality refers to the minimality of the number of attacks to be added to the argumentation framework when performing the normal expansion. A possible way to ensure minimal change is to define a particular tie-break rule δ for selecting one of the resulting argumentation frameworks which is minimal. In order to take advantage of some available optimization software, an alternative approach is to encode the minimality criterion via a pseudo-Boolean objective function:

$$newAtt(A \cup A') = \sum_{(a_i,a_j) \in ((A \cup A') \times (A \cup A')) \setminus (A \times A)} att_{a_i,a_j}$$

Of course, for strong and weak enforcement operators, this representation of the objective function can be simplified since the att_{a_i,a_j} variables corresponding to the forbidden attacks are known to be $false$.

Minimal change for argument-fixed and general enforcement is not easy to be encoded directly using the available Boolean variables. In order to get the expected encoding, we consider additional variables representing the state of the argumentation framework before the enforcement; one then minimizes the number of differences between the truth values of these variables and the corresponding ones in the new argumentation framework. Formally, for every pair of arguments $(a_i, a_j) \in (A \cup A') \times (A \cup A')$, the Boolean variable $prev_{a_i,a_j}$ is $true$ if and only if $(a_i, a_j) \in R$. So, $prev_{a_i,a_j} \oplus att_{a_i,a_j}$, where $\oplus$ is the usual exclusive-or connective, gives the information about the change on the attack (a_i, a_j): if there was previously an attack from a_i to a_j, and this attack is no longer present after the enforcement, $prev_{a_i,a_j} \oplus att_{a_i,a_j}$ is $true$. It is also $true$ if there was no attack before the enforcement, and there is one after the enforcement. The encoding of the structure of the argumentation framework F must thus be updated to take account for the $prev_{a_i,a_j}$ variables:

$$struct_{A'}^{prev} = struct_{A'}(F)_{|att_{a_i,a_j} \leftarrow prev_{a_i,a_j}}$$

Once this is done, minimizing the differences on the attack relation is equivalent to minimizing the objective function

$$attChange(A \cup A') = \sum_{a \in (A \cup A'), b \in (A \cup A')} prev_{a_i,a_j} \oplus att_{a_i,a_j}$$

Clearly, this sum counts 1 for every attack (a_i, a_j) in the output argumentation framework concerning an argument of A', because $prev_{a_i,a_j}$ is always *false* if $a_i \in A'$ or $a_j \in A'$. So the approach can be used in the case of general enforcement.

We now sum up the definitions of the minimal change versions of the enforcement operators:

Definition 122.
For any argumentation framework $F = \langle A, R \rangle$, any set of arguments $E \subseteq A$,

- if $+$ is any enforcement operator among the normal, strong and weak enforcement operators (and their strict counterparts), then enforcing the set of arguments E in F is equivalent to satisfying $Enc(F + E)$ while minimizing $newAtt(A \cup A')$;
- if $+$ is any enforcement operator among the argument-fixed and general enforcement operators (and their strict counterparts), then enforcing the set of arguments E in F is equivalent to satisfying $Enc(F + E) \wedge struct^{prev}_{A'}(F)$ while minimizing $attChange(A \cup A')$.

We notice that using the second optimization problem would prove enough for each enforcement operator. But since this approach requires the addition of Boolean variables $prev_{a_i,a_j}$, we do not use it when it is not mandatory (*i.e.* for normal, weak and strong enforcement), to avoid any loss of computational efficiency.

The formal setting suited to our optimization problem is pseudo-Boolean (PB) optimization, which is an extension of Boolean satisfiability.

Definition 123.
Given a set of Boolean variables $V = \{x_1, \dots, x_n\}$ and a mapping $\mathcal{O} : \{0,1\}^n \mapsto \mathbb{R}$, a PB-Opt problem $\mathcal{P} = (\mathcal{C} = \{c_1, \dots, c_m\}, \mathcal{O})$ on V is the search for an assignment of every variable in V such that the contraints

$$\begin{array}{cc} c_1 : & w^1_1 x_1 + \cdots + w^1_n x_n \geq k^1 \\ \vdots & \vdots \\ c_m : & w^m_1 x_1 + \cdots + w^m_n x_n \geq k^m \end{array}$$

are satisfied and the objective function $\mathcal{O}$ reaches its optimal value.

In our case, the optimal value of the objective function is its minimal value. It is well-known that any propositional formula can be turned into an equivalent conjunctive normal form formula (CNF), and any clause of a CNF formula can be rewritten as a PB constraint: the clause $x_1 \vee x_2 \vee \cdots \vee x_n$ is satisfied if and only if the PB constraint $x_1 + x_2 + \cdots + x_n \geq 1$ is satisfied. Thus the optimization problem described previously can be rewritten easily in the PB setting.

Example 34 (Example 33 Continued)**.**
Let us illustrate the encoding of minimal change argument-fixed enforcement, with F_1 described previously and $E = \{a_2, a_3\}$. We describe the formula $\Phi^{A,E}_{st,s}$, which is exactly $Enc(F_1 +^A_{st,s} E)$, in the previous example. The objective function $attChange(A \cup A')$ is instanciated with $A' = \emptyset$ since argument-fixed enforcement does not allow to add arguments to F. So, the optimization problem to solve in order to enforce E in F_1 is:

Minimize

$$attChange(A)$$

Subject to

$$\Phi^{A,E}_{st,s} \wedge prev_{a_1,a_2} \wedge prev_{a_1,a_3} \wedge prev_{a_2,a_4} \wedge prev_{a_3,a_4}$$
$$\wedge (\bigwedge_{(a_i,a_j) \notin R} \neg prev_{a_i,a_j})$$

Both frameworks F_2 and F_3 given at Example 33 are the possible results when we use arg^A and $Proj^A_{att}$ on the solutions of this optimization problem, since each of them is a strict argument-fixed enforcement of E, and each of them differs from F_1 by only one attack which is added.

6.3.4 Constrained Enforcement

Similarly to the constrained revision defined in Chapter 5, we can take advantage of our propositional language to add an integrity constraint, which is represented as a formula over the set of variables $\{att_{a_i,a_j} \mid a_i, a_j \in A\}$. This kind of constraints indicates if some particular attacks are forced to belong to the argumentation framework, or on the opposite, must be forbidden.

Definition 124 (Integrity Constraint)**.**
Given a set of arguments A, an *integrity constraint on* A is a propositional formula c built up from the set of variables $\{att_{a_i,a_j} \mid a_i, a_j \in A\}$. Given an argumentation framework $F = \langle A, R \rangle$, we say that F *satisfies* c if and only if c admits a models ω such that, for each $a_i, a_j \in A$, $\omega(att_{a_i,a_j}) = true$ if and only if $(a_i, a_j) \in R$.

Definition 125 (Constrained Enforcement)**.**
Let $F = \langle A, R \rangle$ be an argumentation framework, c be an integrity constraint on A, and $E \subseteq A$ be a set of arguments. Given an enforcement operator $+$, the *constrained enforcement operator* $+^c$ is defined such that $F +^c E$ is a mapping from F and E to an argumentation framework F' such that F' is a possible output of $+$, and F' satisfies c.

Of course, even if the underlying operator $+$ is the argument-fixed or general enforcement operator, constrained enforcement may fail, since the integrity constraint can be unsatisfiable, given a particular argumentation framework and a particular set of arguments E to enforce. In particular, when the constraint c is such that $+^c$ is exactly the normal enforcement operator (or equivalently, the strong or weak enforcement operator), then it is already known that enforcement does not always succeed. Other natural cases of failure exist. The most simple one is the case of a constraint c which require an attack (a_i, a_j) to occur, while $a_i, a_j \in E$, which violates conflict-freeness principle. Other cases of failure can be related to the semantics. For instance, given $A = \{a_1, a_2, a_3\}$, if the constraint is $\neg att_{a_1,a_3} \wedge \neg att_{a_2,a_3}$, the strict argument-fixed enforcement of $\{a_1, a_2\}$ with respect to the stable semantics is not possible, since a stable extension must attack its complement (here, either a_1 or a_2 should attack a_3). The unsatisfiability of this constraint depends on the semantics. If the considered semantics is the grounded one, then the constraint can be satisfied: the argumentation framework F_4 given at Figure 6.2 proves enough.

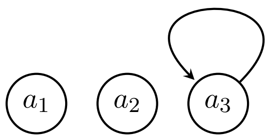


Figure 6.2: The Argumentation Framework F_4

6.4 Experimental Results

In our experimental study, we focused on the minimal change enforcement problem. We implemented the family of enforcement operators described in this paper, using the well-known tool `CPlex` [IBM14] as the underlying optimization engine. For a matter of readability, we present only the obtained results for three approaches: the non-strict strong operator from [BB10], and both the strict and non-strict versions of our argument-fixed enforcement operator. The behaviour of these operators is representative of the whole family. In each case, the semantics used is the stable one.

The empirical protocol we considered is as follows. We focused on some random argumentation frameworks [DJWW11, DJWW14]. Given a set of n arguments, each attack between two arguments is generated using a fixed probability p. In our experiments n varies up to 500 arguments. For each n, the graphs are divided into four families, corresponding to four values of p. We used families of argumentation frameworks from [DJWW11], where $p \in \{0.4, 0.65, 0.9\}$. We also generated argumentation frameworks with a probability $p = 0.1$. It appears in the experiments that the choice of p does not change significantly the performances of the translation-based enforcement algorithm, so the reported results are for $p = 0.1$ only.

We have computed the minimal change enforcement of sets E of arguments in argumentation frameworks F containing n arguments with $n \in \{200, 300, 400, 500\}$. For each argumentation framework F with n arguments, we considered sets E of arguments to be enforced of cardinality m, m varying between 1 and $\frac{35}{100}n$. For each pair of values (n, m), we generated 10 enforcement requests.[17] On Figure 6.3, the y-coordinate of each point of the following curves corresponds to the average computation time over all the pairs (F, E) which have been considered, where the number n of arguments of F is reported on the x-axis.

The first interesting result stemming from our experimentations is that enforcement looks feasible in practice on such randomly generated argumentation frameworks, which was not obvious given that enforcement is NP-hard; as illustrated by Figure 6.3, the computation time increases reasonably with the number n of arguments, up to a mean value of 13.76 seconds (std = 0.48) obtained for argumentation frameworks with 500 arguments, when strict argument-fixed enforcement is considered (+-curve), and up to a mean of 16.61 seconds (std = 5.31) when strong enforcement is considered ($\times$-curve). The non-strict argument-fixed enforcement is not presented on this figure. We will see later that this approach needs more time to obtain a result, depending on some parameters, and then the average time to compute the enforcement is not on the same scale as the average times presented here.

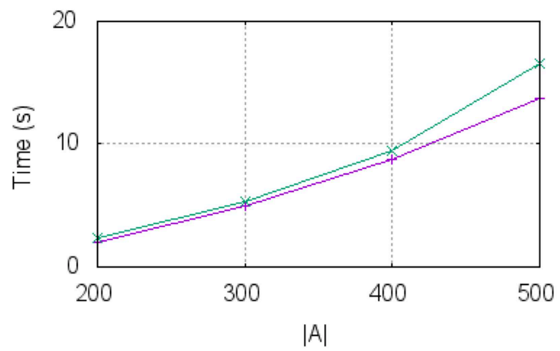
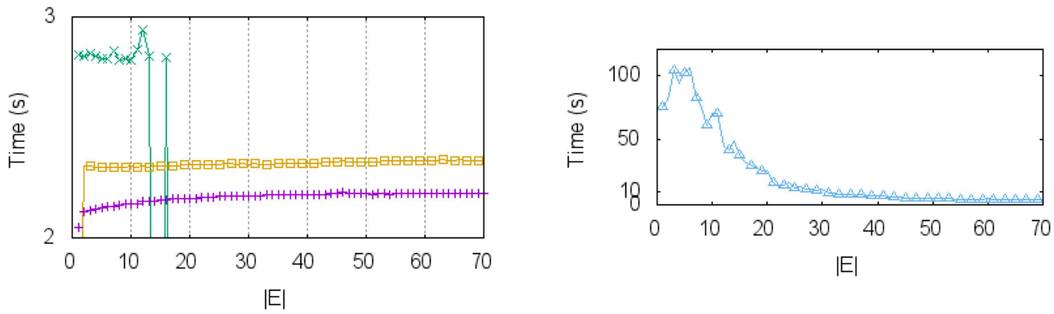


Figure 6.3: Average Time for Strong ($\times$-curve) and Strict Argument-Fixed (+-curve), n Varying from 200 to 500

[17] We call "enforcement request" the set E of arguments expected to be an extension.

Then, we compare the three different approaches on families of argumentation frameworks with 200 arguments, letting the cardinality m of E to vary from 1 to 70. The aim of this comparison is to study the impact of the cardinality of E on the enforcement operators behaviors. We did not discard trivial sets from the experiments, since our approaches can delete attacks, making a conflicting set conflict-free. This allows us to illustrate the failure rate of strong enforcement, which is unsurprisingly high, since it is impossible as soon as the enforcement request is not conflict-free in the input argumentation framework. With a probability p for an attack to occur between two arguments, the probability for a set of arguments E of cardinality m to be conflict-free is $(1-p)^m$. So, the greater the cardinality of the enforcement request, the lower the probability for enforcement to be possible. In particular, in our experiments strong enforcement always fails when $m > 20$; clearly, the failure rate of strong enforcement grows exponentially with m.



(a) Strict Argument-Fixed (+-curve) Enforcement, and Success (×-curve) and Failure (□-curve) for Strong Enforcement

(b) Non-Strict Argument-Fixed Enforcement

Figure 6.4: Average Time, $n = 200$, m Varying from 1 to 70

We compare the enforcement computing times for the three approaches (see Figure 6.4). The ×-curve represents the average time to realize the strong enforcement, while the □-curve corresponds to the time needed by the algorithm to report failure when enforcement is impossible. For strong and strict argument-fixed (+-curve) enforcement, it appears that the time needed for computing the result is almost always the same whatever the cardinality of the enforcement request and the probability of attacks in the graph: between 2 and 3 seconds.

The cardinality has more influence on the non-strict argument-fixed enforcement operator, the smallest enforcement requests being harder to compute. When the cardinality grows, the computing time decreases to a few seconds. There is an intuitive explanation for this phenomenon. When we consider an argumentation framework with n arguments, the number of Boolean variables required by our encoding is $2 \times n^2 + n$, since there are n variables a_i, n^2 variables att_{a_i,a_j} and also n^2 variables $prev_{a_i,a_j}$. As explained previously, the propositional formula which encodes the non-strict enforcement operators includes the conjunction $(\bigwedge_{a_k \in E} a_k)$, while the strict counterpart of the encoding includes the conjunction $(\bigwedge_{a_k \in E} a_k) \wedge (\bigwedge_{a_l \notin E} \neg a_l)$. This means that when the optimization engine is initialized, the truth values of n Boolean variables are known in the case of the strict argument-fixed enforcement, while only $|E|$ variables are known for the non-strict operator. For each variable which has its truth value already fixed, the size of the research space is divided by 2. Even if this is not the only parameter at play, this explains in part that the time to compute the non-strict argument-fixed enforcement is higher when the cardinaly of E is low (since the research space is wide), and it decreases when the cardinality of E grows (now, the research space is narrower than in the first case).

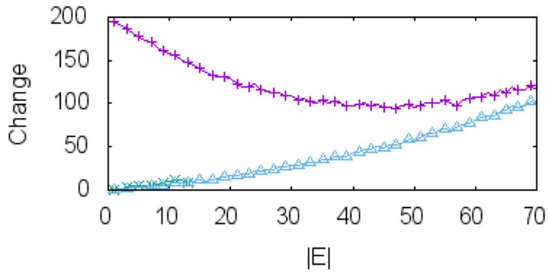


Figure 6.5: Average Change for Strong ($\times$-curve), Strict Argument-Fixed ($+$-curve) and Non-Strict Argument-Fixed ($\triangle$-curve) Enforcement

Lastly, we want to measure the effort required in term of change (i.e., the number of attacks to be added or deleted) for enforcing E in the argumentation framework (see Figure 6.5). Clearly, the effort needed grows up with the cardinality of the enforcement request for strong ($\times$-curve) and non-strict argument-fixed enforcement ($\triangle$-curve).
This comes from the fact that the enforcement is non-strict: to enforce a set $E = \{a_i\}$, adding a new argument $b \notin A$ (in the case of strong enforcement) or choosing an accepted argument $a_j \in A$ (in the case of argument-fixed enforcement), and making it to defend a_i against each of its attackers proves enough to include a_i in a stable extension (which will also contain b or a_j, depending on the considered enforcement method, and possibly some other arguments).
We notice that the curve for strong enforcement stops much before the other ones (at 14 arguments) because of the high failure rate we mentioned. Until this point, as one can observe, this curve is almost identical to the $\triangle$-curve.

Strict argument-fixed enforcement ($+$-curve) requires much more change on the attack relation. As the enforced set is expected to be exactly a stable extension, even a small set of arguments needs the addition of many attacks to be enforced. For instance, with $E = \{a_i\}$, it is required that a_i attacks each other argument to ensure that E is a stable extension, while the set $E' = \{a_1, a_2, \dots, a_{30}\}$ only needs to attack 170 arguments to be a stable extension, rather than 199. However, after reaching a minimal value when m is between 40 and 50, the effort required by the strict argument-fixed enforcement grows up when the cardinality m of E grows up. Intuitively, we understand that it is "harder" to make conflict free E when $m = 70$ than when $m = 40$, since there are possibly much more attacks inside E.
For the highest cardinalities in the study (from $m = 60$ to $m = 70$), the efforts required by the strict and the non-strict version of the argument-fixed operator come closer (between 100 and 200).

After this analysis of the behaviour of the different enforcement operators on argumentation frameworks with 200 arguments, let us briefly describe how our approaches behave in front of larger argumentation frameworks.
First of all, we notice that the behaviour of the non-strict argument-fixed enforcement is similar to what happens with 200 arguments. When the argumentation framework contains 300 arguments, the average time to compute the enforcement is around 70 seconds for small cardinalities of E, and it decreases to about 6 seconds when the cardinality of E becomes closer to 200, which is a upper bound that we used

on this part of the experiments. When we consider argumentation frameworks with 400 arguments, the same phenomenon is observed, but with higher average times: 402.46 seconds when E is a singleton, and it decreases from this maximal value to about 13 seconds when the cardinality of E is 200. The difficulty to compute non-strict argument-fixed enforcement grows again when we consider argumentation frameworks with 500 arguments, and we obtain a very high level of timeout for this family of instances: most of the experiments are stopped after the threshold of 900 seconds.

The behaviours of the other enforcement approaches are also similar to their behaviour when the considered instances contain 200 arguments. Table 6.1 summarizes the results about non-strict strong enforcement and strict argument-fixed enforcement operators when 300, 400 and 500 arguments are considered. The "Minimal" and "Maximal" values are the average times to compute the enforcement (or to state that enforcement fails, in the case of strong enforcement) for some cardinality of E. The "Average" values are the global average computing time for the given operator and the given size of argumentation frameworks, for every cardinality of E.

Operator	$n = 300$			$n = 400$			$n = 500$		
	Minimal	Maximal	Average	Minimal	Maximal	Average	Minimal	Maximal	Average
Strong	5.24	6.49	5.36	9.31	11.49	9.51	14.58	49.46	16.61
Argument-fixed	4.10	5.04	4.97	8.36	8.96	8.81	13.3	15.64	13.76

Table 6.1: Average Computing Times, Rounded at 10^{-2}s, for Non-Strict Strong and Strict Argument-Fixed Enforcement, for $n \in \{300, 400, 500\}$

Now let us describe the results in term of minimal change. Strict argument-fixed enforcement gives the same kind of results. For instance, when considering 500 arguments, the minimal change to enforce a singleton is 495.66 seconds. It decreases to about 260 secondes, when the cardinality of E is around 130, and it grows back to 309.55 seconds with the highest cardinality. We observe the same phenomenon with 300 and 400 arguments.

Again, for $n \in \{300, 400, 500\}$, we observe the same phenomenon for strong enforcement and non-strict argument-fixed enforcement then for $n = 200$. The effort needed is almost null for the lowest cardinalities of E, and it grows with the cardinality. The results for both operators are almost identical until the threshold between success and failure of strong enforcement.

To conclude, let us recall that our approach scales up well, in particular for the strong enforcement and strict argument-fixed enforcement approaches. For the benchmarks used during these experimentations, the cardinality of the enforcement request does not influence the time to compute the result for both these approaches. However, the non-strict argument-fixed enforcement is more sensible to the cardinality of the enforcement request. This is why, for small cardinalities, this operator needs more time to give a result. However, when the cardinality is higher, even this operator gives a result in a few seconds on average.

6.5 Related Work

It is worth noticing that some other kinds of change operations can be recovered as specific extension enforcements. For instance, credulous explanation in [BGK$^+$14] is the search of an argumentation framework which justifies that a given argument a_i is credulously accepted. It can be translated as an enforcement problem: it is equivalent to the non-strict enforcement of the singleton $\{a_i\}$. The argumentation framework which explains the status of a_i must belong to a set of argumentation frameworks, called abducible argumentation frameworks, which are the possible options for the agent to change her argumentation framework. Then, using a general enforcement operator with an integrity constraint which encodes the set of abducible argumentation frameworks proves enough to reproduce the behaviour of credulous explanation.

Similarly, some goal-oriented changes from [KBM$^+$13] are enforcement operators: the credulous positive goals like "the argument a_i must be credulously accepted with respect to the considered semantics" are also equivalent to the non-strict enforcement of the singleton $\{a_i\}$.

Our translation-based enforcement approach, and the prototype software that we have implemented, thus prove also useful for achieving such kinds of changes in argumentation frameworks.

The contributions of this thesis about revision of argumentation frameworks are also related to extension enforcement. This time, we do not propose to encode these previous works as extension enforcement, but on the opposite, extension enforcement can be encoded as a revision. Indeed, the revision approaches described in Chapter 4 and Chapter 5 allow to ensure that a set of arguments will belong to every extension of the result. In the first case, enforcing $E = \{a_1, \dots, a_n\}$ consists in a revision by the formula $\varphi_1 = a_1 \wedge \dots \wedge a_n$, meaning that each extension of the result must contain each of the arguments $a_1, \dots, a_n$. In the case of the translation-based revision, the formula $\varphi_2 = acc_{a_1} \wedge \dots \wedge acc_{a_n}$ leads to the same result: each of the arguments $a_1, \dots, a_n$ must be skeptically accepted, and so must belong to each of the extensions. If E is included in the intersection of the extensions of the result of the revision, it means obviously that this revision correspond to a non-strict enforcement of E. Using the constrained open world revision, which is defined for both revision approaches, we can encode a general enforcement operator, and each of the subclasses of the general enforcement.

6.6 Conclusion

In this chapter, we have investigated the problem of enforcing a set of arguments as an extension of an argumentation framework. Our contribution is manyfold. First, we have shown that existing approaches to enforcement may fail, even when the set of arguments to be enforced is conflict-free. To overcome this weakness and to allow more general cases of enforcement, some new enforcement methods for which the success of the process can be guaranteed have been defined. For each of these methods, we designed some Boolean encodings which allow to take advantage of satisfaction and optimization solvers for the enforcement purpose. We used CPlex, a well-known optimization tool, to implement a library of enforcement operators, and we experimented some of them on a large class of benchmarks. The experimentations showed the approach to be practical and to scale up well.

This work opens several perspectives for further research. As far as we know, none of the existing works about change in argumentation frameworks has led to the implementation of some (quite efficient) piece of software. However, implementing practical argumentation systems is currently a hot topic for the community (in the same vein, see the organization of a competition of argumentation solvers [TV15]). Indeed, the design of our enforcement software comes from the same will to make available argumentation reasoning tools, which is nowadays a necessary step to push forward the domain. A first objective for this software is the search for a more efficient method to compute non-strict argument-fixed enforcement, since the current method has the same efficiency as the other operators only when the cardinality of the enforcement request is high. A translation of this enforcement approach into an optimization problem which is also efficient with low cardinality enforcement requests is necessary to improve the current features of our software.

Then, we want to encode and implement enforcement operators for other semantics. In particular, similarly to the case of the translation-based approach for revision that we have presented previously, the use of QBF encodings and QBF solvers is promising for the semantics with high complexity. Some further extensions of the setting will be also envisioned. For instance, using other objective functions for the optimization problem leads to define some other types of minimality. In particular, we have in mind the encoding of minimal change on arguments statuses, considered in our first contributions.

Chapter 7

On Constraints and Change in Argumentation

It is change, continuing change, inevitable change, that is the dominant factor in society today.

Isaac Asimov – *"My Own View" , The Encyclopedia of Science Fiction*

Our first contributions focused on revision and extension enforcement in Dung's framework. Now, we consider argumentation in a more abstract way: this chapter addresses the issue of the dynamic enforcement of a constraint in an argumentation system, which consists of

1. an argumentation framework, made up, notably, of a set of arguments and of an attack relation,
2. an evaluation semantics, and
3. the evaluation result, computed from 1 and 2.

Of course, the argumentation framework may be a Dung's abstract argumentation framework, but it can be also any of its enrichments, or even any other representation of argumentation.

An agent may want another one to consider a new attack, or to have a given argument accepted, or even to relax the definition of the semantics. A constraint on any of the three components is thus defined, and it has to be enforced in the system. The enforcement may result in changes on components of the system. This chapter briefly recalls the existing approaches for the dynamic enforcement of a constraint in an argumentation framework, and classifies them depending on the kind of constraints and change that they apply. We reveal challenging enforcement cases that remain to be investigated.
We also sketch an approach to define generalized enforcement operators, and we show how to extend our logic-based approaches for revision and extension enforcement to this purpose.

Contents

7.1 Argumentation System

Since Dung's framework is the most influential setting to reason about arguments, and also the one which has received the most attention during the recent years, this was the single representation of argumentation that was considered in our first contributions. But we can also envision dynamic scenarios when the agent uses another kind of argumentation framework. This is why we consider, at a more abstract level, an argumentation system as the global object which contains the arguments and the relations between them, the method to evaluate the statuses of the arguments, and the result of this evaluation.

Definition 126 (Argumentation System)**.**
An *argumentation system* is defined as a set of three components:

1. an *argumentation framework* F, which generally consists of a set of arguments and one or several relation(s) between them;
2. a *semantics* σ that gives a formal definition of a method (either declarative or procedural) ruling the argument evaluation process;
3. an *argument evaluation* $\mathcal{E}_\sigma(F)$, which is the result of the application of the semantics (component 2) on the argumentation framework (component 1).

The argumentation framework and the semantics can be seen as the input of the system, the argument evaluation as the output. Figure 7.1 illustrates this architecture.

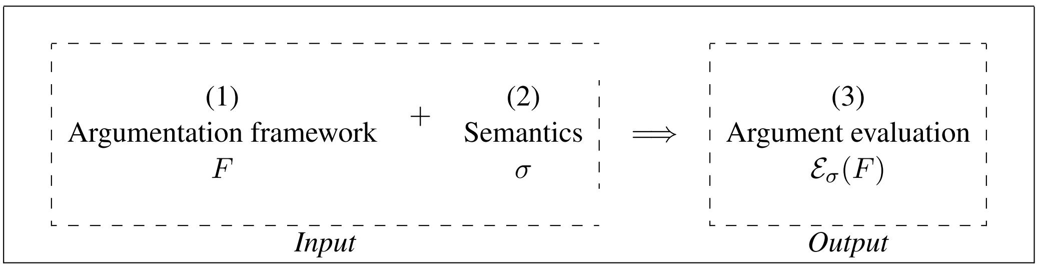


Figure 7.1: Argumentation System

As an example of an argumentation framework (component 1), one may, of course, consider Dung's framework [Dun95], as it is the case in the previous chapters of this thesis. One may consider as well an extended version of this framework, such as:

- a preference-based argumentation framework (PAF) by [AC02b, AC02a], that considers a preference relation which influences the success of attacks;
- a value-based argumentation framework (VAF) by [BC02, BC03], related to the previous one, that attaches values to arguments and handles preferences over values;
- a bipolar argumentation framework (BAF) by [CLS05], that additionally takes into account a support relation between arguments;
- a constrained argumentation framework (CAF) by [CDM06], that adds a constraint over the set of arguments and the attack relation to be taken into account;
- an argumentation frameworks with necessities (AFN) by [NR11], where the attack relation is completed by a support relation which represents the fact that an argument's acceptance is required for another one to be accepted.

Frameworks different from Dung's one may be considered as well, such as abstract dialectical frameworks (ADF) by [BW10], that are based on a set of arguments and attach an acceptability condition to each argument.

A huge range of semantics (component 2) have been defined so far (see [BG09] for an overview). They can be classified into three categories: extension-based semantics (e.g. [Dun95]); labelling-based semantics (e.g. [Cam06]), which are a refinement of extension-based semantics, as already explained in this thesis; and ranking-based semantics (e.g. [ABN13]). A semantics is usually defined for a certain kind of argumentation framework. We have already presented in details the most usual extension-based semantics, and their labelling-based counterpart. Contrary to these ones, which give a precise acceptance status to each argument, a ranking-based semantics produces a total pre-order over the set of arguments: $\mathcal{E}_\sigma(F)$ allows to decide if an argument is "better" than another one. Evaluation principles that underly most of the existing semantics have been identified ([BG07, BCG11]).

Each of the three components of an argumentation system may be subject to some dynamics: some constraint may have to be enforced on one or several components, and enforcement generally causes changes on the system. Such constraints and changes are presented in the next section.

7.2 Three Kinds of Constraints

This section presents the different kinds of constraints that can be considered in an argumentation system, and the changes that their enforcement implies. We illustrate the different types of constraints with the existing approaches for change of argumentation frameworks that we described in Chapter 3 and with our own works presented in Chapter 4, 5 and 6.

7.2.1 Structural Constraints

The first kind of constraint that may have to be taken into account, concerns the argumentation framework (component 1). Typically, when an argumentation-based debate takes place between agents, new arguments, new attacks, may have to be additionally considered. These new elements deal with the structure of the argumentation framework, and represent constraints that must be taken into account. For this reason, we call them *structural constraints*.

If these constraints directly concern the elements of the argumentation framework, namely the arguments and the relations between arguments, they may be called *elementary* structural constraints. Other

structural constraints may address the whole structure of the argumentation framework; these constraints are *global* structural constraints. As an example of such global constraints, one may wish the argumentation framework to be acyclic, or without any odd-length cycle, or to be made of only one connected component. The need for such global constraints may be motivated by computational concerns: it is known that argument evaluation, under some semantics, for argumentation frameworks with particular properties, are easier to compute than for argumentation frameworks without these properties.
For instance, we know that Dung's argumentation frameworks possess a single extension which is grounded, stable, preferred and complete when they are acyclic; similarly, the stable semantics and the preferred semantics coincide when the argumentation framework does not contain any odd-length cycle. These properties allow to compute the usual inference tasks with polynomial time algorithms (for the acyclic argumentation frameworks) or with a call to an NP oracle, while reasoning under the preferred semantics is at the second level of the polynomial hierarchy in the worst case.

As far as we know, only elementary structural constraints have been studied in the existing approaches on change in argumentation systems, although some of them could be easily adapted to tackle global constraints. For instance, [BKvdT09b, BKvdT09a, CdSCLS10] list the elementary constraints existing in Dung's framework: adding or removing one attack between two arguments which belong to the framework, and adding or removing an argument (with the attacks which concern it). The existing approaches on change in argumentation use these kinds of constraints, or combinations of them. For instance, [BCdSL13, DHP14] and our translation-based revision approach (see Chapter 5) encode the argumentation framework and the semantics in a logical setting, and use formulae which can represent constraints like "an attack from a_i to a_k and an attack from a_k to a_j must be added". This kind of logical language allows to combine elementary constraints to express more complex ones. In particular, global constraints such that acyclicity constraint could be encoded in this language, either as the reason of the change, or as an integrity constraint to be satisfied when a change is performed.

The enforcement of a structural constraint on an argumentation framework F obviously leads to a change of the argumentation framework (what we call a *structural change*), to an argumentation framework F', but it may also impact the argument evaluation (*acceptability change*): $\mathcal{E}_\sigma(F)$ may be different from $\mathcal{E}_\sigma(F')$. For instance, if one considers Dung's framework, if an argument which was previously attacked becomes unattacked after the enforcement of the constraint, then it has to appear in each extension, whatever the semantics. If one would like, however, the argument evaluation to stay unchanged, then the structural constraint would have to come together with an acceptability constraint (see Section 7.2.2), which would require the evaluation to be the same. [CdSCLS10] pesents a typology of the acceptability changes induced by the enforcement of some elementary structural constraints.

Structural constraints also make sense if extended versions of Dung's framework are considered. For instance, any kind of attack-addition or attack-removal operation can have a support-addition or support-removal (respectively necessity-addition or necessity-removal) counterpart when considering bipolar argumentation frameworks [CLS05] (respectively argumentation frameworks with necessities [NR11]). If we work with preference-based argumentation framework or value-based argumentation frameworks [AC02b, BC02, BC03], then a change of the preference relation between arguments or between the values can occur. In this last case, there can also be directly a change of the values associated with the arguments. Structural constraints in such frameworks remain to be formally investigated.

7.2.2 Acceptability Constraints

The dynamics of an argumentation framework may also originate in a need for a change of the argument evaluation. For example, an agent may want another one to consider an argument as acceptable, or to consider it "better" than another one, whereas it currently is not the case. This is what we call an *acceptability constraint* that has to be enforced on an argumentation system. This acceptability constraint depends on the kind of evaluation which is used, among the extension-based semantics, the labelling-based semantics and the ranking-based semantics. For each of these evaluation approaches, several kinds of constraints make sense.

Extension-based semantics are maybe the most well-known approaches for argument evaluation [Dun95]. The first work on acceptability constraints is the extension enforcement issue [BB10, Bau12], which we also studied in this thesis (see Chapter 6). [CDM06] and our first work on the revision of argumentation frameworks (see Chapter 4) express the acceptability constraint as a propositional formula over the set of arguments, which has to be satisfied by each extension of the argumentation framework. The same approach is developed in [DHL$^+$15], which also proposes to represent this acceptability constraint by an argumentation framework F' such that the extensions of the outcome of the enforcement are a subset of $\mathcal{E}_\sigma(F')$. In [BCdSL13, DHP14], as well as our translation-based revision approach (see Chapter 5), it is possible to express a constraint on the acceptance status of an argument, such that "argument a_i must be credulously (or skeptically) accepted", credulous (respectively skeptical) acceptance meaning that a_i must belong to at least one (respectively every) extension.
Like structural constraints, acceptability constraints can be categorized into *elementary* acceptability constraints, that concern the particular acceptability of some arguments, or of some sets of arguments, and *global* acceptability constraints, that concern the structure of the argument evaluation (number of extensions, size of the extensions, for instance).

The same kind of acceptability constraints can be considered for labelling-based semantics, since they are a refinement of the extension-based semantics, but with more expressiveness. For instance, it is possible to require an argument evaluation to satisfy the constraint "argument a_i must be out", which is more precise than requiring an argument not to belong to an extension (since in this case, the argument may be either rejected or undecided). Such labelling-based constraints are considered in [BKRvdT13, CMKMM14b], and also in Chapter 4.

Ranking-based semantics are also subject to "acceptability" constraints. In this case, it is more accurate to speak about "evaluation" constraint, since these semantics do not lead to decide if an argument is accepted or not. Several levels of constraints can be defined. It makes sense to enforce a constraint such that "argument a_i is ranked lower than argument a_j holds in the outcome of the enforcement". These constraints, of course, can be combined together for different values of a_i, a_j, and these combinations may lead to require the argument evaluation to be exactly a given order when each possible pair (a_i, a_j) is considered. Up to our knowledge, the characterization and the enforcement of such evaluation constraints has not yet been addressed. Global constraints on ranking-based semantics also make sense, such that "the outcome of the evaluation is a total pre-order between the arguments". It is supposed to be the case with the semantics defined in [ABN13], but this constraint is reasonable if the agent considers some other ranking-based semantics [CL05, GM15].

Regarding the enforcement of an acceptability constraint, the most common method that can be found in the literature consists in changing the argumentation system so that the argument evaluation of the modified system satisfies the constraint. To this end, *structural change* and *semantic change* are both possible.

For instance, [BB10, Bau12] expand an argumentation framework *à la* Dung by a set of new arguments and a set of new attacks concerning these new arguments (and possibly the former ones). It also considers the possibility to change the semantics. Our own work presented in Chapter 6 is a follow-up of the previous ones, in which arbitrary modifications of the structure of the graph are allowed.

The update approaches described in [DHP14] only permit to change the attack relation to satisfy the constraint, while our revision approaches described in Chapter 4 and Chapter 5, although initially designed similarly, also allow to add arguments. In other words, the enforcement is done by a structural change.

As explained in Chapter 3, [BKRvdT13] presents two approaches to satisfy the acceptability constraint. This extension of Dung's framework uses a propositional formula on labellings as an integrity constraint, and considers that the agent's beliefs are the complete labellings of the argumentation framework which satisfy this integrity constraint. Both approaches are used to restore consistency if there is no complete labelling of the framework which satisfies the integrity constraint. The first one is similar to the extension enforcement described in [BB10]. The other one also uses framework expansion, but is a bit more subtle. It takes advantage of belief revision techniques to compute what is called the "fallback beliefs", which are consistent subsets of the current agent beliefs which are the most plausible ones. Then a framework expansion is performed to match these fallback beliefs.

It can be noticed that the enforcement of an acceptability constraint, by a semantic change only, has not yet been addressed. Such an enforcement may however be relevant in case when a structural change is not possible, or not suitable.

The resulting argumentation system after the enforcement of an acceptability constraint may be such that the initial argumentation framework F is modified into an F' (structural change), and/or the initial semantics σ has turned into a semantics σ' (semantic change). In any case, there is an *acceptability change*, that captures the enforcement of the acceptability constraint, but that may also partly result from the structural/semantic change which has been set up to enforce the acceptability constraint. For instance, if an acceptability constraint consists in requiring that "argument a_i belongs to every extension", a structural change that consists in removing all the attacks to this argument, may be carried out; but with this new status, a_i may now be able to make other arguments to belong to some or every extensions: additional acceptability changes hence occur. Minimizing the impact of the enforcement of an acceptability constraint, is a quality requirement that has already been considered in several contributions; this issue is addressed in Section 7.3.

Another method to enforce an acceptability constraint consists in using an *integrity constraint*. Such an approach has been proposed in [CDM06]. In this extension of Dung's framework, the constraint is a propositional formula on the extensions of the framework similar to the revision formulae that we use in Chapter 4. Contrary to [BKRvdT13] this constraint does not lead to a change of the argumentation framework. The semantics which have been defined for, and which have to be used with a constrained argumentation framework, take into account the integrity constraint, and ensure that it is satisfied in the argument evaluation.

7.2.3 Semantic Constraints

Now, let us focus on the constraints dealing with the second component of the argumentation system: its semantics. We have seen in the previous section that a semantic change is a way to enforce an acceptability constraint. It may as well be the case that an agent may want a semantics to change, wholly, or partly; that is, the agent may want a *semantic constraint* to be enforced.

The motivations for such constraints are diverse. For instance, the case of the empty set of extensions, which is allowed for some semantics, may be a weakness for some applications which absolutely require a solution. This may lead to a semantic constraint such that "replace the stable semantics by the complete semantics". On the other hand, we know that the number of extensions may be exponential in the number of arguments for some semantics. This may be a problem from a computational point of view to enumerate an exponential number of extensions. Some applications may also need a "simple" answer, and so it makes sense to replace the semantics σ by another one σ' which guarantees that $|\mathcal{E}_{\sigma'}(F)| < |\mathcal{E}_{\sigma}(F)|$. The extreme scenario is to require a single extension in the evaluation of the system, for instance with a replacement of σ by the grounded semantics.

The semantic constraint may concern the semantics on its whole (as indicated in the previous paragraph). But a more elaborated kind of semantic constraint can be defined, dealing with the semantics principles: if some of the principles of the semantics cause a problem (from a computational or a reasoning point of view) they can be dropped. On the opposite, the agent can consider that her reasoning scheme is not demanding enough, and add some principle to its current semantics. For instance, the maximality principle may be too costly to compute, and then should be relaxed, while the complement attack principle may be required to ensure that each argument which is not in an extension is rejected for a good reason.

The enforcement of a semantic constraint results in a *semantic change* (the original semantics σ evolves to σ'). This change may lead to an acceptability change, as it may have an impact on the argument evaluation ($\mathcal{E}_{\sigma'}(F)$ may be different from $\mathcal{E}_{\sigma}(F)$).

Few approaches study semantic constraint and semantic change in argumentation. First, [DW11] studies the relative expressiveness of a wide range of usual acceptability semantics: they provide some translations from an argumentation framework *à la* Dung F and a semantics σ to a framework F' and a semantics σ' such that $\mathcal{E}_{\sigma}(F)$ and $\mathcal{E}_{\sigma'}(F')$ satisfy some property. For instance, this translation is called "exact" if and only if $\mathcal{E}_{\sigma}(F) = \mathcal{E}_{\sigma'}(F')$, and "faithful" if each element from $\mathcal{E}_{\sigma'}(F')$ is equal to an element from $\mathcal{E}_{\sigma}(F)$ plus some new arguments which belong to F' but not to F.

The framework described in [BB10, Bau12] for enforcement of a set of arguments takes a semantics as a parameter, and so the target set of arguments does not necessarily have to be (included in) a σ-extension of the expanded argumentation framework, but possibly (included in) a σ'-extension. They distinguish "conservative" enforcement (if $\sigma = \sigma'$) and "liberal" enforcement (if $\sigma \neq \sigma'$).

The last existing work on semantic constraints is the study of the realizability of a set of candidates [DDLW14]. We recall that the authors identify some necessary and sufficient condition, for some usual semantics, for a set E of candidates to be realizable with respect to σ, and they prove that this test can be done in polynomial time for most of the usual semantics.

7.2.4 Combinations of Constraints

Of course, the different kinds of constraint described previously can be combined with each other. It is already the case with realizibility checking [DDLW14], which is the combination of a semantic constraint (the expected extension-based semantics σ is a parameter) and an acceptability constraint (demanding a particular set of candidates to be the σ-extensions).

[DW11] may also be seen as a combination of a semantic constraint, along with an acceptability constraint (the extensions under the new semantics should be in correspondence with the ones under the original one). Similarly, extension enforcement approaches [BB10, Bau12] combine an acceptability constraint (the set of arguments expected to be included in an extension) and a semantic constraint.

On the other hand, [BCdSL13, DHP14] and our revision approaches presented in Chapter 5 combine acceptability and structural constraints in their propositional language over the set of arguments.

It seems that in a dynamic context, any kind of constraints combination makes sense. It can be noticed however that a constraint such as "The structure of the argumentation graph (respectively the semantics, the argument evaluation) must not change" makes sense only when it is considered in combination with another constraint.

Semantic constraints are particularly meaningful when considered in combination with an acceptability constraint. It is possible that a particular acceptability constraint cannot be enforced, with respect to some given semantics. In this case, it makes sense to have a possibility to switch the semantics for another one which permits to enforce the acceptability constraint.
To illustrate this case, let us consider an agent which uses the preferred semantics pr to reason with arguments. She can receive a full piece of information about the evaluation, leading to demand a set $E = \{\varepsilon_1, \ldots, \varepsilon_n\}$ to be the extensions of her argumentation framework. But the direct enforcement of the constraint "Build F such that $\mathcal{E}_{pr}(F) = E$" may lead to a problem, since each set of candidates is not realizable for each semantics. A more elaborated constraint like "Build F such that $\mathcal{E}_\sigma(F) = E$ for some σ" allows to avoid this problem.

7.3 Quality of Enforcement

Whatever the kind of enforcement, some notion of quality can be considered. Among several possible solutions to the expected enforcement request, all of them are not equally satisfying for the agent. It can be expressed in several different ways.

7.3.1 Minimal Change

The most obvious one is probably minimality of change, borrowed from belief change [AGM85, KM91]. In this framework, minimal change is a desirable property because an agent expects to avoid any unnecessary loss of information when performing a belief change. We already explained that the notion of minimality is not obviously defined in argumentation settings. Since enforcement in argumentation frameworks deals with three different kinds of constraints and changes, we can consider at least one kind of minimality for each of these kinds of constraints and change.

Minimal change on the argument graph (*minimal structural change*) is the first kind of notion of quality which has been studied [Bau12]. Baumann considers that the predominant information for the agent is the structure of the graph, and minimal change is expressed as the minimization of the number of attacks which are changed in the argumentation framework. We borrow the same notion of minimality in our study of extension enforcement (see Chapter 6). The same kind of minimality is used in [DHP14] and in Chapter 5. Minimization of the changes on the set of arguments, or on other components of the argumentation framework if any, may also be considered.

Another kind of minimality concerns the changes on the output of the argumentation system: the acceptability of arguments. The different possibilities to express *minimal acceptability change* in this case depend on the different expressions of acceptability: skeptical acceptance, credulous acceptance, extension (or labellings) enumerations, rankings... For instance, the approach described in Chapter 4, borrowed from the notion of minimal change in belief revision in propositional logic, considers the set of extensions of the argumentation framework and uses distances between sets of extensions to decide which output is the minimal one for the revision of an argumentation framework, that is, which one enforces the acceptability constraint, and induces a minimal additional acceptability change. This work has been followed by [DHL$^+$15]. Another possible approach to define minimal change on the acceptability is to use distances between sets of arguments, for instance to quantify the difference between the skeptically accepted arguments of two different argumentation frameworks as we have done in Chapter 5.

It is not so obvious to define *minimal semantic change*. We can consider two kinds of minimal change, depending on the level of change expected on the semantics: either the change concerns directly a replacement of the semantics σ to the semantics σ', or the change concerns the addition or removal of some principles of the semantics. As far as we know, none of these solutions has been studied in depth, but we can initiate some research tracks.

First, let us focus on the replacement of semantics. We know that the usual semantics satisfy some inclusion relations: for instance, each stable extension of an argumentation framework F is also a preferred extension of F, and each preferred extension of F is also one of its complete extensions. We use this information to define the semantics dependence graph:

Definition 127.
Let $\Sigma = \{\sigma_1, \dots, \sigma_n\}$ a set of acceptability semantics. The *semantics dependence graph* of Σ is defined by $Dep(\Sigma) = \langle \Sigma, D \rangle$ with $D \subseteq \Sigma \times \Sigma$ such that $(\sigma_i, \sigma_j) \in D$ if and only if:

- for each argumentation framework F, $\mathcal{E}_{\sigma_i}(F) \subseteq \mathcal{E}_{\sigma_j}(F)$;
- there is no $\sigma_k \in \Sigma$ $(k \neq i, k \neq j)$ such that for each argumentation framework F, $\mathcal{E}_{\sigma_i}(F) \subseteq \mathcal{E}_{\sigma_k}(F)$ and $\mathcal{E}_{\sigma_k}(F) \subseteq \mathcal{E}_{\sigma_j}(F)$.

The distance between two semantics σ_i and σ_j in a semantics dependence graph $Dep(\Sigma)$ can be defined as the length of the (non-directed) path between σ_i and σ_j. Let us illustrate minimal change on an example:

Example 35.
Let us use the set of semantics $\Sigma = \{$complete,preferred,stable$\}$. The semantics dependence graph is given on Figure 7.2. Let us suppose that we want to enforce an acceptability constraint in the argu-

Figure 7.2: The Semantics Dependence Graph of Σ

mentation framework F which is not satisfied by the stable semantics. If we know that it is possible to enforce the constraint in F just with a change of semantics, from stable to preferred or complete, then it is minimal to choose the preferred semantics, because the distance between stable and preferred is 1, and the distance between stable and complete is 2.

Now, if we consider semantic change as a modification of some principles of the semantics, rather than a whole modification of the semantics, a possible way to quantify change is to specify exactly the set of principles satisfied by the semantics, and to consider a cost on each addition or removal of principles in the set.

Example 36.
The stable semantics is the conjunction of two principles: conflict-freeness and complement attack. The complete semantics combines conflict-freeness and defense, while the preferred semantics adds the maximality principle to the complete semantics. Then, switching from the stable to the complete semantics causes a change on two principles (removal of complement attack, and addition of defense), while switching from the stable to the preferred semantics causes a change on three principles (due to the requirement of maximality).

We observe that depending on the way to *quantify semantic change*, minimal change from the stable semantics may lead to favour the preferred semantics over the complete one in Example 35, or the opposite in Example 36. This issue deserves to be more formally investigated.

7.3.2 Combining Minimality Criteria

A very interesting question is the possibility to combine different kinds of minimality when several kinds of constraints and change are involved in the enforcement process. The underlying problem is to determine what kind of information are the most important for the agent, and consequently, which kind of minimal change must be applied first. As far as we know, only our works on revision of argumentation frameworks have considered the combination of different minimality criteria.

For instance, the revision approach described at Chapter 4 considers that the primitive information for an agent reasoning with an argumentation framework is its set of extensions. Minimal acceptability change is thus applied first. Two other kinds of minimality are then considered. The first one is minimal structural change, in terms of changes on the set of attacks. The second one comes from the nature of the output of our revision operators. We allow to obtain of set of revised argumentation frameworks. It seems natural, in this case, to regard minimal cardinality of the output as a desirable property. These two kinds of minimality are combined to define different families of revision operators. When the possibility to use auxiliary arguments in the result of the revision is considered, new possibilities are added to combine the minimality criteria.

The revision approach introduced in Chapter 5 combines structural change (additions and removals of attacks or arguments) and acceptability change (the constraints concern the fact that an argument is skeptically accepted or not). Minimal change on the structure of the graph and minimal change on the set of skeptically accepted arguments are considered, and combined through some weighted Hamming-like distances.

Of course, one can imagine some notion of minimal change for each kind of constraint described in the previous section. Combining them is always possible, as soon as the agent is able to decide which kind of information is more important for her, and so to decide which minimality criterion must be applied first. This question depends on the application.

7.3.3 Rationality Postulates for Constraint Enforcement

As we already explained, minimal change has been borrowed from belief change theory in logical settings. We have seen in Chapter 2 that this principle is not the only desirable property for belief change operations. It is usual for these applications to define a set of rationality postulates to be satisfied by any "good" operator. The AGM framework [AGM85] and its adaptation for propositional logic by Katsuno and Mendelzon [KM91] present rationality postulates for belief revision, and define some revision operators satisfying the postulates, while [KM92] gives a similar result for belief update.

Such an idea has been adapted by different authors for revision of argumentation frameworks. In Chapter 4, we adapt the postulates and revision operators from the KM framework to the setting of extension-based semantics. These postulates express the constraints on the acceptability of arguments as set-theoretical relations between the set of extensions of the argumentation framework. We have defined the family of operators which satisfy these postulates. We have also considered in Chapter 5 some revision operators dealing with structural and acceptability constraints, but this time a particular restriction on the definition of such a revision operator ensures that it satisfies another adaptation of the KM postulates. Some similar AGM-like family of rationality postulates has been described in [BKRvdT13, DHP14, BB15, DHL+15].

In [BCdSL13], the authors suppose the possibility to use any propositional language $\mathcal{L}$ able to represent information about an argumentation framework and its semantics, and they adapt the KM update postulates to enforce in an argumentation system any constraint which can be expressed in $\mathcal{L}$.

Until now, the only postulate-based approach for change in argumentation frameworks are borrowed from logical belief change frameworks. It is reasonable to suppose that some change operations are specific to argumentation scenarios, so an interesting research track is to define axiomatic characterization of such approaches. For instance, even if enforcement of an extension [BB10, Bau12, CMKMM15b] is very close to revision, it is not at all defined like our revision approaches. We can suppose that postulates for extension enforcement would be different from postulates for revision.

7.4 Towards Generalized Enforcement

Now, we propose an approach which is a first step towards the definition of a global family which gathers the existing approaches to enforce a constraint in an argumentation framework, and which also allows to encode some new kind of constraints.
Similarly to what we have done for belief revision (see Chapter 5) and extension enforcement (see Chapter 6), we use some logical encodings to represent constraint enforcement as a satisfaction problem.

7.4.1 Formal Setting

We suppose the existence of a finite set of arguments $\Omega = \{a_1, \dots, a_n\}$. We define a propositional language used to express the constraints to be enforced.

Definition 128 (Ω-formula)**.**
Let $F = \langle A, R \rangle$ be any argumentation framework built on the set of arguments $A \subseteq \Omega$. We suppose the existence of a fixed semantics σ.

- For any arguments $a_i, a_j \in \Omega$, the Boolean variable $att^F_{a_i,a_j}$ means that there is an attack from the argument a_i to the argument a_j in F.
- For any argument $a_i \in \Omega$, the Boolean variable $on^F_{a_i}$ means that the argument a_i belongs to A.
- For any argument $a_i \in \Omega$, the Boolean variable $sk^{F,\sigma}_{a_i}$ means that the argument a_i is skeptically accepted by F with respect to the semantics σ.
- For any argument $a_i \in \Omega$, the Boolean variable $cred^{F,\sigma}_{a_i}$ means that the argument a_i is credulously accepted by F with respect to the semantics σ.

- For any set of arguments $E \subseteq \Omega$, the Boolean variable $ext_E^{F,\sigma}$ means that E is a σ-extension of the considered argumentation framework.
- An Ω-formula is any propositional formula built on the set of variables $\{att_{a_i,a_j}^F \mid a_i, a_j \in \Omega\} \cup \{on_{a_i}^F \mid a_i \in \Omega\} \cup \{sk_{a_i}^{F,\sigma} \mid a_i \in \Omega\} \cup \{cred_{a_i}^{F,\sigma} \mid a_i \in \Omega\} \cup \{ext_E^{F,\sigma} \mid E \subseteq \Omega\}$ with the set of connectives $\{\vee, \wedge, \neg\}$.

For any Ω-formula φ, the satisfaction of φ by F, noted $F \mid\!\sim \varphi$, is defined by

- if $\varphi = on_{a_i}^F, a_i \in \Omega$, then $F \mid\!\sim \varphi$ if and only if $a_i \in A$;
- if $\varphi = att_{a_i,a_j}^F, a_i, a_j \in \Omega$, then $F \mid\!\sim \varphi$ if and only if $x, y \in A, (a_i, a_j) \in R$;
- if $\varphi = sk_{a_i}^{F,\sigma}, a_i \in \Omega$, then $F \mid\!\sim \varphi$ if and only if $a_i \in A$ and $a_i \in \bigcap Ext_\sigma(F)$;
- if $\varphi = cred_{a_i}^{F,\sigma}, a_i \in \Omega$, then $F \mid\!\sim \varphi$ if and only if $a_i \in A$ and $a_i \in \bigcup Ext_\sigma(F)$;
- if $\varphi = ext_E^{F,\sigma}, E \subseteq \Omega$, then $F \mid\!\sim \varphi$ if and only if $E \subseteq A$ and $E \in Ext_\sigma(F)$;
- if $\varphi = \varphi_1 \wedge \varphi_2$, then $F \mid\!\sim \varphi$ if and only if $F \mid\!\sim \varphi_1$ and $F \mid\!\sim \varphi_2$;
- if $\varphi = \varphi_1 \vee \varphi_2$, then $F \mid\!\sim \varphi$ if and only if $F \mid\!\sim \varphi_1$ or $F \mid\!\sim \varphi_2$;
- if $\varphi = \neg \varphi_1$, then $F \mid\!\sim \varphi$ if and only if $F \not\mid\!\sim \varphi_1$.

Now, we define formally a constraint enforcement operator, which is an operation which changes an argumentation framework to obtain a result satisfying a given constraint.

Definition 129 (Constraint Enforcement Operator)**.**
A *constraint enforcement operator* $\oplus$ is a mapping from an argumentation framework F and a Ω-formula φ to a set of argumentation frameworks such that:

$$F \oplus \varphi \subseteq \{F' \mid F' \mid\!\sim \varphi\}$$

This definition is quite abstract, but we show in the next section that this family of operators includes several existing approaches for the enforcement of a constraint in an argumentation framework.

7.4.2 Propositional Encoding of Constraint Enforcement Operators

Now we sketch a method to encode any constraint enforcement operator as a satisfaction problem. Similarly to the encoding of translation-based revision described in Chapter 5 and to the encoding of extension enforcement presented in Chapter 6, we consider the formula Φ_F^σ such that every model of it corresponds to a σ-extension of F. For instance, if σ is the stable semantics, we use the encoding from [BD04]:

$$\Phi_F^{st} = \bigwedge_{a_i \in A} (a_i \Leftrightarrow \bigwedge_{a_j : (a_j, a_i) \in R} \neg a_j)$$

which leads to the definition of

$$\Phi_A^{st} = \bigwedge_{a_i \in A} (a_i \Leftrightarrow \bigwedge_{a_j \in A} att_{(a_j,a_i)} \Rightarrow \neg a_j)$$

Now, we reproduce the scheme we used to define translation-based revision and extension enforcement in the previous chapters, and we define the stable theory of A adapted to constraint enforcement:

$$\bigwedge_{a_i \in A}[sk^{\sigma}_{a_i} \Leftrightarrow (\forall a_1, \ldots, a_n, \Phi^{\sigma}_A \Rightarrow a_i)]$$
$$\bigwedge_{a_i \in A}[cred^{\sigma}_{a_i} \Leftrightarrow (\exists a_1, \ldots, a_n, \Phi^{\sigma}_A \wedge a_i)]$$
$$\bigwedge_{E \subseteq \Omega}[ext^{\sigma}_E \Leftrightarrow (\exists a_1, \ldots, a_n, \Phi^{\sigma}_A \wedge (\bigwedge_{a_i \in E} a_i) \wedge (\bigwedge_{a_i \in A \setminus E} \neg a_i))]$$

We recognize the part of the encoding which concerns skeptical acceptance, which is equivalent to the stable theory of A for translation-based revision, and the part concerning extensions which is related to the encoding of extension enforcement operators.
Similarly to what we have done for translation-based revision, we associate a formula with a set of argumentation frameworks *via* the functions $Proj_{att}$ and arg introduced in Chapter 5. We can now define any constraint enforcement operator via the propositional encoding:

$$F \oplus \varphi \subseteq arg(Proj_{att}(\varphi \wedge th_{\sigma}(\Omega))$$

with arg and $Proj_{att}$ as they have been defined for previous approches.

Of course, existing works can be encoded through this kind of constraint enforcement operators, with the accurate restriction on the formulae, and some particular constraints on the selected argumentation frameworks (or similarly, on the set of models of $\varphi \wedge th_{\sigma}(\Omega)$ which must be decoded by arg and $Proj_{att}$). By the construction of the propositional language, our translation-based revision operators and our extension enforcement operators can be written as constraint enforcement through this logical encoding. The update operators by [DHP14] can also be represented easily, since they use variables to represent acceptance of an argument, and variables to represent the attacks, which are incorporated in this new language.

7.5 Conclusion

This chapter comes from a collaboration with Sylvie Doutre and Laurent Perussel. It is an extended and updated version of a previous work [DP13]. In this chapter, we consider some recent contributions on dynamics of argumentation systems, and we present other enforcement cases that remain to be investigated.

Table 7.1 and 7.2 sum up the existing approaches in the topic of change in argumentation systems. We recall, for each of the listed contribution, which kind of constraints are considered, which kind of change is applied to enforce them, and when it is relevant, which notion of quality of enforcement is used. The table only mentions the changes which are applied to enforce a given constraint (structural and/or semantic), and not the changes that this enforcement may imply. In particular, acceptability changes are a usual side effect of structural and semantic constraints, but they are not considered as a first-class citizen of the change operation.

The study shows that many challenging enforcement problems of interest remain to be explored in abstract argumentation. In particular, semantic change has received far less attention than other kinds of change. More than the question of enforcing a semantic constraint, a challenging problem is to enforce an acceptability constraint by a semantic change only; studying the quality of semantic change is then a mean to choose the best option among several possible semantic changes. A last relevant perspective for future work with Dung's framework is the combination of constraints and changes. Some combinations of constraints and "quality of change" criteria have already been considered, but the rich level

Constraints	[CdSCLS10]	[DW11]	[BB10]	[Bau12]	[BCdSL13]	[BKRvdT13]	[KBM$^+$13]
Structural	✓				✓		
Semantic		✓	✓	✓			
Acceptability			✓	✓	✓	✓	✓
Change	[CdSCLS10]	[DW11]	[BB10]	[Bau12]	[BCdSL13]	[BKRvdT13]	[KBM$^+$13]
Structural	✓	✓	✓	✓	✓	✓	✓
Semantic		✓	✓	✓			
Quality	[CdSCLS10]	[DW11]	[BB10]	[Bau12]	[BCdSL13]	[BKRvdT13]	[KBM$^+$13]
Structural				✓		✓	
Semantic							
Acceptability							
Postulates					✓	✓	

Table 7.1: Summary of Existing Approaches of Change in Argumentation (2010–2013)

Constraints	[Rie14]	[CMKMM14b]	[DDLW14]	[DHP14]	[CMKMM14c]	[NW14]	[CMKMM15b]	[BB15]	[DHL$^+$15]
Structural				✓	✓	✓		✓	✓
Semantic			✓						
Acceptability	✓	✓	✓	✓	✓	✓	✓		✓
Change	[Rie14]	[CMKMM14b]	[DDLW14]	[DHP14]	[CMKMM14c]	[NW14]	[CMKMM15b]	[BB15]	[DHL$^+$15]
Structural	✓	✓	✓	✓	✓	✓	✓	✓	✓
Semantic			✓						
Quality	[Rie14]	[CMKMM14b]	[DDLW14]	[DHP14]	[CMKMM14c]	[NW14]	[CMKMM15b]	[BB15]	[DHL$^+$15]
Structural		✓		✓	✓		✓		
Semantic									
Acceptability				✓				✓	
Postulates		✓		✓	✓	✓		✓	✓

Table 7.2: Summary of Existing Approaches of Change in Argumentation (2014–2015)

of expressiveness of argumentation systems allows to consider many other ones. Our ultimate aim is a more general representation of constraints and changes in Dung's framework, sketched in this chapter, which would allow to express each possible kind of enforcement. In particular, the use of our Boolean encoding to process constraint enforcement through satisfaction and optimization problems, similarly to our translation-based revision and extension enforcement approaches, is a challenging issue.

This study has mainly been conducted with Dung's abstract argumentation framework. However, as we have exemplified, the problems typology that has been set in this chapter may apply to other argumentation frameworks as well. Adapting existing work about change in Dung's framework to other argumentation frameworks is not necessarily straightforward, and this is a stimulating question for future work.

A last interesting research track is the question of structured argumentation frameworks. In such frameworks, two levels of constraints and changes may occur: either it is observed on the underlying structure of arguments, and it may have some effects on the argument graph, or on the opposite, the agent may enforce a constraint at the graph level, which impacts the underlying structure of arguments. As far as we know, none of these ideas has been considered in the literature, although it may be particularly useful, for instance when argumentation frameworks are used to reason with inconsistent belief bases [BH01, BH08].

Conclusion

It's the end. . . ... but the moment has been prepared for. . .

The Doctor – *Doctor Who - Logopolis*

Our work began as a Master thesis [Mai12], in which we studied the possibility to use AGM belief revision in the setting of argumentation. The contribution of this master thesis was only an introduction to what have been done in this Ph.D. thesis, but it led to several interesting research questions that we have deepened. First, we have stated that several ways to adapt AGM postulates were possible; we can consider that the important information convoyed by an argumentation framework is one of these types: the extensions of the argumentation frameworks, the corresponding labellings, or the set of skeptically accepted arguments. One of the main differences between these settings is the meaning of the negation. When we consider skeptically accepted arguments as the outcome of the argumentation framework, $\neg a_i$ can be interpreted as "a_i is not a skeptical consequence of the argumentation framework", or said otherwise, there is at least one extension which does not contain a_i. When extensions are considered, $\neg a_i$ means that a_i does not belong to any extension. These are exactly the meanings of the revision formulae that are considered in this thesis. For both of them, we proved that some revision operators satisfy our postulates. We have also presented a version of the postulates suited to labellings, slightly different from the one reported in the Master thesis.

This explains why, in this thesis, we have been firstly interested in revision of argumentation frameworks, which is the subject of our two first contributions. We have studied different ways to use AGM theory with argumentation frameworks, which are particularly interesting to offer an agent different ways to incorporate a new piece of information in their argumentation framework. These revision operators offer some possibilities which were not existing in the litterature when we started our study. In particular, when a agent receives a new piece of information, concerning arguments acceptance statuses, but she does not have some new arguments at her disposal that could explain this change of statuses. Then, we have also presented some extensions of these revision operators to tackle more possible cases, in particular, the integration of an integrity constraint in the revision process, and an open world version which allows the incorporation of new arguments.

One of the differences between our revision approaches is that the second one does not directly deal with the components of the argumentation frameworks, but performs a translation into propositional logic which allows to use propositional revision operators to obtain the expected result. We have been interested in the application of such a logical translation to tackle other scenarios of dynamics of argumentation frameworks. In particular, our work about extension enforcement has led to the developement of some pieces of software which use satisfaction and optimization algorithms to compute the result. This kind of approach seems very promising for argumentation issues, since constraint programming provides often some efficient approaches to solve highly complex problems.

The last part of our contribution is an opening to numerous challenging future works. Indeed, we have proposed a classification of constraints to be enforced in an argumentation framework, and of the different kinds of changes to apply in the argumentation frameworks to satisfy these constraints. We have seen how the existing works, including our own ones, fit in this classification, and we have identified some interesting kinds of change which have not been considered so far. We also want to consider dynamics scenarios in some other argumentation settings, notably the different extensions of Dung's argumentation frameworks.

As we have explained, the use of logical encodings to tackle argumentation issues is particularly interesting. This has led to the implementation of a quite efficient software to perform extension enforcement. We have also been part of another project which has the same idea at its origin: CoQuiAAS [LLM15a, LLM15c, LLM15b]. The developement of this software and its participation to the First International Competition on Computational Models of Argumentation [TV15] come from our objective to provide to the community some efficient software solutions to argumentation reasoning problems. We think that the study of logical encodings of argumentation semantics is a key step for several interesting issues. In this thesis, we have described some methods which take advantage of propositional encodings to perform some reasoning tasks on argumentation frameworks. Our approaches are completely generic, since they are based on the existence of a propositional formula whose models exactly correspond to the extensions of the argumentation framework for the chosen semantics. Then, the same scheme can be used, from the considered propositional encoding, to define our revision and enforcement operators. But this kind of direct polysize propositional encoding of the semantics is not possible with any semantics, since some of them (for instance, the preferred semantics) have a complexity higher than NP. To be able to compute our translation-based revision or enforcement operators, we need to consider either a propositional encoding which will be exponential in the size of the argumentation framework (while the current encodings have a polynomial size), or to choose a new logical setting to represent the semantics. For instance, the Quantified Boolean Formulae setting is a good candidate [EW06]. This kind of encodings is not only useful for revision and enforcement issues, but it can also be a method to check the realizability of a set of candidates [DDLW14], and moreover to associate an argumentation framework with this set of candidates.

Many other interesting future works are envisioned. First, as already explained in the related chapters, the kinds of change that we have characterized with rationality postulates are only a small part of the types of change that can occur in an argumentation framework. In particular, belief revision based on minimal change of the attack relation have not been characterized through the properties that it should satisfy. Also, we want to deepen our understanding of the different enforcement scenarios, the extension enforcement, and more generally the constraint enforcement that we sketched in the last chapter. We also want to formalize the change scenarios that are described in the last chapter, in particular the change of a semantics to reach a goal about acceptability, which is an interesting alternative way to tackle problems similar to the ones that we have studied in this thesis.

The translation-based approach to solve argumentation related issues can be performed through other settings. For instance, we think that establishing a link between the dynamics of argumentation frameworks and planning is a promising approach. If we suppose that each state is an argumentation framework, representing the agent's beliefs, and that the transitions between states are the possible elementary operation to perform on an argumentation framework (adding or removing arguments and attacks), changing an argumentation framework to satisfy some constraint (such that a revision formula or an extension enforcement) is equivalent to find a plan from the current state to a state which satisfies the goal. Finding an optimal plan would guarantee a form of minimal change on the structure of the graph.

Intuitively, this is related to some dialogue protocols such that these defined in [KBM+13]. It is also interesting to determine whether this kind of approach would allow to define new kinds of operators, or would be a rewriting of existing ones.

A very interesting related topic developped in the recent years is the merging of argumentation frameworks [CMDK+07, TBS08, BM11, GR12, DMW12, DKV15], which is very useful if we want to represent the beliefs of a group of agents, each of them using its own argumentation framework. In particular, the question (already present for revision of argumentation frameworks) to decide if the main piece of information is the structure of the framework, or the statuses of arguments, is really important and may lead to the definition of some very different merging approaches. It is well-known that in the logical framework, belief merging [KP98, KP99, Kon00] is connected to belief revision. In particular, [KP99] defines constrained belief merging, which is an extension of AGM belief revision. So the study of the merging of argumentation frameworks is an interesting and natural extension of our works about revision of argumentation frameworks, which would lead to the definition of merging operators focusing on the arguments statuses. Similarly, we know that there is a link between belief revision and contraction in the AGM framework. We think that adapting contraction to argumentation setting is also interesting. Indeed, the revision approach allows an agent to incorporate an information such that "this argument should be accepted by every extension" or "this argument should be rejected by every extension". But it is not possible here to incorporate a piece of information such that "this argument should be neither skeptically accepted nor skeptically rejected", which would be the case with a contraction operator. Since propositional belief contraction has been defined [CKM15], as a follow up of Katsuno and Mendelzon's work on belief revision, we think that an adaptation of belief contraction, similar to our first work on belief revision, will be a promising approach to tackle this scenario.

Appendix

Appendix A

Background Notions

Pure mathematics can not lie!

The Doctor – *Doctor Who - The Claws of Axos*

This first appendix presents the basic mathematical notions which are used in this thesis. Most of these ones do not present any real difficulty of understanding, but it is the opportunity to fix some notations which will appear in this document. It can also be useful to a reader which is not familiar with some of the mathematical concepts which are used in this thesis. So, this chapter introduces basic notions about sets and binary relations, before presenting propositional logic. Without any in-depth analysis, we give some definitions about graphs. We are also interested in complexity theory: we introduce the notions required to understand the complexity classes from the polynomial hierarchy, and then we present some well-known problems which are used in the thesis and we recall their complexity.

Contents

A.1 Sets and Relations

We call a *set* a collection of objects, without any precise order, in which an object cannot appear more than once. In this document, we will usually use capital letter to denote a set (for instance, A, E or S), sometimes written in calligraphy (like $\mathcal{C}$). The set E containing the objects x and y is denoted $E = \{x, y\}$. There exist some relations and operations on sets. Among them, the most usual ones are:

- $x \in E$ means that the object x belongs to the set E.
- $E_1 \subseteq E_2$ is equivalent to $\forall x \in E_1, x \in E_2$, meaning that each object which belongs to E_1 also belongs to E_2. E_1 may be equal to E_2 in this case. The strict counterpart is denoted $\subset$.
- $E_1 \cap E_2 = \{x \mid x \in E_1 \text{ and } x \in E_2\}$ is the intersection of E_1 and E_2, that is the set of objects which are in E_1 and E_2.
- $E_1 \cup E_2 = \{x \mid x \in E_1 \text{ or } x \in E_2\}$ is the union of E_1 and E_2, that is the set of objects which are either in E_1 or in E_2, or both at once.
- $E_1 \backslash E_2 = \{x \mid x \in E_1 \text{ and } x \notin E_2\}$ is the difference between E_1 and E_2, that is the set of objects which appear in E_1 but not in E_2.
- $E_1 \Delta E_2 = (E_1 \backslash E_2) \cup (E_2 \backslash E_1)$ is the symmetrical difference between E_1 and E_2, that is the set of objects which belong to E_1 but do not belong to E_2, and vice-versa.
- $E_1 \times E_2 = \{(x, y) \mid x \in E_1 \text{ and } y \in E_2\}$ is the Cartesian product of E_1 and E_2, that is the set of pairs with the first member is an element from E_1, and the second member is an element from E_2.

A (binary) relation R between two sets E_1 and E_2 is a subset of the Cartesian product $E_1 \times E_2$. $x \in E_1$ is in relation with $y \in E_2$ by R if and only $(x, y) \in R$. A relation on the set E is a subset of $E \times E$. Such a relation is:

- reflexive if $\forall x \in E, (x, x) \in R$;
- transitive if $\forall x, y, z \in E$, if $(x, y) \in R$ and $(y, z) \in R$, then $(x, z) \in R$;
- symmetrical if $\forall x, y \in E$, if $(x, y) \in R$ then $(y, x) \in R$;
- anti-symmetrical if $\forall x, y \in E$, if $(x, y) \in R$ and $(y, x) \in R$ then $x = y$;
- total if $\forall x, y \in E$, either $(x, y) \in R$, or $(y, x) \in R$, partial otherwise.

Then, a binary relation is called

- an equivalence relation if it is reflexive, transitive and symmetrical;
- an order relation if it is reflexive, transitive and anti-symmetrical;
- a pre-order if it is reflexive and transitive.

Given an order (or a pre-order) $\leq$ on a set E, we define its strict counterpart $<$ by the binary relation on E such that, $\forall x, y \in E$, $x < y$ if $x \leq y$ and $y \not\leq x$. The equivalence relation resulting from $\leq$ is denoted by $=$, defined as the binary relation on E such that, $\forall x, y \in E$, $x = y$ if and only if $x \leq y$ and $y \leq x$. Moreover, for each subset E' of E, we define the set of minimal elements of E' with respect to $\leq$ by $\min(E', \leq) = \{x \in E' \mid \nexists x' \in E' \text{ such that } x' < x\}$. Similarly, the maximal elements of E' with respect to $\leq$ are $\max(E', \leq) = \{x \in E' \mid \nexists x' \in E' \text{ such that } x < x'\}$.

Example 37.
Let E be the set $\{John, Paul, George, Ringo\}$.

- The relation R_1 on E defined by $(x, y) \in R_1$ if and only if x was born the same year than y is an equivalence relation.

- The relation R_2 on E defined by $(x, y) \in R_2$ if and only if x is taller than y is an order relation.

So, as Ringo was born on july 7^{th} 1940 and Paul on june 18^{th} 1942, $(Ringo, Paul) \notin R_1$. But as John was born on october 9^{th} 1940, $(Ringo, John) \in R_1$.
We know that John and Paul are both 1.80 m tall, so the pairs $(John, Paul)$ and $(Paul, John)$ both belong to R_2, together with $(Paul, George)$, since George is 1.77 m tall.

Given a set of elements E, it is useful for many applications to be able to quantify how two elements of E are different. This can be done through a notion of (pseudo-)distance.

Definition 130.
A *pseudo-distance* d over E is a mapping from each pair of elements in $E \times E$ to a non-negative real number in $\mathbb{R}$, such that

- $d(e_1, e_2) = 0$ if and only if $e_1 = e_2$;

- $d(e_1, e_2) = d(e_2, e_1)$.

Moreover, d is a *distance* if it satisfies

- $d(e_1, e_2) + d(e_2, e_3) \geq d(e_1, e_3)$.

A.2 Propositional Logic

Now, let us introduce some basic notions on propositional logic. It is one of the simplest fragments of logic. The basic piece of information on which is built a propositional language is a set of *Boolean variables* called the vocabulary: $V = \{x_1, \ldots, x_n\}$. The x_i variables can take two different values, $true$ or $false$. These values are usually noted respectively 1 and 0.

A propositional formula is built on a vocabulary V and a set of connectives used to link the variables, for instance $C = \{\wedge, \neg\}$, with $\wedge$ meaning the *conjunction* (the "and": $x \wedge y$ means that x is $true$ and y is $true$) and $\neg$ meaning the *negation* (the "not": $\neg x$ means that x is $false$). We can use these connectives to define some other ones:

- disjunction ("or") $x \vee y$ can be defined as $\neg(\neg x \wedge \neg y)$, meaning that either x is $true$, or y is $true$ (or both at the same time);

- exclusive or $x \oplus y$ can be defined as $(x \vee y) \wedge (\neg x \vee \neg y)$, meaning that either x is $true$, or y is $true$, but not both at the same time;

- material implication $x \Rightarrow y$ can be defined as $\neg x \vee y$, meaning that if x is $true$, then y is also $true$;

- equivalence $x \Leftrightarrow y$ can be defined as $(x \Rightarrow y) \wedge (y \Rightarrow x)$, meaning that x and y have the same truth value.

These connectives apply to formulae, constants also noted $true$ and $false$ and variables (which are atomic formulae). In this thesis, we usually use Greek letters to denote propositional formulae (φ, μ for instance).

Once the syntax of formulae is established, it is required to define the associated semantics to determine the truth value of a formula φ, given the truth values of the variables belonging to φ. We call an *interpretation* any valuation of an element of $\{true, false\}$ to each variable from the vocabulary used to build the language;[18] such an interpretation is a mapping $\omega : V \longrightarrow \{true, false\}$. We use $\omega(x)$ to represent the truth value associated to the Boolean variable x. We can represent ω as the set $\{x \mid x \in V, \omega(x) = true\} \cup \{\neg x \mid x \in V, \omega(x) = false\}$, or even with the more compact representation $\{x \mid x \in V, \omega(x) = true\}$ meaning that the variables which do not belong to the set receive the value $false$.
Truth values of formulae are defined in a recursive way from the values of their subformulae. For instance, truth tables of the usual connectives are given in Table A.1 and Table A.2.

x	$\neg x$
$false$	$true$
$true$	$false$

Table A.1: Truth Table of the Negation

x	y	$x \wedge y$	$x \vee y$	$x \oplus y$	$x \Rightarrow y$	$x \Leftrightarrow y$
$false$	$false$	$false$	$false$	$false$	$true$	$true$
$false$	$true$	$false$	$true$	$true$	$true$	$false$
$true$	$false$	$false$	$true$	$true$	$false$	$false$
$true$	$true$	$true$	$true$	$false$	$true$	$true$

Table A.2: Truth Table of the Binary Connectives

- We call a *model* of the formula φ any interpretation ω which *satisfies* φ, meaning the the truth value associated to φ is $true$. We note $\omega \models \varphi$. In the other case, ω is a *counter-model* of φ.
- A formula which has at least one model is *satisfiable* (or *consistent*), in the other case it is *unsatisfiable* (or *inconsistent*).
- A formula φ which is satisfied by every interpretation is said to be *valid*, noted $\models \varphi$.
- Let φ and ψ be two formulae. φ *implies* ψ, noted $\varphi \vdash \psi$, if each model of φ is a model of ψ. They are equivalent, noted $\varphi \equiv \psi$, if and only if they have the same set of models.
- Two formulae φ and ψ are equi-satisfiable if and only if φ is satisfiable implies that ψ is also satisfiable, and vice-versa.

We note $\mathrm{Mod}(\varphi)$ the set of models of the propositional formula φ.

Example 38.
Let $V = \{r, s, u\}$ (for *rain*, *sun* and *umbrella*) be a set of Boolean variables. The formula $\varphi = (r \oplus$

[18] We can restrict this valuation to the variables belonging to the formula.

$s) \wedge (r \Rightarrow u)$ represents the information about weather meaning that it may rain, or the weather may be sunny, but not both at the same time; and if it is raining, one must take an umbrella.
This formula is consistent, for instance $\omega = \{r, u\}$ is a model. $\omega' = \{r, s\}$ is a counter-model.

Interpretation	r	s	u	$r \oplus s$	$r \Rightarrow u$	φ
ω	*true*	*false*	*true*	*true*	*true*	*true*
ω'	*true*	*true*	*false*	*false*	*false*	*false*

Table A.3: Truth Table of formula φ for the interpretations ω and ω'

A notion of size of a formula is useful for the definition of some concepts.

Definition 131.
Given a propositional formula φ, the size of the formula φ is defined as the number of symbols (variables and connectives) required to write φ.

For instance, the size of $\varphi_1 = a_1 \wedge a_2$ is 3, while the size of $\varphi_2 = x \vee (y \wedge z)$ is 5.

It is possible to normalize the syntax of propositional formulae, applying some transformation rules to ensure that the syntax of the resulting formula satisfies some constraints, without changing the semantics of the formula (so, without changing the set of models).

Let us introduce two of the most usual normal forms:

Definition 132.

- A formula is a clause if it is written as $x_1 \vee x_2 \vee \cdots \vee x_n$ (with x_i some Boolean variables).
- A formula is a term if it is written as $x_1 \wedge x_2 \wedge \cdots \wedge x_n$ (with x_i some Boolean variables).
- A formula is written under *conjunctive normal form* (CNF) if it is a conjunction of clauses.
- A formula is written under *disjunctive normal form* (DNF) if it is a disjunction of terms.

Every propositional formula can be rewritten as a CNF or DNF formula. The transformation can be particularly hard to compute (we explain this with more details in Section A.4).

Proposition 45.
For each propositional formula φ, Algorithm 1 returns a propositional formula φ' such that $\varphi' \equiv \varphi$ and φ' is a CNF formula.

Let us mention that the first method, described in Algorithm 1, requires exponential space in the size of the formula. If n is the size of the formula φ, the equivalent CNF formula obtained by Algorithm 1 may have its size which is exponential in n. We can use another method, with a polynomial size increase when the objective is to preserve only equisatisfiability (and not the set of models):

Proposition 46 (Tseitin Transformation [Tse68])**.**
For each propositional formulae φ_1 and φ_2 built on the vocabulary V,

1. *$\varphi_1 \wedge \varphi_2$ can be replaced $(\neg\varphi_1 \vee \neg\varphi_2 \vee \alpha) \wedge (\varphi_1 \vee \neg\alpha) \wedge (\varphi_2 \vee \neg\alpha)$ with $\alpha \notin V$;*

Algorithm 1: naiveCNFTransformation

Input: A propositional formula φ
Output: A CNF formula φ' such that $\varphi' \equiv \varphi$
if *(φ is a variable) or (φ is a constant)* **then**
  return φ
else if $\varphi = \varphi_1 \wedge \varphi_2$ **then**
  `/* naiveCNFTransformation(`φ_1`) is equal to` $\varphi_1^1 \wedge \varphi_1^2 \wedge \ldots \varphi_1^n$ `*/`
  `/* naiveCNFTransformation(`φ_2`) is equal to` $\varphi_2^1 \wedge \varphi_2^2 \wedge \ldots \varphi_2^m$ `*/`
  return $\varphi_1^1 \wedge \varphi_1^2 \wedge \ldots \varphi_1^n \wedge \varphi_2^1 \wedge \varphi_2^2 \wedge \ldots \varphi_2^m$
else if $\varphi = \varphi_1 \vee \varphi_2$ **then**
  `/* naiveCNFTransformation(`φ_1`) gives` $\varphi_1^1 \wedge \varphi_1^2 \wedge \ldots \varphi_1^n$ `*/`
  `/* naiveCNFTransformation(`φ_2`) gives` $\varphi_2^1 \wedge \varphi_2^2 \wedge \ldots \varphi_2^m$ `*/`
  return $(\varphi_1^1 \vee \varphi_2^1) \wedge (\varphi_1^2 \vee \varphi_2^1) \wedge \ldots (\varphi_1^n \vee \varphi_2^1) \wedge \cdots \wedge (\varphi_1^1 \vee \varphi_2^m) \wedge (\varphi_1^2 \vee \varphi_2^m) \wedge \ldots (\varphi_1^n \vee \varphi_2^m)$
else if $\varphi = \neg\varphi'$ **then**
  if $\varphi = \neg x$ **then**
    return $\neg x$
  else if $\varphi = \neg(\neg\varphi_1))$ **then**
    return naiveCNFTransformation(φ_1)
  else if $\varphi = \neg(\varphi_1 \vee \varphi_2)$ **then**
    return naiveCNFTransformation($\neg\varphi_1 \wedge \neg\varphi_2$)
  else if $\varphi = \neg(\varphi_1 \wedge \varphi_2)$ **then**
    return naiveCNFTransformation($\neg\varphi_1 \vee \neg\varphi_2$)
else if $\varphi = (\varphi_1 \Rightarrow \varphi_2)$ **then**
  return naiveCNFTransformation($\neg\varphi_1 \vee \varphi_2$)
else if $\varphi = (\varphi_1 \Leftrightarrow \varphi_2)$ **then**
  return naiveCNFTransformation($(\varphi_1 \wedge \varphi_2) \vee (\neg\varphi_1 \wedge \neg\varphi_2)$)
else if $\varphi = (\varphi_1 \oplus \varphi_2)$ **then**
  return naiveCNFTransformation($(\varphi_1 \wedge \neg\varphi_2) \vee (\neg\varphi_1 \wedge \varphi_2)$)

2. *$\varphi_1 \vee \varphi_2$ can be replaced by $(\varphi_1 \vee \varphi_2 \vee \neg\alpha) \wedge (\neg\varphi_1 \vee \alpha) \wedge (\neg\varphi_1 \vee \alpha)$ with $\alpha \notin V$;*

3. *$\neg\varphi_1$ can be replaced by $(\varphi_1 \vee \alpha) \wedge (\neg\varphi_1 \vee \neg\alpha)$ with $\alpha \notin V$.*

The recursive applications of the rules above lead to a CNF formula which is equi-satisfiable to the input fomula φ. For each model m' of the resulting formula φ', a projection of it on V gives a model m of φ.

At last, to conclude this presentation of propositional logic, let us address the notion of knowledge base (or belief base). It is usual to represent the pieces of knowledge (or beliefs) of an agent by a set of logical formulae $K = \{\varphi_1, \ldots, \varphi_n\}$ called a belief base. This kind of set is usually interpreted in a conjunctive way, meaning that $K \equiv (\varphi_1 \wedge \cdots \wedge \varphi_n)$. Depending on the applications, it is possible to reason with a knowledge base exactly in the same way as a "simple" propositional formula. Moreover, we are restricted here to the case of propositional knowledge and belief bases, but it is of course possible to reason with other kinds of logics. A belief base is called a theory when it is closed with respect to the consequence relation.

Definition 133 (Cn-Theory)**.**
A consequence operator *à la* Tarski [Tar30] Cn is such that, for every belief bases K, K':

- $K \subseteq Cn(K)$;
- $Cn(K) = Cn(Cn(K))$;
- If $K \subseteq K'$, then $Cn(K) \subseteq Cn(K')$.

The belief base K is a Cn-theory if and only if it is deductively closed for Cn, meaning $K = Cn(K)$.

In the usual case, Cn is defined by the consequence relation is $Cn(K) = \{\varphi \mid K \vdash \varphi\}$, with $\vdash$ the classical inference relation that we defined previously. So we use the notation $K \vdash \varphi$ to mean that the formula φ is a consequence of the belief base K.
$K_\perp$ denotes the trivial belief base, meaning the base containing the set of all formulae in the language used by the agent: $K_\perp = Cn(\{\perp\})$.

A.3 Graph Notions

Let us continue this presentation of background notions with a section about graph theory. A graph is a structure composed of two kinds of data. The first kind is a (finite) set of elements called *nodes*, linked by the second kind of data: *edges*. These edges can be directed from one node to another one, or they can be bi-directed. We distinguish two kinds of graphs associated to both kind of links:

- A *non-directed graph* is a pair $G = \langle N, E \rangle$ with N the set of nodes, and $E \subseteq \{\{x, y\} | x, y \in N\}$ the set of edges.
- A *directed graph* (also called *digraph*) is a pair $G = \langle N, E \rangle$ with N the set of nodes, and $E \subseteq N \times N$ the set of edges.

In this thesis, we are essentially interested in digraphs.

Example 39.
Let $N = \{l, m, n, p, t\}$ (for Lens, Marseille, Nancy, Paris, Toulouse) be a set of cities.

- NS associated with the set of non-directed edges $E_1 = \{\{l, p\}, \{p, n\}, \{p, m\}, \{l, m\}, \{m, t\}, \{n, m\}, \{l, n\}\}$ generates the non-oriented graph G_1 given on Figure A.1.

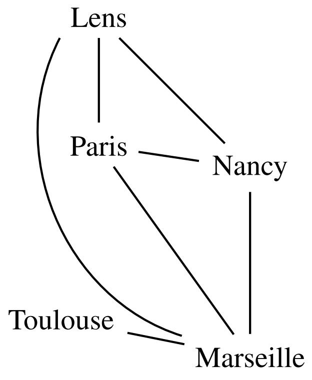


Figure A.1: The Non Oriented Graph G_1

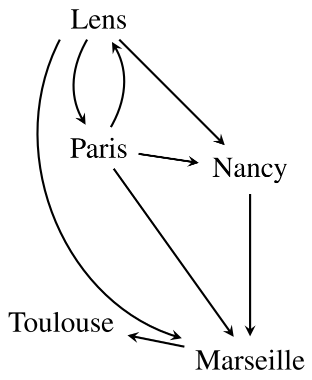


Figure A.2: The Digraph G_2

- N associated with the set of oriented edges $E_2 = \{(l,p),(p,l),(p,n),(p,m),(l,m),(m,t),(n,m),(l,n)\}$ generates the digraph G_2 given on Figure A.2.

The structure of a graph influence the processing which is done with it. Some particular patterns in a graph structure have been defined; in particular, these ones are suited to digraphs:

- A path $(x_1,\dots,x_n)$ is a sequence of nodes $x_1,\dots,x_n \in S$ such that for each $i \in \{1,\dots,n-1\}$, $(x_i,x_{i+1}) \in E$. n is the *length* of the path.
- We call the node x_i an ascendant (respectively a descendant) of the node x_j if there exists a path from x_i to x_j (respectively from x_j to x_i).
- A cycle is a path such that the first node is equal to the last one.
- A loop is a cycle of length 1, that is an edge from a node x to itself.

Example 40.
Let G be the digraph described on Figure A.3. (e,a,b) is a path of length 3 in the digraph G. (a,b,c,d) is a cycle of length 4, and (c,c) is a loop. e is an ascendant of each other node, while c is a descendant of each node (including itself).

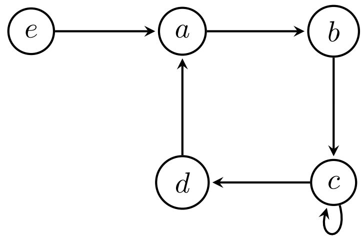


Figure A.3: The Digraph G

A.4 Computational Complexity

When a program performs a task, two kinds of resources are used: time (required to compute the result of the problem) and space (for instance, the number of variables required to represent and solve the problem). Power of modern computers, fitted with processors which frequency usually exceeds several GHz and memory which exceeds several Go, seems to make useless saving of time and space resources. It is true that some simple problems can be solved even with naive algorithms, but it is not the case for every interesting problem. In particular, most of problems in artificial intelligence, including the ones we study in this thesis, are concerned with the resolution of computationally hard problems. So we need to use some more sophisticated technics to be sure to solve our problems with reasonable time and space resource.
Complexity theory allows to classify and compare problems depending on the hardness to solve them. This section aims at presenting the basic of this theory, and the complexity of some well-known problems, which are used in this thesis.

A.4.1 Turing Machines and Decidability

One of the seminal works on complexity theory is a mathematical model described in 1936 by Alan Turing [Tur36], who called it *computing machine*: it is an abstract model able to compute any "calculable" decimal number, meaning any number which can be computing using a finite quantity of resources. Although this Turing machine was described years before the apparition of computers, it is a quite faithful abstraction of our modern machines: it is a tape composed of a sequence of squares, which is browsed by a reader/writer (resembling to the memory of a computer), altogether with a transition function (representing the processor of a computer).
Each algorithm can be associated to a Turing machine, and vice-versa. So studying Turing machines properties is equivalent to study the properties of algorithms. First, let us briefly present Turing's model:

Definition 134.
A Turing machine is a 7-tuple $\langle Q, \Gamma, B, \Sigma, q_0, \delta, F \rangle$ with :

- $Q = \{q_0, q_1, \ldots, q_m\}$, a finite set of *states*;
- Γ, the finite set of symbols used by the machine (*machine vocabulary*);
- $B \in \Gamma$, a particular symbol called *blank*;
- Σ, the *input vocabulary*;
- $q_0 \in Q$, the *initial state* of the machine;
- $\delta : Q \times \Sigma \longrightarrow Q \times \Gamma \times \{L, R\}$, the *transition function* ;

- $F \subseteq Q$, the set of *final states*.

The behaviour of the Turing machine is as follows: starting at the initial state q_0 with the symbol in the first square of the tape, the transition function chooses the next state of the machine, the symbol to write in the current square, and the direction in which the reader/writer must move. The machine repeats this behaviour, with the new state and the symbol which is read by the reader/writter, until it reaches a final state. Then, the outcome of the machine (and so, the result of the algorithm) is the word written on the tape.

Example 41.
Let us describe a Turing machine which multiplies any integer number (represented in binary notation) by 2. We recall that this multiplication is done just by adding a 0 at the end of the representation of the number. This very simple algorithm can be described by the machine $\mathcal{M} = \langle Q, \Gamma, \sqcup, \Sigma, q_0, \delta, F\rangle$ defined by:

- $Q = \{q_0, q_1\}$
- $\Gamma = \{0, 1, \sqcup\}$
- $\Sigma = \{0, 1\}$
- $F = \{q_1\}$
- δ described on Table A.4.

Current state	Current symbol	Next state	Symbol to write	Direction of the move
q_0	0	q_0	0	R
q_0	1	q_0	1	R
q_0	$\sqcup$	q_1	0	R
q_1	STOP			

Table A.4: The Transition Function δ for the Turing Machine $\mathcal{M}$

So, if 42 is the input of the machine (represented under its binary notation 101010), the behaviour of the machine is this one:

- The machine reads the symbol 1 at the first square of the tape, and applies the rule associated with the configuration $(q_0, 1)$ in the transition function, which re-writes the symbol 1 in the square, moves the reader/writer to the right, and stays in the state q_0.
- The machine reads the symbol 0 at the second square of the tape, and applies the rule associated with $(q_0, 0)$: re-write 0 in the square, stay in state q_0, and move the reader/writer to the right.
- The previous steps are repeated twice.
- At the 7^{th} square of the tape, the reader/writer reads the symbol $\sqcup$. The rule to apply is to write 0 in the square, to move the reader/writer to the right, and to change the state from q_0 to q_1, which is the finale state.

The Turing machine output is the sequence of symbols 1010100, which corresponds to 84.

Turing's model, although it is very simple, is expressive enough to represent any existing algorithm. And so, for instance, a programming language is called Turing-complete if it is expressive enough to represent any Turing machine.

More that studying the expressiveness of a programming language, we use Turing machine to classify problems according to their hardness. First, let us introduce different kinds of problems that are useful in this thesis.

Definition 135 (Decision Problem, Function Problem)**.**
A *decision problem* $\mathcal{P}$ over a set of input data E is a mapping from any element in E to a value in $\{true, false\}$ (or equivalently, {YES, NO}) .
A *function problem* $\mathcal{P}$ over a set of input data E is a mapping from any element in E to a single outcome.

The outcome of a decision problem is an answer YES or NO to a question, while the outcome of a function problem is supposed to be more complex than YES or NO. For instance, the problem of determining if an equation has a solution is a decision problem, it is not required to exhibit a solution, only to prove that there exists at least one. Then, the problem of determining a solution of the same equation is a function problem. We can also refine the notion of function problem.

Definition 136 (Enumeration Problem, Optimization Problem)**.**
The *enumeration problem* ENUM-$\mathcal{P}$ associated with the function problem $\mathcal{P}$ is a mapping from any element $e_i \in E$ to the set of all the outcomes of $\mathcal{P}$ over e_i.
The *optimization problem* OPT-$\mathcal{P}$ associated with the function problem $\mathcal{P}$ is a mapping from any element $e_i \in E$ to a single outcome of $\mathcal{P}$ over e_i which minimizes (or maximizes) a given criterion.

Keeping on the example of solving an equation, the enumeration problem is "Give all the solutions of the equation", while a possible optimization problem is "Give the smallest solution of the equation" (with respect to the natural order over the numbers).
In the rest of this chapter, we present essentially the complexity of decision problems.

A first distinction between decision problems is their *decidability*. It has been proved by Turing that it is not possible to define a machine $\mathcal{M}$ which takes as input a second machine $\mathcal{M}'$ and a data e, and which answers YES if $\mathcal{M}'$ can reach a finale state from the input e, and NO in the other case. This problem is called HALTING, it is the most well-known undecidable problem.

A decidable problem is defined by:

Definition 137.
Let $\mathcal{P}$ be a problem defined on a set of data $E = \{e_1, \dots, e_n\}$. $\mathcal{P}$ is decidable if there exists a Turing machine $\mathcal{M}$ such that for each $e_i \in E$, $\mathcal{M}$ reaches a final state when it is executed on the input e_i, and gives a solution to the problem $\mathcal{P}$ for the input e_i.

A.4.2 Determinism, Hardness, Completeness and Polynomial Hierarchy

The class of decidable problems can be refined, still with the use of Turing machines. Indeed, depending on the properties of the machine used to solve a problem, the hardness of this problem is not the same.

Definition 138.
Let $\mathcal{M}$ be a Turing machine with δ its transition function. $\mathcal{M}$ is *deterministic* if and only if δ is a mapping from any configuration of the machine (q', x', m) to a single image. Otherwise, $\mathcal{M}$ is said to be *non-deterministic*.

Roughly speaking, the simplest problems are those which can be solved by a deterministic Turing machine without needing a high number of steps. Then, the complexity of harder problems can be expressed with their need to use non-deterministic machines. This principle leads to the classification of some of the decidable problems into the different classes of the *polynomial hierarchy* [GJ79, Pap94].
Let us first present the model of resources (time and space) used. The resolution time of a problem $\mathcal{P}$ with the input e by a Turing machine $\mathcal{M}$ can be interpreted as the number of steps required for $\mathcal{M}$ to reach a final state when it starts from e. We consider each step as a time unit. In a similar way, Turing machine can be used to determine the use of space required to solve $\mathcal{P}$. It is the number of squares of the tape used to obtain the result. Let us notice that this space complexity can be used as a lower bound of time complexity: we know that at least k steps are required to browse k squares of the tape.
For both of these measures, we define the complexity of the problem by a mapping $f(n)$, where n is the size of the input data e (for instance, when the input is a binary number, n can be the number of bits used to represent e), and f is the mapping which returns the number of steps (respectively squares) required to obtain the result of the Turing machine processing, and the notation $\mathcal{O}(f(n))$ means, intuitively, that the problem can be solved in the worst case in $f(n)$ steps (respectively, with $f(n)$ squares of the tape), and maybe less. More formally, a mapping g satisfies $g(n) \in \mathcal{O}(f(n))$ if and only if there exists a pair of finite constants (c, n_0) such that:

$$\forall n \geq n_0, g(n) \leq c \times f(n)$$

Then we define formally time complexity and space complexity of a problem:

Definition 139.
A problem $\mathcal{P}$ belongs to the complexity class TIME$(f(n))$ if and only if there exists a deterministic Turing machine which solves $\mathcal{P}$ in a number of steps $\mathcal{O}(f(n))$ for input data of size n.

Definition 140.
A problem $\mathcal{P}$ belongs to the complexity class SPACE$(f(n))$ if and only if there exists a deterministic Turing machine which solves $\mathcal{P}$ with a number of squares of the tape $\mathcal{O}(f(n))$ for input data of size n.

These time and space complexity have been used to define usual complexity classes. The less difficult problems are those which belong to the class TIME$(f(n))$ with f a polynomial. This class of problems is called P [GJ79, Pap94].

Definition 141.
The class P is the set of all the decision problems which can be solved by a deterministic Turing machine in polynomial time in the size of the input, meaning that P is equivalent to $\bigcup_{k=0}^{+\infty}$TIME(n^k).

This class contains "simple" problems. For instance, given a list l of integers, sorting l in increasing order can be done in polynomial time by a deterministic Turing machine. This is not the case of every problem, and in particular most of interesting problems studied in Artificial Intelligence cannot be solved in polynomial time with a deterministic Turing machine. So we need to define complexity classes for harder problems than P problems.

Definition 142.
The class NP is the set of all the decision problems that can be solved by a non-deterministic Turing machine in polynomial time in the size of the input.

From the definition, each problem in P also belongs to NP, since a deterministic Turing machine can be seen as a particular non-deterministic Turing machine, so P $\subseteq$ NP. The usual conjecture says that the converse inclusion is false, but it is still an open question.

A problem in NP is decidable, because it can be solved by a deterministic Turing machine in time $\mathcal{O}(a^{f(n)})$, with f a polynomial and a a constant strictly greater than 1. Moreover, given a possible solution s of a problem $\mathcal{P} \in$ NP, and an input e, it is possible to determine in polynomial time whether s is a solution of $\mathcal{P}$ for the input e.
Given a complexity class C, it is possible to define its complement:

Definition 143.
The problem CO$\mathcal{P}$ is called the *complement* of problem $\mathcal{P}$ if, for each input e, $\mathcal{P}$ answers YES if and only if CO$\mathcal{P}$ does not.
The class COC is the complement class of C, it contains all the complement problems of problems from C.

It is known that P $=$ COP but it is conjectured that this is not the case for NP and its complement class CONP. The ground of the hierarchy of complexity classes, called *polynomial hierarchy*, is:

- P $\subseteq$ NP
- P $\subseteq$ CONP

It is possible to define classes of decidable problems with a greater complexity than NP and CONP. For this purpose, let us introduce an extension of the Turing machines model.

Definition 144.
A Turing machine (deterministic or not) with an oracle of class C is a Turing machine able to compute in a single step the answer of a problem in the class C.

These Turing machines with oracles are used to define complexity classes customized by the complexity of the oracle.

Definition 145.
Let C_1 and C_2 be two complexity classes. The complexity class $C_1^{C_2}$ contains all the problems which can be solved by a Turing machine from the class C_1 with an oracle from the class C_2.

These customized complexity classes are used to define the polynomial hierarchy.

Definition 146.
The polynomial hierarchy [Sto76] is the set of complexity classes defined recursively by:

- $\Delta_0^P = \Sigma_0^P = \Pi_0^P =$P
- $\Delta_{k+1}^P = \mathrm{P}^{\Sigma_k^P}$
- $\Sigma_{k+1}^P = \mathrm{NP}^{\Sigma_k^P}$
- $\Pi_{k+1}^P =$COΣ_{k+1}^P

The first levels of these hierarchy is illustrated on figure A.4. $C_1 \rightarrow C_2$ means that class C_1 is included in class C_2.
Determining that two problems belong to a same complexity class from the hierarchy is not enough to determine which one is harder to solve, or to determine if their hardness is equivalent. Indeed, if $\mathcal{P}_1$ and $\mathcal{P}_2$ both belong to NP, it is possible that one of them belongs to P but not the other one, which is consequently harder to solve than the first one. We can use polynomial-time reduction to determine the relative complexity of two problems.

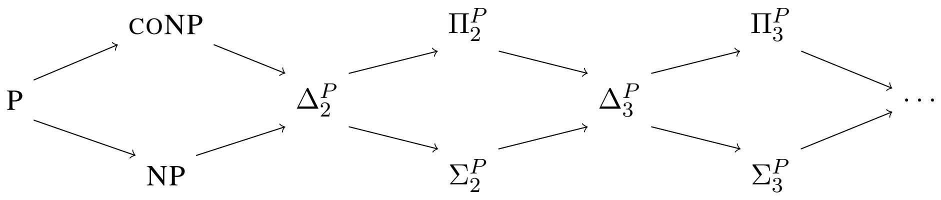


Figure A.4: Schematic Representation of the Polynomial Hierarchy

Definition 147.
Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two problems, and φ an algorithm the input of which is an instance of $\mathcal{P}_1$ and the output of which is an instance of $\mathcal{P}_2$. φ is a polynomial-time reduction from $\mathcal{P}_1$ to $\mathcal{P}_2$ if and only if:

- φ can be computed in polynomial-time in the size of its input;
- for each instance I of $\mathcal{P}_1$, $\mathcal{P}_1(I) = YES$ if and only if $\mathcal{P}_2(\varphi(I)) = YES$.

If there is a polynomial-time reduction from a problem $\mathcal{P}_1$ to a problem $\mathcal{P}_2$, then $\mathcal{P}_2$ is at least as hard to solve as $\mathcal{P}_1$. We use this definition to determine if the complexity of a problem is at least equal to the complexity of a class in the polynomial hierarchy.

Definition 148.
The problem $\mathcal{P}$ is C-hard if and only if there exists a polynomial-time reduction from each problem $\mathcal{P}' \in$ C to $\mathcal{P}$.

This means that the problem $\mathcal{P}$ is at least as hard to solve than any problem in C, and it may be harder. We can guarantee the precise complexity of a problem if it is C-hard and it belongs to C.

Definition 149.
The problem $\mathcal{P}$ is C-complete if and only if it is C-hard and it belongs C.

Even if it is still an open question, it is usually admitted that P $\neq$ NP. Consequently, a problem in the class P is supposed to be strictly less hard to solve than a NP-complete complete. In the other case, there would be a collapse of the polynomial hierarchy at its ground: each problem in the polynomial hierarchy would be solvable in polynomial-time. Even if it would be possible to distinguish between them thanks to the degree of the associated polynomial, it would let us consider a pratical improvement for solving the reasoning tasks associated with Artificial Intelligence.
Let us still remark that this result would have a negative effect in some other domains: in cryptography, for instance, the robustness of some approaches to ensure the security of data and information transfers (like online payment) comes from the pratical difficulty to solve NP-hard problems.

A particular class of problems can be defined from NP and coNP. To define it, we need to introduce the notion of intersection of two problems.

Definition 150.
The problem P is the intersection of the problems P_1 and P_2, denoted by $P_1 \cap P_2$, if and only if P_1 and P_2 are defined on the same set of inputs, and for each input e, $P(E) = YES$ if and only if $P_1(e) = YES$ and $P_2(e) = YES$.
The complexity class DP is the set of problems which are the intersection of a problem $P_1 \in$ NP and a problem $P_2 \in$ coNP.

A DP-complete problem is at least as hard to solve as a NP-complete problem or a coNP-complete problem.

We will finish this introduction to complexity of decision problems with the presentation of the last complexity class useful for a good understanding of this thesis. Until now, we focused on time complexity. However, problems can be classified as well with respect to their space complexity.

Definition 151.
The complexity class PSPACE is the set of decision problems which can be solved by a deterministic Turing machine with a space $\mathcal{O}(f(n))$ for each input of size n, with f a polynomial.

It is well-known that some problems in PSPACE are not particularly easy to solve. Indeed, NP $\subseteq$ PSPACE. For practical applications, PSPACE-complete problems are usually more difficult than NP-complete ones.

A.4.3 Complexity of Function Problems

By definition, function, optimization and enumeration problems are not decision problems, and their complexity cannot be evaluated thanks to the polynomial hierarchy. However, we see easily that the time and space resources required to solve a decision problem give lower bounds to the resources required to solve function, optimization and enumeration problems. Indeed, if it is possible to compute a solution of a problem (or an optimal solution, or every solution) in a time (respectively space) $\mathcal{O}(f(n))$, then it is possible to decide if a solution exists in $\mathcal{O}(f(n))$: applying the function, optimization or optimization algorithm proves enough.

Since the practical approaches to solve decision problems usually give a solution of the problem (and not only YES or NO), we do not present the complexity classes corresponding to function problems. In this thesis, we< either refer to the complexity of the related decision problems, or give a lower bound of the complexity, when considering function problems.

A.4.4 Complexity of Well-Known Problems: Constraint Satisfaction and Optimization

Boolean Constraint Satisfaction

To finish this chapter, let us introduce some well-known problems, along with their complexity. These problems are useful in the rest of this thesis. The first one, which is the most famous, is the problem of Boolean Satisfiability (SAT). This problem consists in checking, for a propositional formula, if it possesses at least one model. Most of solving approaches are focused on its variant CNF-SAT, which is the restriction of the SAT problem to CNF formulae, given that every propositional formula can be rewritten in conjunctive normal form (see Section A.2).

SAT is the first problem proved to be NP-complete [Coo71]. It is possible to prove that a problem $\mathcal{P}$ is NP-hard pointing out a polynomial-time reduction from SAT to $\mathcal{P}$. The simplicity to formulate the SAT problem, and the expressiveness of propositional logic, explain the interest of Artificial Intelligence researchers for this topic. During the recent decades, the power of SAT solvers has significantly increased. Indeed, modern technics for SAT solving allow to consider instances with more and more variables and constraints. We can overview the current state of SAT practical approaches in the result of the international SAT competition [BDHJ14], organised as a satellite event of the *International Conference on Theory and Applications of Satisfiability Testing*. Results of the last edition can be obtained online: http://www.satcompetition.org/2014/results.shtml.

We explained previously that transforming any propositional formula into an equivalent DNF formula is generally hard to compute. It is related to the complexity of the SAT problem: any DNF formula is satisfiable as soon as one of its term is satisfiable, which can be checked in polynomial time. Transforming any formula into an equivalent DNF formula requires exponential space in the worst case.

Pseudo-Boolean Constraint Satisfaction and Optimization

The problem of Pseudo-Boolean (PB) constraint satisfaction PB-SAT is an extension of SAT. Here, Boolean variables can be linked by arithmetic constraints. For instance, a possible constraint is to expect a sum of variables defined on the domain $\{0, 1\}$ to be lesser or equal to a given value.
Any CNF formula can be re-written as a Pseudo-Boolean formula. Indeed, let $x_1 \vee x_2 \vee \cdots \vee x_n$ be a clause in the considered CNF formula, it can be written simply as the PB constraint $x_1+x_2+\cdots+x_n \geq 1$, meaning that the constraint is satisfied if and only if at least one of the variables x_i is not equal to 0, which is equivalent to expect x_i to be $true$. So the clause is satisfied.

This proves that PB-SAT is NP-hard, since each SAT instance can be translated into a PB-SAT instance. It is in fact well-known that PB-SAT is NP-complete [Kar72].

A variant of PB-SAT, noted here PB-OPT, is the problem of optimizing a set of Pseudo-Boolean constraint with respect to some objective function. Formally, it consists in adding to the Pseudo-Boolean constraint a function $f(X)$, with X a subset of the problem variables, such that the expected solutions are those which minimize the value of $f(X)$ (or maximize, which is equivalent to minimize $-f(x)$).

Example 42.
With the vocabulary $V = \{x_1, x_2, x_3, x_4\}$, the propositional formula $\varphi = (x_1 \oplus x_2) \wedge (x_3 \vee x_4)$ can be translated into a Pseudo-Boolean formula in this way:

$$x_1 + x_2 = 1$$
$$x_3 + x_4 \geq 1$$

So, obtaining a model of φ is equivalent to obtaining a solution of this set of Pseudo-Boolean constraints. For instance, $S_1 : x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1$ is a solution of the PB-SAT problem corresponding to the SAT problem on φ.
Moreover, if we want to minimize the cardinality of the solution, we have to add the linear objective function

$$\min f(x_1, x_2, x_3, x_4) = x_1 + x_2 + x_3 + x_4$$

In this case, the solution S_1 is not optimal, because the value associated with this solution is 3, while $S_2 : x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 0$ satisfies the constraints and minimizes the objective function.

The complexity of PB-OPT is higher than the complexity of PB-SAT: given a real number k, determining if k is the optimal value of the objective function for a given instance of ths problem is DP-complete, and computing the optimal solution itself is FP^{NP}-complete[19]. We can explain this intuitively: each PB-SAT instance is a particular PB-OPT instance with the objective function $f(V) = 0$; this explains that solving a set of Pseudo-Boolean constraints with respect to an objective function is at least as hard as solving it without an objective function.

[19] FP^{NP} is the class of all function problems which can be solved with a polynomial number of calls to an oracle in NP.

Quantified Boolean Constraint Satisfaction

In this last part dedicated to the complexity of some well-known problems, we introduce another extension of SAT: QBF-SAT. It concerns satisfaction of Quantified Boolean Formulae. Such a formula is a propositional formula enriched with some universal (∀) and existential (∃) quantifiers on the variables.
For instance, $\forall x_1, \exists x_2, x_1 \Rightarrow x_2$ is the quantified formula meaning that for each truth value assigned to x_1, it is possible to assign x_2 a truth value such that $x_1 \Rightarrow x_2$ is $true$.
More generally, we define a QBF by $Q_1 x_1, Q_2 x_2, \ldots, Q_n x_n, \varphi$, with $Q_i \in \{\forall, \exists\}$ for each value of i, and φ a propositional formula built on the vocabulary $\{x_1, \ldots, x_n\}$.

We can show quite simply that QBF-SAT is NP-hard. Indeed, there is an obvious polynomial-time reduction from SAT to QBF-SAT: for each propositional formula φ built on the vocabulary $\{x_1, \ldots, x_n\}$, we can build a QBF $\exists x_1, \ldots, \exists x_n, \varphi$, which is true if and only if φ is satisfiable.

The opposite is not true. It is possible to associated a propositional formula to each QBF, but this mapping is not always possible in polynomial-time. The corresponding algorithm is simple. It is a recursive application of the following rewritting rules, stopping when all the quantifiers have disappeared:

1. $\exists x_i, \varphi \rightsquigarrow subst(\{x_i \mapsto false\}, \varphi) \vee subst(\{x_i \mapsto true\}, \varphi)$
2. $\forall x_i, \varphi \rightsquigarrow subst(\{x_i \mapsto false\}, \varphi) \wedge subst(\{x_i \mapsto true\}, \varphi)$

with $subst(\{x_i \mapsto V\}, \varphi)$ the propositional formula obtained when the truth value $V \in \{true, false\}$ is assigned to the variable x_i in φ.
For instance, $subst(\{x_1 \mapsto true\}, (x_1 \vee x_2) \wedge (\neg x_1 \vee x_3))$ is the propositional formula $(true \vee x_2) \wedge (false \vee x_3)$, which is equivalent to x_3.
In the worst case (depending on the structure of φ and the quantifiers in the formula), this quantifier elimination technique may result in a propositional formula exponentially greater than φ, which explains that solving the QBF-SAT problem is harder than solving the SAT problem: it is in fact known as the canonical PSPACE-complete problem [GJ79].

Appendix B

Proofs of the Results from Chapter 4

Proposition 25 (Representation Theorem)**.**
*Given a semantics σ, a revision operator $\star$ satisfies the rationality postulates **(AE1)** - **(AE6)** if and only if there exists a faithful assignment which maps every framework $F = \langle A, R \rangle$ to a total pre-order $\leq_F^\sigma$ so that*

$$Ext_\sigma(F \star \varphi) = \min(\mathcal{A}_\varphi^\sigma, \leq_F^\sigma).$$

Proof. Given a semantics σ and a faithful assignment which matches AF to the pre-order $\leq_{AF}$, let $\star$ be a revision operator such thatt $Ext_\sigma(AF \star \varphi) = \min(A_\varphi^\sigma, \leq_{AF})$. We want to show that $\star$ satisfies the postulates.

(AE1) $Ext_\sigma(AF \star \varphi) \subseteq A_\varphi^\sigma$.
The definition of the operator is enough to show that this postulate is satisfied, because $\min(A_\varphi^\sigma, \leq_{AF}) \subseteq A_\varphi^\sigma$.

(AE2) If $Ext_\sigma(AF) \cap A_\varphi^\sigma \neq \emptyset$ then $Ext_\sigma(AF \star \varphi) = Ext_\sigma(AF) \cap A_\varphi^\sigma$.

Suppose that $Ext_\sigma(AF) \cap A_\varphi^\sigma \neq \emptyset$.
We want to show the inclusion $Ext_\sigma(AF) \cap A_\varphi^\sigma \subseteq Ext_\sigma(AF \star \varphi)$. Let $\varepsilon \in Ext_\sigma(AF) \cap A_\varphi^\sigma$. We know that $\varepsilon \in A_\varphi^\sigma$. Moreover, since $\varepsilon \in Ext_\sigma(AF)$, $\forall \varepsilon' \in A_\varphi^\sigma$, either

$$\varepsilon <_{AF} \varepsilon' \text{ if } \varepsilon' \notin Ext_\sigma(AF)$$

or

$$\varepsilon \approx_{AF} \varepsilon' \text{ if } \varepsilon' \in Ext_\sigma(AF)$$

(from the definition of faithful assignment).
So $\forall \varepsilon' \in A_\varphi^\sigma$, $\varepsilon \leq_{AF} \varepsilon'$, therefore $\varepsilon \in \min(A_\varphi^\sigma, \leq_{AF}) = Ext_\sigma(AF \star \varphi)$.

Now we want to show the converse inclusion.
Let $\varepsilon \in Ext_\sigma(AF \star \varphi)$. $\varepsilon \in A_\varphi^\sigma$ since $\star$ satisfies **(AE1)**.
Using *reductio ad absurdum*, let us suppose that $\varepsilon \notin Ext_\sigma(AF)$. Since we proved that $Ext_\sigma(AF) \cap A_\varphi^\sigma \subseteq Ext_\sigma(AF \star \varphi)$, and since this intersection is non-empty, $\exists \varepsilon' \in Ext_\sigma(AF \star \varphi)$ such that $\varepsilon' \in Ext_\sigma(AF) \cap A_\varphi^\sigma$, and in particular $\varepsilon' \in Ext_\sigma(AF)$. Consequently, $\varepsilon' <_{AF} \varepsilon$ (from the definition of a faithful assignment), which is in contradiction with the fact that ε is a minimal element in A_φ^σ with respect to $\leq_{AF}$.

So, $\varepsilon \in Ext_\sigma(AF) \cap A^\sigma_\varphi$, which holds $\forall \varepsilon \in Ext_\sigma(AF \star \varphi)$, so $Ext_\sigma(AF \star \varphi) \subseteq Ext_\sigma(AF) \cap A^\sigma_\varphi$.

The postulate is satisfied.

(AE3) If φ is σ-consistent, then $Ext_\sigma(AF \star \varphi) \neq \emptyset$.
From the definition of the operator, this postulate is satisfied. Indeed, if the formula is σ-consistent, then A^σ_φ is a finite non-empty set. There exists in this finite set some minimal elements with respect to $\leq_{AF}$, *i.e.* $Ext_\sigma(AF \star \varphi) \neq \emptyset$.

(AE4) If $\varphi \equiv_\sigma \psi$, then $Ext_\sigma(AF \star \varphi) = Ext_\sigma(AF \star \psi)$.
This postulate is satisfied from the definition of the operator.

$$\begin{aligned} Ext_\sigma(AF \star \varphi) &= \min(\mathcal{A}^\sigma_\varphi, \leq_{AF}) \\ &= \min(\mathcal{A}^\sigma_\psi, \leq_{AF}) \quad \text{from } \mathcal{A}^\sigma_\varphi = \mathcal{A}^\sigma_\psi \\ &= Ext_\sigma(AF \star \psi) \end{aligned}$$

(AE5) $Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi \subseteq Ext_\sigma(AF \star \varphi \wedge \psi)$.
If $Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi = \emptyset$, the postulate is satisfied. So we suppose now that $Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi \neq \emptyset$.

We first prove the inclusion $Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi \subseteq Ext_\sigma(AF \star \varphi \wedge \psi)$. Using *reductio ad absurdum*, suppose that $\exists \varepsilon \in Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi$ such that $\varepsilon \notin Ext_\sigma(AF \star \varphi \wedge \psi)$. We can rephraze it in this way:

$$\varepsilon \in \min(A^\sigma_\varphi, \leq_{AF}) \cap \mathcal{A}^\sigma_\psi \text{ and } \varepsilon \notin \min(A_{\varphi \wedge \psi}, \leq_{AF})$$

From the first part, we deduce $\varepsilon \in \mathcal{A}_{\varphi \wedge \psi}$. However, ε is not a minimal element in this set with respect to $\leq_{AF}$. Consequently, $\exists \varepsilon' \in \mathcal{A}_{\varphi \wedge \psi}$ such that $\varepsilon' <_{AF} \varepsilon$. From the definition of $A_{\varphi \wedge \psi}$, $\varepsilon' \in \mathcal{A}^\sigma_\varphi$ holds. It is in contradiction with $\varepsilon \in \min(A^\sigma_\varphi, \leq_{AF})$.
So $Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi \subseteq Ext_\sigma(AF \star \varphi \wedge \psi)$.

(AE6) If $Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi \neq \emptyset$ then $Ext_\sigma(AF \star \varphi \wedge \psi) \subseteq Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi$.
If $Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi \neq \emptyset$, let us suppose $\exists \varepsilon \in Ext_\sigma(AF \star \varphi \wedge \psi)$ such that $\varepsilon \notin Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi$.
If $\varepsilon \in \min(A_{\varphi \wedge \psi}, \leq_{AF})$, then $\varepsilon \in \mathcal{A}_{\varphi \wedge \psi}$, and so $\varepsilon \in \mathcal{A}^\sigma_\psi$ holds.
From this and $\varepsilon \notin Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi$, we deduce $\varepsilon \notin Ext_\sigma(AF \star \varphi)$.
Since we suppose that the intersection is non-empty, $\exists \varepsilon' \in Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi$. In particular, ε' is a model of φ and ψ, *i.e.* $\varepsilon' \in \mathcal{A}_{\varphi \wedge \psi}$. From $\varepsilon \in Ext_\sigma(AF \star \varphi \wedge \psi) = \min(A_{\varphi \wedge \psi}, \leq_{AF})$ and $\leq_{AF}$ is a total relation, we get that $\varepsilon \leq_{AF} \varepsilon'$.
As $\varepsilon' \in Ext_\sigma(AF \star \varphi) = \min(A^\sigma_\varphi, \leq_{AF})$, we have $\varepsilon \in \min(A^\sigma_\varphi, \leq_{AF})$, and so we have a contradiction.
So $Ext_\sigma(AF \star \varphi \wedge \psi) \subseteq Ext_\sigma(AF \star \varphi) \cap \mathcal{A}^\sigma_\psi$ holds.

Now, let $\star$ be a revision operator satisfying postulates **(AE1)-(AE6)**. Given an argumentation system $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ and a semantics σ, we define a relation $\leq_{AF}$ such that

$$\forall \varepsilon_1, \varepsilon_2 \in \mathcal{P}(\mathcal{A}), \varepsilon_1 \leq_{AF} \varepsilon_2 \Leftrightarrow \begin{cases} \varepsilon_1 \in Ext_\sigma(AF) \\ \text{or} \\ \varepsilon_1 \in Ext_\sigma(AF \star \text{form}(\varepsilon_1, \varepsilon_2)) \end{cases}$$

where $\text{form}_\sigma(\varepsilon_1, \ldots, \varepsilon_n)$ is a formula such that $\mathcal{A}^\sigma_{\text{form}_\sigma(\varepsilon_1, \ldots, \varepsilon_n)} = \{\varepsilon_1, \ldots, \varepsilon_n\}$.

Let us study the properties of $\leq_{AF}$.
Let $\varepsilon_1, \varepsilon_2 \in \mathcal{P}(\mathcal{A})$. $\varphi = \text{form}_\sigma(\varepsilon_1, \varepsilon_2)$ admits at least one model, *i.e.* it is consistent. Since $\star$ satisfies **(AE3)**, $Ext_\sigma(AF \star \varphi) \neq \emptyset$. Moreover, from **(AE1)**, we deduce $Ext_\sigma(AF \star \varphi) \subseteq \mathcal{A}^\sigma_\varphi = \{\varepsilon_1, \varepsilon_2\}$. From there, $\varepsilon_1 \in Ext_\sigma(AF \star \varphi)$ or $\varepsilon_2 \in Ext_\sigma(AF \star \varphi)$, *i.e.* $\varepsilon_1 \leq_{AF} \varepsilon_2$ or $\varepsilon_2 \leq_{AF} \varepsilon_1$, so the relation is total. Moreover, if $\varepsilon_1 = \varepsilon_2$, $\varepsilon_1 \in Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_1))$ holds, *i.e.* $\varepsilon_1 \leq_{AF} \varepsilon_1$: the relation is reflexive. Finally, we want to prove that the relation is transitive. Let $\varepsilon_1, \varepsilon_2, \varepsilon_3 \in \mathcal{P}(\mathcal{A})$ such that $\varepsilon_1 \leq_{AF} \varepsilon_2$ and $\varepsilon_2 \leq_{AF} \varepsilon_3$. There are three possible cases:

1. $\varepsilon_1 \in Ext_\sigma(AF)$;
2. $\varepsilon_1 \notin Ext_\sigma(AF)$ and $\varepsilon_2 \in Ext_\sigma(AF)$;
3. $\varepsilon_1 \notin Ext_\sigma(AF)$ and $\varepsilon_2 \notin Ext_\sigma(AF)$.

Case 1 If $\varepsilon_1 \in Ext_\sigma(AF)$, then $\varepsilon_1 \leq_{AF} \varepsilon_3$ holds from the definition of the relation.

Case 2 If $\varepsilon_1 \notin Ext_\sigma(AF)$ and $\varepsilon_2 \in Ext_\sigma(AF)$, then $Ext_\sigma(AF) \cap \{\varepsilon_1, \varepsilon_2\} = \{\varepsilon_2\}$. From **(AE2)**, $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2)) = \{\varepsilon_2\}$ holds. Therefore, $\varepsilon_1 \not\leq_{AF} \varepsilon_2$ since neither $\varepsilon_1 \in Ext_\sigma(AF)$ nor $\varepsilon_1 \in Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2))$ hold. It is a contradiction: the second case is impossible.

Case 3 If $\varepsilon_1 \notin Ext_\sigma(AF)$ and $\varepsilon_2 \notin Ext_\sigma(AF)$, then since $\star$ satisfies **(AE1)** and **(AE3)**, $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3))$ is a non-empty subset of $\{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$. There are two possible alternatives:

1. $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3)) \cap \{\varepsilon_1, \varepsilon_2\} = \emptyset$, *i.e.* $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3)) = \{\varepsilon_3\}$;
2. $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3)) \cap \{\varepsilon_1, \varepsilon_2\} \neq \emptyset$.

(1) If $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3)) \cap \{\varepsilon_1, \varepsilon_2\} = \emptyset$, then $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3)) = \{\varepsilon_3\}$ (because from **(AE1)** and **(AE3)** it is a non-empty subset of $\{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$), and from **(AE5)** and **(AE6)**, $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3)) \cap \{\varepsilon_2, \varepsilon_3\} = Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_2, \varepsilon_3)) = \{\varepsilon_3\}$ which is in contradiction with the fact that $\varepsilon_2 \leq_{AF} \varepsilon_3$ and $\varepsilon_2 \notin Ext_\sigma(AF)$. So this case is impossible.

(2) Suppose that $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3)) \cap \{\varepsilon_1, \varepsilon_2\} \neq \emptyset$. Since $\varepsilon_1 \leq_{AF} \varepsilon_2$ and $\varepsilon_1 \notin Ext_\sigma(AF)$, $\varepsilon_1 \in Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2))$ holds. From **(AE5)** and **(AE6)**, $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3)) \cap \{\varepsilon_1, \varepsilon_2\} = Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2))$ holds. So $\varepsilon_1 \in Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3)) \cap \{\varepsilon_1, \varepsilon_2\}$, in particular $\varepsilon_1 \in Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3))$.
Similarly, since $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3)) \cap \{\varepsilon_1, \varepsilon_2\} \neq \emptyset$, from **(AE5)** and **(AE6)**, $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2, \varepsilon_3)) \cap \{\varepsilon_1, \varepsilon_3\} = Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_3))$ holds, and we deduce $\varepsilon_1 \in Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_3))$, *i.e.* $\varepsilon_1 \leq_{AF} \varepsilon_3$.

This concludes the proof that the relation is transitive. So $\leq_{AF}$ is a total pre-order. Let us now show that associating this a pre-order with an argumentation framework is a faithful assignment.

1. Given an argumentation system $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, let $\varepsilon_1, \varepsilon_2$ be two elements in $\mathcal{P}(\mathcal{A})$ such that $\varepsilon_1 \in Ext_\sigma(AF)$ and $\varepsilon_2 \in Ext_\sigma(AF)$. Since $\varepsilon_1 \in Ext_\sigma(AF)$, $\varepsilon_1 \leq_{AF} \varepsilon_2$. Likewise, $\varepsilon_2 \in Ext_\sigma(AF)$ implies $\varepsilon_2 \leq_{AF} \varepsilon_1$. Consequently, $\varepsilon_1 \simeq_{AF} \varepsilon_2$.
2. Given an argumentation system $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, let $\varepsilon_1, \varepsilon_2$ be two elements of $\mathcal{P}(\mathcal{A})$ such that $\varepsilon_1 \in Ext_\sigma(AF)$ and $\varepsilon_2 \notin Ext_\sigma(AF)$. Like previously, $\varepsilon_1 \leq_{AF} \varepsilon_2$ holds. Moreover, now

$\varepsilon_2 \notin Ext_\sigma(AF)$ holds. From this and **(AE2)**, $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon_1, \varepsilon_2)) = \{\varepsilon_1\}$ holds, so $\varepsilon_2 \not\leq_{AF} \varepsilon_1$. Therefore, $\varepsilon_1 <_{AF} \varepsilon_2$.

The last thing to prove is the equality $Ext_\sigma(AF \star \varphi) = \min(A^\sigma_\varphi, \leq_{AF})$. First, if φ is not consistent, from **(AE1)** we deduce $Ext_\sigma(AF \star \varphi) = \emptyset = \min(A^\sigma_\varphi, \leq_{AF})$. So we suppose now that φ is consistent, that is $\mathcal{A}^\sigma_\varphi \neq \emptyset$.

Let us prove first the inclusion $Ext_\sigma(AF \star \varphi) \subseteq \min(A^\sigma_\varphi, \leq_{AF})$. Using *reductio ad absurdum*: suppose that $\exists \varepsilon \in Ext_\sigma(AF \star \varphi)$ such that $\varepsilon \notin \min(A^\sigma_\varphi, \leq_{AF})$. From **(AE1)**, $\varepsilon \in \mathcal{A}^\sigma_\varphi$ holds, and since $\varepsilon \notin \min(A^\sigma_\varphi, \leq_{AF})$, $\exists \varepsilon' \in \mathcal{A}^\sigma_\varphi$ such that $\varepsilon' <_{AF} \varepsilon$. Two cases must be considered:

Case 1 $\varepsilon' \in Ext_\sigma(AF)$, as $\varepsilon' \in \mathcal{A}^\sigma_\varphi$, $Ext_\sigma(AF) \cap \mathcal{A}^\sigma_\varphi \neq \emptyset$, which implies from **(AE2)** $Ext_\sigma(AF \star \varphi) = Ext_\sigma(AF) \cap \mathcal{A}^\sigma_\varphi$.

$$\begin{aligned} \varepsilon \in Ext_\sigma(AF \star \varphi) &\Rightarrow \varepsilon \in Ext_\sigma(AF) \cap \mathcal{A}^\sigma_\varphi \\ &\Rightarrow \varepsilon \in Ext_\sigma(AF) \\ &\Rightarrow \varepsilon \leq_{AF} \varepsilon' \\ &\Rightarrow \text{ contradiction with } \varepsilon' <_{AF} \varepsilon \end{aligned}$$

Case 2 $Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon, \varepsilon')) = \{\varepsilon'\}$. Since ε and ε' are models of φ, $\varphi \wedge \text{form}_\sigma(\varepsilon, \varepsilon') \equiv \text{form}_\sigma(\varepsilon, \varepsilon')$. From **(AE5)**, $Ext_\sigma(AF \star \varphi) \cap \{\varepsilon, \varepsilon'\} \subseteq Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon, \varepsilon')) = \{\varepsilon'\}$, therefore $\varepsilon \notin Ext_\sigma(AF \star \varphi)$, which is a contradiction.

So $Ext_\sigma(AF \star \varphi) \subseteq \min(A^\sigma_\varphi, \leq_{AF})$ holds, let us now prove the converse inclusion. We still use *reductio ad absurdum*, suppose that $\exists \varepsilon \in \min(A^\sigma_\varphi, \leq_{AF})$ such that $\varepsilon \notin Ext_\sigma(AF \star \varphi)$.

Since φ is consistent, from **(AE3)**, $\exists \varepsilon' \in Ext_\sigma(AF \star \varphi)$, and from **(AE1)** $\varepsilon' \in \mathcal{A}^\sigma_\varphi$. Since ε and ε' are models of φ, $\text{form}_\sigma(\varepsilon, \varepsilon') \wedge \varphi = \text{form}_\sigma(\varepsilon, \varepsilon')$. From **(AE5)-(AE6)**, $Ext_\sigma(AF \star \varphi) \cap \{\varepsilon, \varepsilon'\} = Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon, \varepsilon')) = \{\varepsilon'\}$ since $\varepsilon \notin Ext_\sigma(AF \star \varphi)$, so $\varepsilon' \leq_{AF} \varepsilon$. On the other hand, ε is minimal in A^σ_φ with respect to $\leq_{AF}$, *i.e.* $\varepsilon \leq_{AF} \varepsilon'$. Since $\varepsilon \notin Ext_\sigma(AF \star \text{form}_\sigma(\varepsilon, \varepsilon')) = \{\varepsilon'\}$, $\varepsilon \in Ext_\sigma(AF)$ holds, and from **(AE2)** $\varepsilon \in Ext_\sigma(AF \star \varphi)$. It is a contradiction.

We can conclude that $Ext_\sigma(AF \star \varphi) = \min(A^\sigma_\varphi, \leq_{AF})$. □

Proposition 26.
*Let σ be any semantics. Any pseudo-distance-based revision operator $\star^d$ satisfies the rationality postulates **(AE1) - (AE6)**.*

Proof. We just have to prove that the application $AF \mapsto \leq^{\sigma,d}_{AF}$ is a faithful assignment.

1. Given c_1 and c_2 two extensions of AF, $d(c_1, Ext_\sigma(AF)) = d(c_2, Ext_\sigma(AF)) = 0$, so $c_1 \simeq^{\sigma,d}_{AF} c_2$.
2. Given c_1 an extension of AF and c_2 another candidate which is not an extension of AF, $d(c_1, Ext_\sigma(AF)) = 0$ and $d(c_2, Ext_\sigma(AF)) > 0$ so $c_1 <^{\sigma,d}_{AF} c_2$.

So $\leq^{\sigma,d}_{AF}$ is a faithful assignment. Then Proposition 25 concludes the proof. □

Proposition 27.
Let m, n, o be three integers. $d_{(m,n,o)}$ is a pseudo-distance.

Proof. Let us prove that $d_{(m,n,o)}$ satisfies the two conditions of a pseudo-distance.
Let L_1, L_2 be two labellings. First,

$$\begin{aligned} d_{(m,n,o)}(L_1, L_2) &= \sum_{a \in A} ad(L_1(a), L_2(a)) \\ &= \sum_{a \in A} ad(L_2(a), L_1(a)) \\ &= d_{(m,n,o)}(L_2, L_1) \end{aligned}$$

Then,

$$\begin{aligned} d_{(m,n,o)}(L_1, L_2) = 0 &\Leftrightarrow \sum_{a\in A} ad(L_1(a), L_2(a)) = 0 \\ &\Leftrightarrow \forall a \in A, ad(L_1(a), L_2(a)) = 0 \\ &\Leftrightarrow L_1 = L_2 \end{aligned}$$

□

Proposition 28.
*Let σ be any semantics. Any labelling-pseudo-distance-based revision operator $\star_{d^{\mathcal{L}}}$ satisfies the rationality postulates **(AE1) - (AE6)**.*

Proof. It is obvious that $\min(L_\varphi, \leq_{AF}^{\sigma,d_L})$ is a subset of L_φ, and so $E(\min(L_\varphi, \leq_{AF}^{\sigma,d_L})) \subseteq A_\varphi^\sigma$. Let us show that this set is composed of minimal elements of A_φ^σ with respect to a particular pseudo-distance-based pre-order on sets of arguments, which means that the operator can be represented with a faithful assignment on candidates, and then it satisfies the postulates **(AE1)-(AE6)**.

Given c_1, c_2 two candidates, let $d_c(c_1, c_2) = \min_{L_1\in Labs(c_1), L_2\in Labs(c_2)} d_{\mathcal{L}}(L_1, L_2)$.
$d_c(c_1, c_2) = 0$ iff $c_1 = c_2$ can be shown easily: $\min_{L_1\in Labs(c_1), L_2\in Labs(c_2)} d_{\mathcal{L}}(L_1, L_2) = 0$ means that there is at least one of the labellings corresponding to c_1 which also corresponds to c_2, *i.e.* a specific pair (L_1, L_2) such that $L_1 = L_2$, so $in(L_1) = in(L_2)$, which leads to $c_1 = c_2$. The converse obviously holds: if $c_1 = c_2$, then the corresponding sets of L_1 and L_2 are the same sets, so the pseudo-distance is 0.
Symmetry is obvious, because d_c is based on $d_{\mathcal{L}}$ which satisfies this property.
We extend the definition of d_c to the distance between a candidate and a set of candidates by stating $d_c(c, C) = \min_{c'\in C} d_c(c, c')$.

Let us prove that $E(\min(L_\varphi, \leq_{AF}^{\sigma,d_L})) = \min(A_\varphi^\sigma, \leq_{AF}^{\sigma,d_c})$, where $\leq_{AF}^{\sigma,d_c}$ is the total pre-order defined from the pseudo-distance d_c. We use *reductio ad absurdum*: suppose that $\exists c \notin E(\min(L_\varphi, \leq_{AF}^{\sigma,d_L}))$ a candidate such that $\forall c' \in E(\min(L_\varphi, \leq_{AF}^{\sigma,d_L}))$, $d_c(c, Ext_\sigma(AF)) < d_c(c', Ext_\sigma(AF))$. From the definition of d_c,

$$\min_{\varepsilon\in Ext_\sigma(AF)} d_c(c, \varepsilon) < \min_{\varepsilon\in Ext_\sigma(AF)} d_c(c', \varepsilon)$$

which leads to

$$\begin{aligned} \min_{\varepsilon\in Ext_\sigma(AF)} \quad &(\min_{L_1\in Labs(c), L_2\in Labs(\varepsilon)} d_{\mathcal{L}}(L_1, L_2)) \\ &< \min_{\varepsilon\in Ext_\sigma(AF)}(\min_{L_1\in Labs(c'), L_2\in Labs(\varepsilon)} d_{\mathcal{L}}(L_1, L_2)). \end{aligned}$$

It is in contradiction with $c' \in E(\min(L_\varphi, \leq_{AF}^{\sigma,d_L}))$. □

Proposition 33.
*The revision operator $\star$ satisfies **(AL1)-(AL6)** if and only if for every semantics σ there exists a faithful assignment which maps every argumentation framework F to a total pre-order $\leq_F^\sigma$ such that for every formula $\varphi \in \mathcal{L}_A^{Labs}$:*

$$Labs_\sigma(F \star \varphi) = \min(L_\varphi^\sigma, \leq_F^\sigma)$$

Proof. First, let us suppose the existence of a faithful assignment such that $Labs_\sigma(F\star\varphi)$=$\min(L_\varphi^\sigma, \leq_F^\sigma)$, and let us prove that $\star$ satisfies the postulates.

(AL1) $Labs_\sigma(F \star \varphi) \subseteq L_\varphi^\sigma$
This result is trivial, from the definition of the operator.

(AL2) If $Labs_\sigma(F) \cap L_\varphi^\sigma \neq \emptyset$, then $Labs_\sigma(F \star \varphi) = Labs_\sigma(F) \cap L_\varphi^\sigma$.
We define $M = Labs_\sigma(F) \cap L_\varphi^\sigma \neq \emptyset$. $\forall L \in M, L \in \min(L_\varphi^\sigma, \leq_F^\sigma)$ holds from the definition of a faithful

assignment, so $M \subseteq Labs_\sigma(F \star \varphi)$.
Now, let us prove the converse inclusion. Reasoning with *reductio ad absurdum*, suppose that there exists $L \in Labs_\sigma(F \star \varphi)$ such that $L \notin Labs_\sigma(F) \cap L^\sigma_\varphi$. From **(AL1)**, $L \in L^\sigma_\varphi$ holds, so $L \notin Labs_\sigma(F)$ also holds. Since $L \in \min(L^\sigma_\varphi, \leq^\sigma_F)$, $\forall L' \in L^\sigma_\varphi, L \leq^\sigma_F L'$. But from the definition of a faithful assignment, any labelling $L'' \in Labs_\sigma(F)$ satisfies $L'' <^\sigma_F L$. It is a contradiction, which leads to the second inclusion, and so $\star$ satisfies **(AL2)**.

(AL3) If φ is σ-consistent, then $Labs_\sigma(F \star \varphi) \neq \emptyset$.
Follows trivially the definitions of L^σ_φ and $\star$. Indeed, if φ is σ-consistent, then L^σ_φ is a finite non-empty set, and then it admits some minimal elements with respect to the total pre-order $\leq^\sigma_F$. This leads to the conclusion of the postulate.

(AL4) If $\varphi \equiv_\sigma \psi$, then $Labs_\sigma(F \star \varphi) = Labs_\sigma(F \star \psi)$.
Follows trivially from the definition of the operator. As $\varphi \equiv_\sigma \psi$, $L^\sigma_\varphi = L^\sigma_\psi$, and then

$$\begin{aligned} Labs_\sigma(F \star \varphi) &= \min(L^\sigma_\varphi, \leq^\sigma_F) \\ &= \min(L^\sigma_\psi, \leq^\sigma_F) \\ &= Labs_\sigma(F \star \psi) \end{aligned}$$

(AL5) $Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi \subseteq Labs_\sigma(F \star \varphi \wedge \psi)$.
If $Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi = \emptyset$, then the postulate is satisfied. So we suppose now that $Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi \neq \emptyset$.

Let us prove the inclusion $Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi \subseteq Labs_\sigma(F \star \varphi \wedge \psi)$. Using *reductio ad absurdum*, suppose that $\exists L \in Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi$ such that $L \notin Labs_\sigma(F \star \varphi \wedge \psi)$. We can rephraze it in this way:

$$L \in \min(L^\sigma_\varphi, \leq^\sigma_F) \cap L^\sigma_\psi \text{ and } L \notin \min(L_{\varphi \wedge \psi}, \leq^\sigma_F)$$

From the first part, we deduce $L \in L_{\varphi \wedge \psi}$. However, L is not a minimal element in this set with respect to $\leq^\sigma_F$. Consequently, $\exists L' \in L_{\varphi \wedge \psi}$ such that $L' <^\sigma_F L$. From the definition of $L_{\varphi \wedge \psi}$, $L' \in L^\sigma_\varphi$ holds. This is in contradiction with $L \in \min(L^\sigma_\varphi, \leq^\sigma_F)$.
So $Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi \subseteq Labs_\sigma(F \star \varphi \wedge \psi)$.

(AL6) If $Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi \neq \emptyset$, then $Labs_\sigma(F \star \varphi \wedge \psi) \subseteq Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi$.
If $Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi \neq \emptyset$, then let us suppose $\exists L \in Labs_\sigma(F \star \varphi \wedge \psi)$ such that $L \notin Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi$.
$L \in \min(L_{\varphi \wedge \psi}, \leq^\sigma_F)$ means that $L \in L_{\varphi \wedge \psi}$, and so $L \in L^\sigma_\psi$ holds.
From this and $L \notin Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi$, we deduce $L \notin Labs_\sigma(F \star \varphi)$.
Since we suppose that the intersection is non-empty, $\exists L' \in Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi$. In particular, L' is a σ-model of φ and ψ, *i.e.* $L' \in L_{\varphi \wedge \psi}$. From $L \in Labs_\sigma(F \star \varphi \wedge \psi) = \min(L_{\varphi \wedge \psi}, \leq^\sigma_F)$ and $\leq^\sigma_F$ is a total relation, $L \leq^\sigma_F L'$.
As $L' \in Labs_\sigma(F \star \varphi) = \min(L^\sigma_\varphi, \leq^\sigma_F)$, $L \in \min(L^\sigma_\varphi, \leq^\sigma_F)$. It is a contradiction.
So $Labs_\sigma(F \star \varphi \wedge \psi) \subseteq Labs_\sigma(F \star \varphi) \cap L^\sigma_\psi$ holds.

Now, let us suppose the existence of a revision operator $\star$ which satisfies the postulates. Given an argumentation system $F = \langle A, R \rangle$ and a semantics σ, we define a relation $\leq^\sigma_F$ such that $\forall L_1, L_2 \in Labs^A_\sigma$,

$$L_1 \leq_{AF} L_2 \Leftrightarrow \begin{cases} L_1 \in Labs_\sigma(F) \\ \text{or} \\ L_1 \in Labs_\sigma(F \star \mathrm{form}_\sigma(L_1, L_2)) \end{cases}$$

where $\mathrm{form}_\sigma(L_1, \ldots, L_n)$ is the formula such that $L^\sigma_{\mathrm{form}_\sigma(L_1,\ldots,L_n)} = \{L_1, \ldots, L_n\}$.

Let us study the properties of $\leq^\sigma_F$.
Let $L_1, L_2 \in Labs^A_\sigma$. $\varphi = \mathrm{form}_\sigma(L_1, L_2)$ admits at least one model. Since $\star$ satisfies **(AL3)**, $Labs_\sigma(F \star \varphi) \neq \emptyset$. Moreover, from **(AL1)**, we deduce that $\emptyset \subset Labs_\sigma(F \star \varphi) \subseteq L^\sigma_\varphi = \{L_1, L_2\}$. From there,

$L_1 \in Labs_\sigma(F \star \varphi)$ or $L_2 \in Labs_\sigma(F \star \varphi)$, *i.e.* $L_1 \leq_F^\sigma L_2$ or $L_2 \leq_F^\sigma L_1$, so the relation is total. Moreover, if $L_1 = L_2$, $L_1 \in Labs_\sigma(F \star \text{form}_\sigma(L_1, L_1))$ holds, *i.e.* $L_1 \leq_F^\sigma L_1$: the relation is reflexive. Finally, we want to prove that the relation is transitive. Let $L_1, L_2, L_3 \in Labs_\sigma^A$ such that $L_1 \leq_F^\sigma L_2$ and $L_2 \leq_F^\sigma L_3$. There are three possible cases:

1. $L_1 \in Labs_\sigma(F)$;
2. $L_1 \notin Labs_\sigma(F)$ and $L_2 \in Labs_\sigma(F)$;
3. $L_1 \notin Labs_\sigma(F)$ and $L_2 \notin Labs_\sigma(F)$.

Case 1 If $L_1 \in Labs_\sigma(F)$, $L_1 \leq_F^\sigma L_3$ holds from the definition of the relation.

Case 2 If $L_1 \notin Labs_\sigma(F)$ and $L_2 \in Labs_\sigma(F)$, then $Labs_\sigma(F) \cap \{L_1, L_2\} = \{L_2\}$. From **(AL2)**, $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2)) = \{L_2\}$ holds. Therefore, $L_1 \not\leq_F^\sigma L_2$ since neither $L_1 \in Labs_\sigma(F)$ nor $L_1 \in Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2))$ hold. This is a contradiction: the second case is impossible.

Case 3 If $L_1 \notin Labs_\sigma(F)$ and $L_2 \notin Labs_\sigma(F)$, then since $\star$ satisfies **(AL1)** and **(AL3)**, $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3))$ is a non-empty subset of $\{L_1, L_2, L_3\}$. There are two possible alternatives:

1. $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3)) \cap \{L_1, L_2\} = \emptyset$, *i.e.* $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3)) = \{L_3\}$;
2. $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3)) \cap \{L_1, L_2\} \neq \emptyset$.

(1) If $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3)) \cap \{L_1, L_2\} = \emptyset$, then $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3)) = \{L_3\}$ (because it is a non-empty subset of $\{L_1, L_2, L_3\}$), and from **(AL5)** and **(AL6)**, $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3)) \cap \{L_2, L_3\} = Labs_\sigma(F \star \text{form}_\sigma(L_2, L_3)) = \{L_3\}$ which is in contradiction with the fact that $L_2 \leq_F^\sigma L_3$ and $L_2 \notin Labs_\sigma(F)$. So this case is impossible.

(2) Suppose that $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3)) \cap \{L_1, L_2\} \neq \emptyset$. Since $L_1 \leq_F^\sigma L_2$ and $L_1 \notin Labs_\sigma(F)$, $L_1 \in Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2))$ holds. From **(AL5)** and **(AL6)**, $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3)) \cap \{L_1, L_2\} = Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2))$ holds. So $L_1 \in Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3)) \cap \{L_1, L_2\}$, in particular $L_1 \in Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3))$.
Similarly, since $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3)) \cap \{L_1, L_2\} \neq \emptyset$, from **(AL5)** and **(AL6)**, $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2, L_3)) \cap \{L_1, L_3\} = Labs_\sigma(F \star \text{form}_\sigma(L_1, L_3))$ holds, and we deduce $L_1 \in Labs_\sigma(F \star \text{form}_\sigma(L_1, L_3))$, *i.e.* $L_1 \leq_F^\sigma L_3$.

This concludes the proof that the relation is transitive. So $\leq_F^\sigma$ is a total pre-order. Let us now show that mapping such a pre-order to an argumentation framework is a faithful assignment.

1. Given an argumentation system $F = \langle A, R \rangle$, let L_1, L_2 two elements in $Labs_\sigma^A$ such that $L_1 \in Labs_\sigma(F)$ and $L_2 \in Labs_\sigma(F)$. Since $L_1 \in Labs_\sigma(F)$, $L_1 \leq_F^\sigma L_2$. Likewise, $L_2 \in Labs_\sigma(F)$ implies $L_2 \leq_F^\sigma L_1$. Consequently, $L_1 \approx_F^\sigma L_2$.
2. Given an argumentation system $F = \langle A, R \rangle$, let L_1, L_2 two elements of $Labs_\sigma^A$ such that $L_1 \in Labs_\sigma(F)$ and $L_2 \notin Labs_\sigma(F)$. Like previously, $L_1 \leq_F^\sigma L_2$ holds. Moreover, now $L_2 \notin Labs_\sigma(F)$ holds. From this and **(AL2)**, $Labs_\sigma(F \star \text{form}_\sigma(L_1, L_2)) = \{L_1\}$ holds, so $L_2 \not\leq_F^\sigma L_1$. Therefore, $L_1 <_F^\sigma L_2$.

The last thing to prove is the equality $Labs_\sigma(F \star \varphi) = \min(L^\sigma_\varphi, \leq^\sigma_F)$. First, if φ is not σ-consistent, from **(AL1)** we deduce $Labs_\sigma(F \star \varphi) = \emptyset = \min(L^\sigma_\varphi, \leq_F)$. So we suppose now that φ is σ-consistent, *i.e.* $L^\sigma_\varphi \neq \emptyset$.
Let us prove first the inclusion $Labs_\sigma(F \star \varphi) \subseteq \min(L^\sigma_\varphi, \leq^\sigma_F)$. Using *reductio ad absurdum*: suppose that $\exists L \in Labs_\sigma(F \star \varphi)$ such that $L \notin \min(L^\sigma_\varphi, \leq^\sigma_F)$. From **(AL1)**, $L \in L^\sigma_\varphi$ holds, and since $L \notin \min(L^\sigma_\varphi, \leq^\sigma_F)$, $\exists L' \in L^\sigma_\varphi$ such that $L' <^\sigma_F L$. Two cases must be considered:
<u>Case 1</u> $L' \in Labs_\sigma(F)$, as $L' \in L^\sigma_\varphi$, $Labs_\sigma(F) \cap L^\sigma_\varphi \neq \emptyset$, which implies $Labs_\sigma(F \star \varphi) = Labs_\sigma(F) \cap L^\sigma_\varphi$ (from **(AL2)**).

$$\begin{aligned} L \in Labs_\sigma(F \star \varphi) &\Rightarrow L \in Labs_\sigma(F) \cap L^\sigma_\varphi \\ &\Rightarrow L \in Labs_\sigma(F) \\ &\Rightarrow L \leq^\sigma_F L' \\ &\Rightarrow \text{ contradiction with } L' <^\sigma_F L \end{aligned}$$

<u>Case 2</u> $Labs_\sigma(F \star \text{form}_\sigma(L, L')) = \{L'\}$. Since L and L' are σ-models of φ, $\varphi \wedge \text{form}_\sigma(L, L') \equiv \text{form}_\sigma(L, L')$. From **(AL5)**, $Labs_\sigma(F \star \varphi) \cap \{L, L'\} \subseteq Labs_\sigma(F \star \text{form}_\sigma(L, L')) = \{L'\}$, therefore $L \notin Labs_\sigma(F \star \varphi)$, which is a contradiction.

So $Labs_\sigma(F \star \varphi) \subseteq \min(L^\sigma_\varphi, \leq^\sigma_F)$ holds, let us now prove the converse. We still use *reductio ad absurdum*. Suppose that $\exists L \in \min(L^\sigma_\varphi, \leq^\sigma_F)$ such that $L \notin Labs_\sigma(F \star \varphi)$.
Since φ is σ-consistent, from **(AL3)**, $\exists L' \in Labs_\sigma(F \star \varphi)$, and from **(AL1)** $L' \in L^\sigma_\varphi$. Since L and L' are σ-models of φ, $\text{form}_\sigma(L, L') \wedge \varphi = \text{form}_\sigma(L, L')$. From **(AL5)-(AL6)**, $Labs_\sigma(F \star \varphi) \cap \{L, L'\} = Labs_\sigma(F \star \text{form}_\sigma(L, L')) = \{L'\}$ since $L \notin Labs_\sigma(F \star \varphi)$, so $L' \leq^\sigma_F L$. On the other hand, L is minimal in L^σ_φ with respect to $\leq^\sigma_F$, *i.e.* $L \leq^\sigma_F L'$. Since $L \notin Labs_\sigma(F \star \text{form}_\sigma(L, L')) = \{L'\}$, $L \in Labs_\sigma(F)$ holds, and from **(AL2)** $L \in Labs_\sigma(F \star \varphi)$. It is a contradiction.
We can conclude that $Labs_\sigma(F \star \varphi) = \min(L^\sigma_\varphi, \leq^\sigma_F)$. □

Proposition 34.
Let d be a pseudo-distance between labellings. The labelling-distance-based revision operator $\star_d$ *such that, for every argumentation framework F, every semantics σ and every formula σ*

$$Labs_\sigma(F \star_d \varphi) = \min(L^\sigma_\varphi, \leq^d_F)$$

*satisfies the rationality postulates **(AL1)-(AL6)**.*

Proof. Let d be a distance between labellings. Let us prove that the mapping from an argumentation system F to a total pre-order $\leq^d_F$ is a faithful assignment. Let L_1, L_2 be two σ-representable labellings:

- If $L_1 \in Labs_\sigma(F)$ and $L_2 \in Labs(F)$, then $d(L_1, Labs_\sigma(F)) = d(L_2, Labs_\sigma(F)) = 0$ and so $L_1 \approx^d_F L_1$;
- If $L_1 \in Labs_\sigma(F)$ and $L_2 \notin Labs(F)$, then $d(L_1, Labs_\sigma(F)) = 0$ and $d(L_2, Labs_\sigma(F)) > 0$, so $L_1 <^d_F L_1$.

This result with Proposition 33 proves that $\star_d$ satisfies **(AL1)-(AL6)**. □

Lemma 1.
For each formula $\varphi \in \mathcal{L}^{Labs}_A$ such that φ only contains in variables, there is a formula $\varphi' \in \mathcal{L}_A$ such that $\forall c \in \mathcal{A}^\sigma_{\varphi'}$, $\exists L \in L^\sigma_\varphi$ such that $in(L) = c$; and $\forall L \in L^\sigma_{\varphi'}$, $\exists c \in \mathcal{A}^\sigma_\varphi$ such that $in(L) = c$.
We call φ' the extension-based formula equivalent to φ.

Proof. Follows from the definitions of $\mathcal{L}_A$-formulae and $\mathcal{L}^{Labs}_A$-formulae. □

Proposition 35.
*Each operator satisfying **(AL1)-(AL6)**, restricted to formulae built on the in variables, satisfies **(AE1)-(AE6)**.*

Proof. Let $\star_{Labs}$ be a revision operator satisfying **(AL1)-(AL6)**. From the definition, we know that for each argumentation framework F, there exists a total pre-order between labellings $\leq_F^{Labs}$, satisfying faithful assignment properties, such that for each $\varphi \in \mathcal{L}_A^{Labs}$, $Labs_\sigma(F \star_{Labs} \varphi) = \min(L_\varphi^\sigma, \leq_F^{Labs})$. Let us prove that there exist a total-pre order between candidates $\leq_F$ such that $Ext_\sigma(F \star_{Labs} \varphi) = \min(A_{\varphi'}^\sigma, \leq_F)$, with φ' the extension-based formula equivalent to φ.

First, we call the *level* i of a pre-order $\leq$ the set of elements $E' \subseteq E$ such that, for each element $e \in E'$, the length of the longest sequence of elements $e_1, \ldots, e_i$ such that $e_1 < e_2, \ldots, e_{i-1} < e_i$ and $e_i = e$ is i.
Now, let us define a pre-order on candidates from the pre-order on labellings. For each level of $\leq_F^{Labs}$, from 0 to the highest level, we consider each labelling L. We call c the candidate associated with L (meaning, the in part of L). If c does not already belong to a level $j \leq i$ of $\leq_F$, then c is added to $\leq_F$ at the level i.
From the definition of $\star_{Labs}$, we know that the labellings of the outcome of $F \star_{Labs} \varphi$ are the minimal elements with respect to $\leq_F^{Labs}$ among the labellings which satisfy φ. Let us prove that the in-part of these labellings correspond to the minimal elements of $A_{\varphi'}^\sigma$ with respect to $\leq_F$.
Let us suppose that there is a labelling $L \in \min(L_\varphi^\sigma, \leq_F^{Labs})$ such that $E(L) \notin \min(A_{\varphi'}^\sigma, \leq_F)$. From the definition of φ', $E(L) \in \mathcal{A}_{\varphi'}^\sigma$. So, there is a candidate $c \in \mathcal{A}_{\varphi'}^\sigma$ such that $c <_F E(L)$. The definitions of $\leq_F$ and φ' implies that there exist a labelling $L' \in L_\varphi^\sigma$ such that $E(L') = c$ and $L' <_F^{Labs} L$, which is a contradiction.
So, $Ext_\sigma(F \star_{Labs} \varphi) = \min(A_{\varphi'}^\sigma, \leq_F)$. Since $\leq_F^{Labs}$ satisfies the conditions of faithful assignments, $\leq_F$ also satisfies these conditions: the level 0 of $\leq_F$ only contains candidates which are the in part of the σ-labellings of F, meaning the σ-extensions of F. So, from Proposition 25, $\star_{Labs}$ satisfies **(AE1)-(AE6)**. □

Proposition 29.
*Every revision operator $\star$ defined following Definition 85 satisfies the postulates **(AE1)-(AE6)**.*

Proof. Given a semantics σ, the definition of $\mathcal{AF}$ implies that $Ext_\sigma(\mathcal{AF}(C)) = C$, where C is a set of candidates. So, given $\star$ a revision operator and $\varphi \in \mathcal{L}_A$, $Ext_\sigma(AF \star \varphi) = \min(\mathcal{A}_\varphi^\sigma, \leq_{AF}^\sigma)$. $\leq_{AF}^\sigma$ being a faithful assignment, the operator $\star$ satisfies the postulates **(AE1)-(AE6)** (as a consequence of Proposition 25). □

Proposition 36.
*Every revision operator $\star$ defined following Definition 99 satisfies the postulates **(AE1)-(AE6)** under the assumption that $\mathfrak{C}$ is not conflicting with the revision formula.*

Proof. Given a semantics σ, the definition of $\mathcal{AF}$ implies that $Ext_\sigma(\mathcal{AF}(C, \mathfrak{C})) = C$, where C is a set of candidates. So, given $\star$ a revision operator and $\varphi \in \mathcal{L}_A$, $Ext_\sigma(AF \star \varphi) = \min(\mathcal{A}_\varphi^\sigma, \leq_{AF}^\sigma)$. $\leq_{AF}^\sigma$ being a faithful assignment, the operator $\star$ satisfies the postulates **(AE1)-(AE6)** (as a consequence of Proposition 25). □

Proposition 30.
Let $\star$ be a revision operator based on the generation operator $\mathcal{AF}_\sigma^{dg,F}$. The size of $F \star \varphi$ can be exponential in $|A|$.

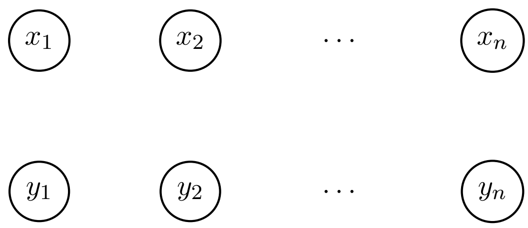


Figure B.1: The framework AF_{18}

Proof. Let $AF_{18} = \langle\{x_1, \ldots, x_n, y_1, \ldots, y_n\}, \emptyset\rangle$ be an argument framework. Whatever the semantics σ (for instance the stable semantics), $Ext_\sigma(AF_{18}) = \{\{x_1, \ldots, x_n, y_1, \ldots, y_n\}\}$.
Let $\varphi_7 = \bigwedge_{i=1}^{n}(x_i \vee y_i) \wedge (\neg x_i \vee \neg y_i)$, and let the revision operator $\star$ be based on the Hamming distance between candidates and the generation operator $AF_\sigma^{dg,AF}$.
The computation of the first step of the revision process can be intuitively explained as follow. A candidate extension satisfying φ is a choice, for each i, of exactly one argument among x_i and y_i. 2^n candidates, each one of size n, containing either x_i or y_i, must be considered.
To build argumentation frameworks corresponding to these candidates such that the distance from AF_{18} is minimal, we can add n attacks: for each i, either x_i attacks y_i or the converse. It is not possible to add less than n attacks, in this case there would be at least one i such that x_i and y_i both belong to the extension of the system.
This leads to the generation of 2^n graphs, each one corresponding exactly to one candidate. One of these frameworks is given, as an example, in Fig. B.2. □

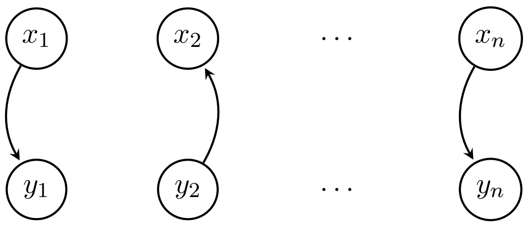


Figure B.2: One of the revised frameworks from $AF_{18} \star \varphi_7$

Proposition 31.
Let C *be a complexity class which is closed under polynomial-time reductions. Suppose that* $\star$ *satisfies* ***(AE1)*** *to* ***(AE6)****, and that the semantics* σ *ensures the existence of an extension for every* F*. If the inference problem from an argumentation system is* C*-hard, then the inference problem from a revised argumentation system is* C*-hard as well.*

Proof. We prove that if $\star$ is a revision operator and σ is such that $Ext_\sigma(AF) \neq \emptyset$, then $AF \star (a \vee \neg a) \mid\!\sim_\sigma \psi$ if and only if $AF \mid\!\sim_\sigma \psi$ (where a is any argument in A). Postulate **(AE2)** states that for every formula $\varphi \in \mathcal{L}_A$, if $Ext_\sigma(AF) \cap \mathcal{A}_\varphi^\sigma \neq \emptyset$, then $Ext_\sigma(AF \star \varphi) = Ext_\sigma(AF) \cap \mathcal{A}_\varphi^\sigma$. Since $\mathcal{A}_{a \vee \neg a}$ is the power set of A, if $Ext_\sigma(AF)$ is not empty, then $Ext_\sigma(AF) \cap \mathcal{A}_{a \vee \neg a}$ is not empty as well. Hence, $AF \star (a \vee \neg a) \mid\!\sim_\sigma \psi$ if and only if $AF \mid\!\sim_\sigma \psi$, and the result follows. □

Proposition 32.
Suppose that $\star$ *satisfies* ***(AE1)*** *and* ***(AE3)****. The inference problem from a revised argumentation system with respect to the grounded semantics is* coNP*-hard, even under the restriction when the queries* ψ *are restricted to CNF formulae on* A*.*

Proof. By reduction from UNSAT. Let Σ be a propositional formula over the set of variables $\{x_1, \dots, x_n\}$. We associate with Σ in polynomial time the triple $(AF = (\{x_1, \dots, x_n, y\}, \emptyset), \varphi = \Sigma \wedge \neg y, \psi = y)$. From **(AE1)**, it must be the case that $AF \star \varphi \mid\!\sim_{gr} \varphi$, hence we must have $AF \star \varphi \mid\!\sim_{gr} \neg y$.

If in addition we have $AF \star \varphi \mid\!\sim_{gr} y$, then it must be the case that $Ext_{gr}(AF) = \emptyset$. Then, from **(AE3)**, it must be the case that φ is inconsistent, which implies that Σ is inconsistent. Conversely, if Σ is consistent, then φ is inconsistent as well. In this case $\mathcal{A}^{\sigma}_{\varphi}$ is empty, so from **(AE1)**, we must have $Ext_{gr}(AF \star \varphi) = \emptyset$. Subsequently, $AF \star \varphi \mid\!\sim_{gr} y$. □

Appendix C

Proofs of the Results from Chapter 5

Proposition 37.
For each propositional formula $\varphi \in \mathcal{L}_A$, mapping φ to the total pre-order defined by

$$\omega_1 \leq_\varphi^{rem} \omega_2 \text{ if and only if } \min_{\omega_3 \in \mathrm{Mod}(\varphi)}(d^{rem}(\omega_1, \omega_3)) \leq \min_{\omega_3 \in \mathrm{Mod}(\varphi)}(d^{rem}(\omega_2, \omega_3))$$

is a faithful assignment.

Proof.

- If $\omega_1 \models \varphi$ and $\omega_2 \models \varphi$, then $\min_{\omega_3 \in \mathrm{Mod}(\varphi)}(d^{att}(\omega_1, \omega_3)) = \min_{\omega_3 \in \mathrm{Mod}(\varphi)}(d^{att}(\omega_2, \omega_3)) = 0$.
- If $\omega_1 \models \varphi$ and $\omega_2 \not\models \varphi$, then $\min_{\omega_3 \in \mathrm{Mod}(\varphi)}(d^{att}(\omega_1, \omega_3)) = 0$ and $\min_{\omega_3 \in \mathrm{Mod}(\varphi)}(d^{att}(\omega_2, \omega_3)) > 0$.
- For each φ' such that $\varphi' \equiv \varphi$, the definition of the pre-order is such that $\leq_\varphi^{att} = \leq_{\varphi'}^{att}$.

□

Proposition 38.
Given a pseudo-distance d between sets of arguments and an argumentation framework F, $\leq_F^d$ denotes the total pre-order between sets of arguments defined by: $\varepsilon_1 \leq_F^d \varepsilon_2$ iff $d(\varepsilon_1, Sc_\sigma(F)) \leq d(\varepsilon_2, Sc_\sigma(F))$. The pseudo-distance based revision operator *$\star_d$ which satisfies*

$$Sc_\sigma(F \star_d \varphi) = \min(\mathcal{S}(\varphi), \leq_F^d)$$

*satisfies the postulates **(AS1)** - **(AS6)**.*

Proof. **(AS1)** is satisfied from the definition of the operator.

If $Sc_\sigma(F) \cap \mathcal{S}(\varphi) \neq \emptyset$, then obviously $\forall \varepsilon \in Sc_\sigma(F) \cap \mathcal{S}(\varphi)$, $\varepsilon \in Sc_\sigma(F)$, and $d(\varepsilon, Sc_\sigma(F)) = 0$. Any ε' which is not in $Sc_\sigma(F) \cap \mathcal{S}(\varphi)$ either does not satisfy φ (and so does not belong to $\mathcal{S}(\varphi)$), or does not belong to $Sc_\sigma(F)$ (and so $d(\varepsilon', Sc_\sigma(F)) > 0$). So $\min(\mathcal{S}(\varphi), \leq_F^d) = Sc_\sigma(F) \cap \mathcal{S}(\varphi)$, which leads to **(AS2)**.

If φ is acc-consistent, $\mathcal{S}(\varphi) \neq \emptyset$, so $\min(\mathcal{S}(\varphi), \leq_F^d) \neq \emptyset$. So **(AS3)** holds.

$\varphi \equiv_{acc} \varphi$ can be rewritten $\mathcal{S}(\varphi) = \mathcal{S}(\psi)$, which leads to $\min(\mathcal{S}(\varphi), \leq_F^d) = \min(\mathcal{S}(\psi), \leq_F^d)$. It is enough to prove **(AS4)**.

If $Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi) = \emptyset$, **(AS5)-(AS6)** are satisfied. We suppose now that $Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi) \neq \emptyset$.
We first prove the inclusion $Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi) \subseteq Sc_\sigma(F \star \varphi \wedge \psi)$. By *reductio ad absurdum*, suppose that $\exists \varepsilon \in Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\varphi \wedge \psi)$ such that $\varepsilon \notin Sc_\sigma(F \star \varphi \wedge \psi)$, also written as $\varepsilon \in \min(\mathcal{S}(\varphi), \leq_F^d) \cap \mathcal{S}(\psi)$ and $\varepsilon \notin \min(\mathcal{S}(\varphi \wedge \psi), \leq_F^d)$. From the first part, we deduce $\varepsilon \in \mathcal{S}(\varphi \wedge \psi)$. However, ε is not a minimal element in this set with respect to $\leq_F^d$. Consequently, $\exists \varepsilon' \in \mathcal{S}(\varphi \wedge \psi)$ such that $\varepsilon' <_F^d \varepsilon$. From the definition of $\mathcal{S}(\varphi \wedge \psi)$, $\varepsilon' \in \mathcal{S}(\varphi)$ holds. This contradicts $\varepsilon \in \min(\mathcal{S}(\varphi), \leq_F^d)$. So $Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\varphi \wedge \psi) \subseteq Sc_\sigma(F \star \varphi \wedge \psi)$, **(AS5)** holds.

If $Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi) \neq \emptyset$, let us suppose $\exists \varepsilon \in Sc_\sigma(F \star \varphi \wedge \psi)$ such that $\varepsilon \notin Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi)$. $\varepsilon \in \min(\mathcal{S}(\varphi \wedge \psi), \leq_F^d) \Rightarrow \varepsilon \in \mathcal{S}(\varphi \wedge \psi) \Rightarrow \varepsilon \in \mathcal{S}(\psi)$ holds. From this and $\varepsilon \notin Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi)$, we deduce $\varepsilon \notin Sc_\sigma(F \star \varphi)$.
Since we suppose that the intersection is non-empty, $\exists \varepsilon' \in Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi)$. In particular, ε' satisfies φ and ψ, i.e. $\varepsilon' \in \mathcal{S}(\varphi) \cap \mathcal{S}(\psi) = \mathcal{S}(\varphi \wedge \psi)$. From $\varepsilon \in Sc_\sigma(F \star \varphi \wedge \psi) = \min(\mathcal{S}(\varphi \wedge \psi), \leq_F^d)$ and $\leq_F^d$ is a total relation, $\varepsilon \leq_F^d \varepsilon'$.
As $\varepsilon' \in Sc_\sigma(F \star \varphi) = \min(\mathcal{S}(\varphi), \leq_F^d)$, $\varepsilon \in \min(\mathcal{S}(\varphi), \leq_F^d)$. It is a contradiction.
So $Sc_\sigma(F \star \varphi \wedge \psi) \subseteq Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi)$ holds.

□

Proposition 39.
*The arguments statuses minimal revision operator satisfies the postulates **(AS1)-(AS6)**.*

Proof. Let us show that the arguments statuses minimal revision operator is a pseudo-distance based revision operator. We define $Proj_{acc}$ as the counterpart of $Proj_{att}$ to project the formulae on their acc-part.
$F \star_D^{acc} \varphi = arg(Proj_{att}(f_\sigma(F) \circ_D^{acc} (\varphi \wedge th_\sigma(A))))$ leads to

$$\begin{aligned} Sc_\sigma(F \star_D^{acc} \varphi) &= Proj_{acc}(f_\sigma(F) \circ_D^{acc} (\varphi \wedge th_\sigma(A))) \\ &= Proj_{acc}(\min(Mod(\varphi \wedge th_\sigma(A)), \leq_F^{d_H^{acc}})) \end{aligned}$$

Let us prove that projecting the minimal models of $\varphi \wedge th_\sigma(A)$ leads to the minimal sets of skeptically accepted arguments. The models of $\varphi \wedge th_\sigma(A)$ are the propositional representations of argumentation frameworks which satisfy φ, so it is obvious that the projection of the models on the acc variables allows to obtain a subset of $\mathcal{S}(\varphi)$. Let us show that these sets of arguments are minimal with respect to $\leq_F^d$:
Given $m \in \min(Mod(\varphi \wedge th_\sigma(A)), \leq_F^{d_H^{acc}})$, we have $d_H^{acc}(m, Mod(f_\sigma(F))$ is minimal. $f_\sigma(F)$ has a single model m_F, so $d_H^{acc}(m, m_F)$ is minimal. In other words,

$$(|A|^2 + 1) \sum_{a \in A} (m(acc(a)) \oplus m_F(acc(a))) + \sum_{a,b \in A} (m(att(a,b)) \oplus m_F(att(a,b)))$$

is minimal. Let us suppose that the acc part of the distance is not minimal, i.e. there exists m' such that

$$(|A|^2 + 1) \sum_{a \in A} (m'(acc(a)) \oplus m_F(acc(a))) < (|A|^2 + 1) \sum_{a \in A} (m(acc(a)) \oplus m_F(acc(a)))$$

In the extreme case when $\sum_{a,b \in A} (m(att(a,b)) \oplus m_F(att(a,b))) = 0$ and $\sum_{a,b \in A} (m'(att(a,b)) \oplus m_F(att(a,b))) = |A|^2$,

$$\begin{aligned} &(|A|^2 + 1) \textstyle\sum_{a \in A} (m'(acc(a)) \oplus m_F(acc(a))) \\ &\qquad + \textstyle\sum_{a,b \in A} (m'(att(a,b)) \oplus m_F(att(a,b))) \\ &< (|A|^2 + 1) \textstyle\sum_{a \in A} (m(acc(a)) \oplus m_F(acc(a))) \\ &\qquad + \textstyle\sum_{a,b \in A} (m(att(a,b)) \oplus m_F(att(a,b))) \end{aligned}$$

is ensured by the weight $|A|^2 + 1$ on the acc part. By *reductio ad absurdum*, we proved that the acc part of $d_H^{acc}(m, m_F)$ is minimal, i.e., $d_H(Proj_{acc}(m), Sc_\sigma(F))$ is minimal, with d_H the Hamming distance [Ham50]. It implies

$$\begin{array}{rcl} Sc_\sigma(F \star_D^{acc} \varphi) & = & Proj_{acc}(\min(Mod(\varphi \wedge th_\sigma(A)), \leq_F^{d_F^{acc}})) \\ & = & \min(\mathcal{S}(\varphi), \leq_F^{d_H}) \end{array}$$

From Prop. 38, $\star_D^{acc}$ satisfies the postulates **(AS1)-(AS6)**. □

Appendix D

Proofs of the Results from Chapter 6

Proposition 40.
For every $F = \langle A, R \rangle$ and $E \subseteq A$ a stable non-trivial set in F, there is no strict enforcement of E in F with respect to the stable semantics.

Proof.
E is known not to be a stable extension of F. It means that $\exists a_i \in A \backslash E$ such that E does not attack a_i. Whatever A' used to perform the normal (including weak or strong) expansion, $a_i \in (A \cup A') \backslash E$ is not attacked by E (because the attacks between arguments in A are not changed), so E cannot be a stable extension of any normal expansion of F. □

Proposition 41.
For every $F = \langle A, R \rangle$, and $E \subseteq A$ a complete non-trivial set in F,

1. *if E is not admissible, then there is no strict enforcement of E in F with respect to the complete semantics.*
2. *else, if E defends some argument $a_i \in A \backslash E$, then*
   - *(a) there is no strict weak enforcement of E in F with respect to the complete semantics.*
   - *(b) if odd-length cycles are not allowed, then there is no strict strong enforcement of E in F with respect to the complete semantics.*

Proof.

1. $\exists a_i \in A \backslash E$ such that a_i attacks some argument in E, and E does not attack a_i. Whatever A' used to perform the normal (including weak or strong) expansion, $a_i \in (A \cup A') \backslash E$ is not attacked by E (because the attacks between arguments in A are not changed), so E cannot be a complete extension of any normal expansion of F.
2. There is some argument $a_i \in A \backslash E$ such that E defends a_i against all its attackers. As it is not allowed to change the attacks between the arguments in A, the only way for E to be a complete extension is to add some b attacking a_i, such that E does not attack b.
   - (a) It is impossible with *weak* enforcement.
   - (b) To ensure that b is not included in the extension, it has to be attacked. It cannot be by an argument in E (or else, E still defends a_i, and so E is not a complete extension). Two cases are possible:

- If b is attacked by an argument in $a_j \in A \backslash E$, then the enforcement is not strong (but it is possible with normal enforcement).
- Let us reason with the setting of labellings. We expect E to be the in part of a complete labelling. b must be $undec$ with respect to ths complete labelling (it cannot be in, since E is expected to be strictly enforced, and it cannot be out since it would mean that E attacks b, which is forbidden with strong enforcement). The only way to ensure that b is $undec$ is to add other arguments, which form an odd-length cycle including b.

□

Proposition 42.
For every $F = \langle A, R \rangle$ and $E \subseteq A$ a grounded non-trivial set in F, if $Ext_{gr}(F) = \{\emptyset\}$, then there is no strict enforcement of E in F with respect to the grounded semantics.

Proof.
If $Ext_{gr}(F) = \{\emptyset\}$, it means that there is no unattacked argument in F. Let us call F' any normal expansion of F. If there is no unattacked argument in F', then $Ext_{gr}(F') = \{\emptyset\}$. Else, $\exists b \notin A$ such that $b \in E'$, with $Ext_{gr}(F') = \{E'\}$. Whatever the case, E cannot be enforced by any normal enforcement operator. □

Proposition 43.
Let $F = \langle A, R \rangle$ be an argumentation framework, σ an acceptability semantics and $E \subseteq A$ a set of arguments. There is a strict enforcement F' of E.

Proof.
We just need to define $F' = \langle A, R' \rangle$ with $R' = E \times (A \backslash E)$ to ensure that E is a σ-extension, whatever the semantics σ. □

Proposition 44.
Let $F = \langle A, R \rangle$ be an argumentation framework, $E \subseteq A$, and k be an integer. Determining whether it is possible to enforce E in F under the stable semantics with at most k changes (addition or removal) of attacks is NP*-hard.*

Proof.
Let us consider the particular case of $k = 0$, and $E = \{a\}$. Let $F = \langle A, R \rangle$ an AF such that $a \notin A$. Enforcing E under the stable semantics in $F' = \langle A \cup \{a\}, R \rangle$ without any change of attacks is possible if and only if $Ext_{st}(F) \neq \emptyset$. It is well-known that testing the existence of a stable extension of an AF is NP-hard, so our problem is also NP-hard. □

List of Figures

List of Tables

Bibliography

[ABN13] Leila AMGOUD and Jonathan BEN-NAIM. « Ranking-Based Semantics for Argumentation Frameworks ». In *Proceedings of the Seventh International Conference on Scalable Uncertainty Management (SUM 2013)*, pages 134–147, 2013. *2 citations pages 126 and 128*

[AC02a] Leila AMGOUD and Claudette CAYROL. « Inferring from Inconsistency in Preference-Based Argumentation Frameworks ». *International Journal of Automated Reasoning*, 29(2):125–169, 2002. *2 citations pages 70 and 126*

[AC02b] Leila AMGOUD and Claudette CAYROL. « A Reasoning Model Based on the Production of Acceptable Arguments ». *Annals of Mathematics and Artificial Intelligence*, 34(1-3):197–215, 2002. *2 citations pages 126 and 127*

[AC13] Ofer ARIELI and Martin W. A. CAMINADA. « A QBF-Based Formalization of Abstract Argumentation Semantics ». *Journal of Applied Logic*, 11(2):229–252, 2013. *Cited page 108*

[ADM08] Leila AMGOUD, Yannis DIMOPOULOS, and Pavlos MORAITIS. « Making Decisions through Preference-Based Argumentation ». In *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR 2008)*, pages 113–123, 2008. *2 citations pages 8 and 21*

[AGM85] Carlos E. ALCHOURRÓN, Peter GÄRDENFORS, and David MAKINSON. « On The Logic Of Theory Change : Partial Meet Contraction And Revision Functions ». *Journal of Symbolic Logic*, 50:510–530, 1985. *10 citations pages 2, 4, 25, 30, 31, 48, 75, 79, 131, and 133*

[AH06] Leila AMGOUD and Nabil HAMEURLAIN. « An Argumentation-Based Approach for Dialog Move Selection ». In *Proceedings of the Third International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006)*, pages 128–141, 2006. *Cited page 2*

[AM82] Carlos E. ALCHOURRÓN and David MAKINSON. « On the Logic of Theory Change: Contraction Functions and their associated revision functions ». *Theoria*, 48:14–37, 1982. *Cited page 31*

[AP09] Leila AMGOUD and Henri PRADE. « Using Arguments for Making and Explaining Decisions ». *Artificial Intelligence*, 173:413–436, 2009. *Cited page 21*

[AV12] Leila AMGOUD and Srdjan VESIC. « Revising Option Status in Argument-Based Decision Systems ». *Journal of Logic and Computation*, 5(22):1019–1058, 2012. *Cited page 21*

[Bau12] Ringo BAUMANN. « What Does it Take to Enforce an Argument? Minimal Change in abstract Argumentation ». In *Proceedings of the Twentieth European Conference on Artificial Intelligence (ECAI 2012)*, pages 127–132, 2012. *13 citations pages 2, 50, 51, 68, 69, 112, 115, 128, 129, 130, 131, 134, and 137*

[BB10] Ringo BAUMANN and Gerhard BREWKA. « Expanding Argumentation Frameworks: Enforcing and Monotonicity Results ». In *Proceedings of the Third International Conference on Computational Models of Argument (COMMA 2010)*, pages 75–86, 2010. *11 citations pages 2, 49, 68, 110, 118, 128, 129, 130, 131, 134, and 137*

[BB15] Ringo BAUMANN and Gerhard BREWKA. « AGM Meets Abstract Argumentation: Expansion and Revision for Dung Frameworks ». In *Proceedings of the 24th Internation Joint Conference on Artificial Intelligence (IJCAI 2015)*, 2015. *5 citations pages 60, 65, 68, 134, and 137*

[BC02] Trevor J. M. BENCH-CAPON. « Value-Based Argumentation Frameworks ». In *Proceedings of the 9th International Workshop on Non-Monotonic Reasoning (NMR'02)*, pages 443–454, 2002. *3 citations pages 70, 126, and 127*

[BC03] Trevor JM BENCH-CAPON. « Persuasion in Practical Argument Using Value-Based Argumentation Frameworks ». *Journal of Logic and Computation*, 13(3):429–448, 2003. *2 citations pages 126 and 127*

[BCdSCLS11] Pierre BISQUERT, Claudette CAYROL, Florence Dupin de SAINT-CYR, and Marie-Christine LAGASQUIE-SCHIEX. « Change in Argumentation Systems: Exploring the Interest of Removing an Argument ». In *Proceedings of the Sixth International Conference on Scalable Uncertainty Management (SUM 2011)*, pages 275–288, 2011. *6 citations pages 2, 43, 48, 49, 68, and 107*

[BCdSL13] Pierre BISQUERT, Claudette CAYROL, Florence Dupin de SAINT-CYR, and Marie-Christine LAGASQUIE-SCHIEX. « Enforcement in Argumentation Is a Kind of Update ». In *Proceedings of the Seventh International Conference on Scalable Uncertainty Management (SUM 2013)*, pages 30–43, 2013. *7 citations pages 58, 107, 127, 128, 131, 134, and 137*

[BCG11] Pietro BARONI, Martin CAMINADA, and Massimiliano GIACOMIN. « An Introduction to Argumentation Semantics ». *Knowledge Engineering Review*, 26(4):365–410, 2011. *Cited page 126*

[BCPR12] Richard BOOTH, Martin CAMINADA, Mikolaj PODLASZEWSKI, and Iyad RAHWAN. « Quantifying Disagreement in Argument-Based Reasoning ». In *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 493–500, 2012. *Cited page 86*

[BD04] Philippe BESNARD and Sylvie DOUTRE. « Checking the Acceptability of a Set of Arguments ». In *Proceedings of the Tenth International Workshop on Non-Monotonic Reasoning (NMR 2004)*, pages 59–64, 2004. *12 citations pages 4, 8, 16, 17, 18, 59, 94, 98, 99, 108, 113, and 135*

[BDH14] Philippe BESNARD, Sylvie DOUTRE, and Andreas HERZIG. « Encoding Argument Graphs in Logic ». In *Proceedings of the Fifteenth International Conference on In-*

formation Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'14), pages 345–354, 2014. *2 citations pages 17 and 108*

[BDHJ14] Anton BELOV, Daniel DIEPOLD, Marijn HEULE, and Matti JÄRVISALO, editors. *Proceedings of SAT Competition 2014: Solver and Benchmark Descriptions*. 2014. *Cited page 157*

[BG07] Pietro BARONI and Massimiliano GIACOMIN. « On Principle-Based Evaluation of Extension-Based Argumentation Semantics ». *Artificial Intelligence*, 171(10):675–700, 2007. *Cited page 126*

[BG09] Pietro BARONI and Massimiliano GIACOMIN. Semantics of Abstract Argument Systems. In Guillermo SIMARI and Iyad RAHWAN, editors, *Argumentation in Artificial Intelligence*, pages 25–44. 2009. *Cited page 126*

[BGK+14] Richard BOOTH, Dov M. GABBAY, Souhila KACI, Tjitze RIENSTRA, and Leendert W. N. van der TORRE. « Abduction and Dialogical Proof in Argumentation and Logic Programming ». In *Proceedings of the Twenty-First European Conference on Artificial Intelligence (ECAI 2014)*, pages 117–122, 2014. *Cited page 121*

[BH01] Philippe BESNARD and Anthony HUNTER. « A Logic-Based Theory of Deductive Arguments ». *Artificial Intelligence*, 128(1-2):203–235, 2001. *Cited page 137*

[BH08] Philippe BESNARD and Anthony HUNTER. *Elements of Argumentation*. MIT press Cambridge, 2008. *Cited page 137*

[BHT13] Philippe BALBIANI, Andreas HERZIG, and Nicolas TROQUARD. « Dynamic Logic of Propositional Assignments: A Well-Behaved Variant of PDL ». In *Twenty-Eighth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2013)*, pages 143–152, 2013. *Cited page 25*

[BKM91] Chitta BARAL, Sarit KRAUS, and Jack MINKER. « Combining Multiple Knowledge Bases ». *IEEE Transactions on Knowledge and Data Engineering*, 3(2):208–220, 1991. *Cited page 80*

[BKMS92] Chitta BARAL, Sarit KRAUS, Jack MINKER, and V. S. SUBRAHMANIAN. « Combining Knowledge Bases Consisting of First-Order Theories ». *Computational Intelligence*, 8(1):45–71, 1992. *Cited page 80*

[BKRvdT13] Richard BOOTH, Souhila KACI, Tjitze RIENSTRA, and Leon van der TORRE. A Logical Theory about Dynamics in Abstract Argumentation. In *Seventh International Conference on Scalable Uncertainty Management (SUM 2013)*, pages 148–161. Springer, 2013. *7 citations pages 60, 61, 85, 128, 129, 134, and 137*

[BKvdT09a] Guido BOELLA, Souhila KACI, and Leendert van der TORRE. « Dynamics in Argumentation with Single Extensions: Abstraction Principles and the Grounded Extension ». In *Proceedings of the Tenth European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2009)*, pages 107–118, 2009. *7 citations pages 2, 43, 44, 45, 68, 107, and 127*

[BKvdT09b] Guido BOELLA, Souhila KACI, and Leendert van der TORRE. « Dynamics in Argumentation with Single Extensions: Attack Refinement and the

Grounded Extension ». In *Proceedings of the International Conference on Autonomous Agents and Multiagents Systems (AAMAS 2009)*, pages 1213–1214, 2009. *6 citations pages 2, 43, 45, 68, 107, and 127*

[BM11] Elise BONZON and Nicolas MAUDET. « On the Outcomes of Multiparty Persuasion ». In *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 47–54, 2011. *Cited page 140*

[BMM14] Elise BONZON, Nicolas MAUDET, and Stefano MORETTI. « Coalitional Games for Abstract Argumentation ». In *Proceedings of the Fifth International Conference on Computational Models of Argument (COMMA'14)*, pages 161–172, 2014. *2 citations pages 8 and 23*

[BW10] Gerhard BREWKA and Stefan WOLTRAN. « Abstract Dialectical Frameworks ». In *Proceedings of the Twelfth International Conference on Principles of Knowledge Representation and Reasoning (KR 2010)*, 2010. *Cited page 126*

[Cam06] Martin CAMINADA. « On the Issue of Reinstatement in Argumentation ». In *Proceedings of the Tenth European Conference on Logics in Artificial Intelligence (JELIA 2006)*, pages 111–123. Springer, 2006. *4 citations pages 8, 13, 14, and 126*

[Cam07] Martin CAMINADA. « Comparing Two Unique Extension Semantics for Formal Argumentation: Ideal and Eager ». In *Proceedings of the Nineteenth Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2007)*, pages 81–87, 2007. *Cited page 11*

[CDE+99] Manuel CLAVEL, Francisco DURÁN, Steven EKER, Patrick LINCOLN, Narciso MARTÍ-OLIET, José MESEGUER, and Jose F. QUESADA. « The Maude System ». In *Proceedings of the Tenth International Conference on Rewriting Techniques and Applications (RTA 1999)*, pages 240–243, 1999. *Cited page 57*

[CDLS99] Marco CADOLI, Francesco M. DONINI, Paolo LIBERATORE, and Marco SCHAERF. « The Size of a Revised Knowledge Base ». *Artificial Intelligence*, 115(1):25–64, 1999. *Cited page 84*

[CDM05] Sylvie COSTE-MARQUIS, Caroline DEVRED, and Pierre MARQUIS. « Symmetric Argumentation Frameworks ». In *Proceedings of the Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU 2005*, pages 317–328, 2005. *Cited page 16*

[CDM06] Sylvie COSTE-MARQUIS, Caroline DEVRED, and Pierre MARQUIS. « Constrained Argumentation Frameworks ». In *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 112–122, 2006. *4 citations pages 61, 126, 128, and 129*

[CdSCLS10] Claudette CAYROL, Florence Dupin de SAINT-CYR, and Marie-Christine LAGASQUIE-SCHIEX. « Change in Abstract Argumentation Frameworks: Adding an Argument ». *Journal of Artificial Intelligence Research*, 38:49–84, 2010. *8 citations pages 2, 43, 45, 68, 104, 107, 127, and 137*

[CKM15] Thomas CARIDROIT, Sébastien KONIECZNY, and Pierre MARQUIS. « Contraction in Propositional Logic ». In *Proceedings of the Thirteenth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2015)*, 2015. *2 citations pages 94 and 140*

[CL05] Claudette CAYROL and Marie-Christine LAGASQUIE-SCHIEX. « Graduality in Argumentation ». *Journal of Artificial Intelligence Research (JAIR)*, 23:245–297, 2005. *Cited page 128*

[CL12] Julien COJAN and Jean LIEBER. « Belief Revision-Based Case-Based Reasoning ». In *Workshop on Similarity and Analogy-based Methods in Artificial Intelligence (SAMAI 2012)*, pages 33–39, 2012. Workshop at ECAI 2012. *Cited page 97*

[CLS05] Claudette CAYROL and Marie-Christine LAGASQUIE-SCHIEX. « On the Acceptability of Arguments in Bipolar Argumentation Frameworks ». In *Proceedings of the Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU 2005*, pages 378–389. Springer, 2005. *2 citations pages 126 and 127*

[CMDK+07] Sylvie COSTE-MARQUIS, Caroline DEVRED, Sébastien KONIECZNY, Marie-Christine LAGASQUIE-SCHIEX, and Pierre MARQUIS. « On the Merging of Dung's Argumentation Systems ». *Artificial Intelligence*, 171:730–753, 2007. *3 citations pages 81, 89, and 140*

[CMKMM13a] Sylvie COSTE-MARQUIS, Sébastien KONIECZNY, Jean-Guy MAILLY, and Pierre MARQUIS. « On the Revision of Argumentation Systems: Minimal Change of Arguments Status ». In *Second International Workshop on Theory and Applications of Formal Argumentation (TAFA 2013)*, 2013. workshop at IJCAI 2013. *no citation*

[CMKMM13b] Sylvie COSTE-MARQUIS, Sébastien KONIECZNY, Jean-Guy MAILLY, and Pierre MARQUIS. « Révision de systèmes d'argumentation : changement minimal du statut des arguments ». In *Septièmes Journées d'Intelligence Artificielle Fondamentale (IAF 2013)*, pages 107–116, 2013. in french. *no citation*

[CMKMM14a] Sylvie COSTE-MARQUIS, Sébastien KONIECZNY, Jean-Guy MAILLY, and Pierre MARQUIS. « Approche par traduction pour la révision de systèmes d'argumentation ». In *Huitièmes Journées d'Intelligence Artificielle Fondamentale (IAF 2014)*, pages 77–85, 2014. in french. *no citation*

[CMKMM14b] Sylvie COSTE-MARQUIS, Sébastien KONIECZNY, Jean-Guy MAILLY, and Pierre MARQUIS. « On the Revision of Argumentation Systems: Minimal Change of Arguments Statuses ». In *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 72–81, 2014. *4 citations pages 93, 104, 128, and 137*

[CMKMM14c] Sylvie COSTE-MARQUIS, Sébastien KONIECZNY, Jean-Guy MAILLY, and Pierre MARQUIS. « A Translation-based Approach for Revision of Argumentation Frameworks ». In *Proceedings of the Fourteenth European Conference on Logics in Artificial Intelligence (JELIA 2014)*, pages 77–85, 2014. *Cited page 137*

[CMKMM15a] Sylvie COSTE-MARQUIS, Sébastien KONIECZNY, Jean-Guy MAILLY, and Pierre MARQUIS. « Forçage d'extension en argumentation abstraite par optimisation booléenne ». In *Neuvièmes Journées d'Intelligence Artificielle Fondamentale (IAF 2015)*, 2015. in french. *no citation*

[CMKMM15b] Sylvie COSTE-MARQUIS, Sébastien KONIECZNY, Jean-Guy MAILLY, and Pierre MARQUIS. « Extension Enforcement in Abstract Argumentation as an Optimization Problem ». In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2876–2882, 2015. *2 citations pages 134 and 137*

[Coo71] Stephen A. COOK. « The Complexity of Theorem-Proving Procedures ». In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971. *Cited page 157*

[Dal88] Mukesh DALAL. « Investigations into a Theory of Knowledge Base Revision: Preliminary Report ». In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI'88)*, pages 475–479, 1988. *2 citations pages 36 and 101*

[DDLW14] Paul E. DUNNE, Wolfgang DVORÁK, Thomas LINSBICHLER, and Stefan WOLTRAN. « Characteristics of Multiple Viewpoints in Abstract Argumentation ». In *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 52–61, 2014. *7 citations pages 18, 19, 91, 95, 130, 137, and 139*

[DHL+15] Martin DILLER, Adrian HARET, Thomas LINSBICHLER, Stefan RÜMMELE, and Stefan WOLTRAN. « An Extension-Based Approach to Belief Revision in Abstract Argumentation ». In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2926–2932, 2015. *5 citations pages 93, 128, 132, 134, and 137*

[DHP14] Sylvie DOUTRE, Andreas HERZIG, and Laurent PERRUSSEL. « A Dynamic Logic Framework for Abstract Argumentation ». In *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 62–71, 2014. *9 citations pages 59, 107, 127, 128, 129, 131, 134, 136, and 137*

[DJWW11] Wolfgang DVORÁK, Matti JÄRVISALO, Johannes Peter WALLNER, and Stefan WOLTRAN, 2011. see `http://www.dbai.tuwien.ac.at/research/project/argumentation/dynpartix/examples/`. *Cited page 118*

[DJWW14] Wolfgang DVORÁK, Matti JÄRVISALO, Johannes Peter WALLNER, and Stefan WOLTRAN. « Complexity-Sensitive Decision Procedures for Abstract Argumentation ». *Artificial Intelligence*, 206:53–78, 2014. *Cited page 118*

[DKV15] Jérôme DELOBELLE, Sébastien KONIECZNY, and Srdjan VESIC. « On the Aggregation of Argumentation Frameworks ». In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2911–2917, 2015. *2 citations pages 95 and 140*

[DMT07] Phan Minh DUNG, Paolo MANCARELLA, and Francesca TONI. « Computing ideal sceptical argumentation ». *Artificial Intelligence*, 171(10-15):642–674, 2007. *Cited page 11*

[DMW12] Paul E. DUNNE, Pierre MARQUIS, and Michael WOOLDRIDGE. « Argument Aggregation: Basic Axioms and Complexity Results ». In *Proceedings of the Fourth International Conference on Computational Models of Argument (COMMA 2012)*, pages 129–140, 2012. *Cited page 140*

[Doy92] Jon DOYLE. Reason Maintenance and Belief Revision - Foundations vs. Coherence Theories. In Peter GÄRDENFORS, editor, *Belief Revision*, pages 29–51. 1992. *Cited page 69*

[DP11] James P. DELGRANDE and Pavlos PEPPAS. « Revising Horn Theories ». In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 839–844, 2011. *Cited page 93*

[DP13] Sylvie DOUTRE and Laurent PERRUSSEL. « On Enforcing a Constraint in Argumentation ». In *Proceedings of the Eleventh European Workshop on Multi-Agent Systems (EUMAS'13)*, 2013. *Cited page 136*

[Dun95] Phan Minh DUNG. « On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming, and n-Person Games ». *Artificial Intelligence*, 77(2):321–357, 1995. *8 citations pages 1, 4, 8, 9, 11, 125, 126, and 128*

[dV97] Alvaro del VAL. « Nonmonotonic Reasoning and Belief Revision: Syntactic, Semantic, Foundational, and Coherence Approaches ». *Journal of Applied Non-Classical Logics*, 7(2), 1997. *Cited page 69*

[DW09] Paul E. DUNNE and Michael WOOLDRIDGE. Complexity of Abstract Argumentation. In Guillermo SIMARI and Iyad RAHWAN, editors, *Argumentation in Artificial Intelligence*, Chapter 5, pages 85–104. 2009. *Cited page 16*

[DW11] Wolfgang DVOŘÁK and Stefan WOLTRAN. « On the Intertranslatability of Argumentation Semantics ». *Journal of Artificial Intelligence Research*, 41(2):445–475, 2011. *2 citations pages 130 and 137*

[EW06] Uwe EGLY and Stefan WOLTRAN. « Reasoning in Argumentation Frameworks Using Quantified Boolean Formulas ». In *Proceedings of the First International Conference on Computational Models of Argument (COMMA 2006)*, pages 133–144, 2006. *2 citations pages 108 and 139*

[Fer99] Eduardo FERMÉ. « *Revising the AGM Postulates* ». PhD thesis, Universidad de Buenos Aires, 1999. *Cited page 3*

[FKS02] Marcelo A. FALAPPA, Gabriele KERN-ISBERNER, and Guillermo Ricardo SIMARI. « Explanations, Belief Revision and Defeasible Reasoning ». *Artificial Intelligence*, 141(1/2):1–28, 2002. *Cited page 62*

[FKUV86] Ronald FAGIN, Gabriel M. KUPER, Jeffrey D. ULLMAN, and Moshe Y. VARDI. « Updating Logical Databases ». *Advances in Computing Research*, 3:1–18, 1986. *Cited page 80*

[For89] Kenneth D. FORBUS. « Introducing Actions into Qualitative Simulation ». In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI 1989)*, pages 1273–1278, 1989. *Cited page 39*

[Gär88] Peter GÄRDENFORS. *Knowledge In Flux*. Cambridge University Press, Cambridge, UK, 1988. *3 citations pages 28, 30, and 75*

[GJ79] M. R. GAREY and David S. JOHNSON. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. *2 citations pages 154 and 159*

[GM15] Davide GROSSI and Sanjay MODGIL. « On the Graded Acceptability of Arguments ». In *Proceedings of the 24th Internation Joint Conference on Artificial Intelligence (IJCAI 2015)*, 2015. *Cited page 128*

[GR12] Dov GABBAY and Odinaldo RODRIGUES. « A Numerical Approach to the Merging of Argumentation Networks ». In *Proceedings of the Thirteenth International Workshop on Computational Logic in Multi-Agent Systems (CLIMA 2012)*, pages 195–212, 2012. *Cited page 140*

[Gro88] Adam GROVE. « Two Modellings for Theory Change ». *Journal of Philosophical Logic*, 17:157–170, 1988. *2 citations pages 32 and 35*

[GRR98] Dov GABBAY, Odinaldo RODRIGUES, and Alessandra RUSSO. « Revision by Translation ». In *Proceedings of the Seventh International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 98)*, volume Information, Uncertainty and Fusion, pages 3–32, 1998. *Cited page 97*

[GT13] Simone GABBRIELLINI and Paolo TORRONI. « MS Dialogues: Persuading and Getting Persuaded, A Model of Social Network Debates that Reconciles Arguments and Trust ». In *Tenth International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2013)*, 2013. *Cited page 70*

[Ham50] Richard W. HAMMING. « Error Detecting and Error Correcting Codes ». *Bell System Technical Journal*, 29(2):147–160, 1950. *2 citations pages 36 and 173*

[Her14] Andreas HERZIG. « Belief Change Operations: A Short History of Nearly Everything, Told in Dynamic Logic of Propositional Assignments ». In *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, 2014. *Cited page 40*

[HLMT11] Andreas HERZIG, Emiliano LORINI, Frédéric MOISAN, and Nicolas TROQUARD. « A Dynamic Logic of Normative Systems ». In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 228–233, 2011. *Cited page 25*

[IBM14] IBM. « IBM ILOG CPLEX Optimization Studio: Optimization model development toolkit for mathematical and constraint programming », 2014. See `http://www.ibm.com/software/products/en/ibmilogcpleoptistud/`. *Cited page 117*

[Kar72] Richard M. KARP. « Reducibility Among Combinatorial Problems ». In *Proceedings of a symposium on the Complexity of Computer Computations*, pages 85–103, 1972. *Cited page 158*

[KBM+13] Dionysios KONTARINIS, Elise BONZON, Nicolas MAUDET, Alan PEROTTI, Leon van der TORRE, and Serena VILLATA. « Rewriting Rules for the Computation of Goal-Oriented Changes in an Argumentation System ». In *Proceedings of the Fourteenth International Workshop on Computational Logic in Multi-Agent Systems (CLIMA 2013)*, pages 51–68, 2013. *7 citations pages 2, 55, 57, 107, 122, 137, and 140*

[KBMM14] Dionysios KONTARINIS, Elise BONZON, Nicolas MAUDET, and Pavlos MORAITIS. « Empirical Evaluation of Strategies for Multiparty Argumentative Debates ». In *Fifteenth International Workshop on Computational Logic in Multi-Agent Systems (CLIMA 2014)*, pages 105–122, 2014. *Cited page 2*

[KLM90] Sarit KRAUS, Daniel J. LEHMANN, and Menachem MAGIDOR. « Nonmonotonic Reasoning, Preferential Models and Cumulative Logics ». *Artificial Intelligence*, 44(1-2):167–207, 1990. *2 citations pages 1 and 55*

[KM91] Hirofumi KATSUNO and Alberto O. MENDELZON. « Propositional Knowledge Base Revision and Minimal Change ». *Artificial Intelligence*, 52:263–294, 1991. *12 citations pages 2, 4, 25, 33, 34, 35, 75, 76, 100, 101, 131, and 133*

[KM92] Hirofumi KATSUNO and Alberto O. MENDELZON. On the Difference between Updating a Knowledge Base and Revising it. In Peter GÄRDENFORS, editor, *Belief Revision*, pages 183–203. 1992. *6 citations pages 2, 4, 25, 33, 37, and 134*

[KMM04] Antonis KAKAS, Nicolas MAUDET, and Pavlos MORAITIS. « Layered Strategies and Protocols for Argumentation-Based Agent Interaction ». In *Proceedings of the First International Workshop on Argumentation in MultiAgent Systems (ArgMAS 2004)*, pages 66–79, 2004. *Cited page 2*

[Kon00] Sébastien KONIECZNY. « On the Difference between Merging Knowledge Bases and Combining Them ». In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR 2000)*, pages 135–144, 2000. *2 citations pages 80 and 140*

[KP98] Sébastien KONIECZNY and Ramón PINO PÉREZ. « On the Logic of Merging ». In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR 1998)*, pages 488–498, 1998. *Cited page 140*

[KP99] Sébastien KONIECZNY and Ramón PINO PÉREZ. « Merging with Integrity Constraints ». In *Proceedings of the Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 1999)*, pages 233–244, 1999. *2 citations pages 94 and 140*

[LLM15a] Jean-Marie LAGNIEZ, Emmanuel LONCA, and Jean-Guy MAILLY. « CoQuiAAS : Applications de la programmation par contraintes à l'argumentation abstraite ». In *Onzièmes Journées Francophones de la Programmation par Contraintes (JPFC 2015)*, 2015. in french. *Cited page 139*

[LLM15b] Jean-Marie LAGNIEZ, Emmanuel LONCA, and Jean-Guy MAILLY. « CoQuiAAS: A Constraint-based Quick Abstract Argumentation Solver ». In *Twenty-seventh IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2015)*, 2015. *Cited page 139*

[LLM15c] Jean-Marie LAGNIEZ, Emmanuel LONCA, and Jean-Guy MAILLY. « CoQuiAAS: Applications of Constraint Programming to Abstract Argumentation ». In *First International Competition on Computational Models of Argumentation (ICCMA 2015)*, 2015. *Cited page 139*

[LM92] Daniel J. LEHMANN and Menachem MAGIDOR. « What does a Conditional Knowledge Base Entail? ». *Artificial Intelligence*, 55(1):1–60, 1992. *Cited page 55*

[Mai12] Jean-Guy MAILLY. « Révision de Systèmes d'Argumentation », 2012. Master thesis, in french. English title: *Revision of Argumentation Frameworks*. *Cited page 138*

[Mai13] Jean-Guy MAILLY. « Revising Argumentation Systems: Argument Status Versus Graph Minimization », 2013. ACAI Summer School 2013, Student Session (ACAI 2013). *no citation*

[NAD14] Samer NOFAL, Katie ATKINSON, and Paul E. DUNNE. « Algorithms for Decision Problems in Argument Systems under Preferred Semantics ». *Artificial Intelligence*, 207:23–51, 2014. *Cited page 108*

[Neb98] Bernhard NEBEL. How Hard is it to Revise a Belief Base? In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 3: Belief Change, pages 77–145. Kluwer, 1998. *Cited page 84*

[NOC07] Juan Carlos NIEVES, Mauricio OSORIO, and Ulises CORTÉS. « Inferring Preferred Extensions by Minimal Models ». pages 114–124, 2007. Workshop at Logic Programming and Non-Monotonic Reasonning 2007 (LPNMR07). *Cited page 108*

[NR11] Farid NOUIOUA and Vincent RISCH. « Argumentation Frameworks with Necessities ». In *Proceedings of the Fifth International Conference on Scalable Uncertainty Management (SUM 2011)*, pages 163–176, 2011. *2 citations pages 126 and 127*

[NW14] Farid NOUIOUA and Eric WÜRBEL. « Removed Set-Based Revision of Abstract Argumentation Frameworks ». In *Proceedings of the Twenty-Sixth IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2014)*, pages 784–791, 2014. *2 citations pages 62 and 137*

[OW11] Emilia OIKARINEN and Stefan WOLTRAN. « Characterizing Strong Equivalence for Argumentation Frameworks ». *Artificial Intelligence*, 175(14-15):1985–2009, 2011. *no citation*

[Pap92] Odile PAPINI. « A Complete Revision Function in Propositional Calculus ». In *Proceedings of the Tenth European Conference on Artificial Intelligence, (ECAI 1992)*, pages 339–343, 1992. *Cited page 63*

[Pap94] Christos H. PAPADIMITRIOU. *Computational Complexity*. Addison-Wesley, 1994. *Cited page 154*

[Pea09] Judea PEARL. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009. *Cited page 53*

[QLB06] Guilin QI, Weiru LIU, and David A. BELL. « Knowledge Base Revision in Description Logics ». In *Proceedings of the Tenth European Conference on Logics in Artificial Intelligence (JELIA 2006)*, pages 386–398, 2006. *Cited page 75*

[Rei80] Raymond REITER. « A Logic for Default Reasoning ». *Artificial Intelligence*, 13(1-2):81–132, 1980. *Cited page 1*

[Rie14] Tjitze RIENSTRA. « *Argumentation in Flux: Modelling Change in the Theory of Argumentation* ». PhD thesis, Université du Luxembourg and Université de Montpellier II, 2014. *3 citations pages 54, 107, and 137*

[Sto76] Larry J. STOCKMEYER. « The Polynomial-Time Hierarchy ». *Theoretical Computer Science*, 3(1):1–22, 1976. *Cited page 155*

[Tar30] Alfred TARSKI. « Fundamentale Begriffe der Methodologie der Deduktiven Wissenschaften ». *Monatshefte Fur Mathematik*, 37:361–404, 1930. *Cited page 149*

[TBS08] Fernando A. TOHMÉ, Gustavo Adrian BODANZA, and Guillermo Ricardo SIMARI. « Aggregation of Attack Relations: A Social-Choice Theoretical Analysis of Defeasibility Criteria ». In *Foundations of Information and Knowledge Systems, 5th International Symposium, FoIKS 2008, Pisa, Italy, February 11-15, 2008, Proceedings*, pages 8–23, 2008. *Cited page 140*

[Tou58] Stephen TOULMIN. *The Use of Argument*. Cambridge University Press, 1958. *Cited page 9*

[Tse68] Grigorii Samuilovich TSEITIN. On the Complexity of Derivation in Propositional Calculus. In A.O SLISENKO, editor, *Structures in Constructive Mathematics and Mathematical Logic, Part II, Seminars in Mathematics (translated from Russian)*, pages 115–125. Steklov Mathematical Institute, 1968. *Cited page 147*

[Tur36] Alan Mathison TURING. « On Computable Numbers, with an Application to the Entscheidungsproblem ». *Proceedings of the London Mathematical Society. Second Series*, 42:230–265, 1936. *Cited page 151*

[TV15] Matthias THIMM and Serena VILLATA. « First International Competition on Computational Models of Argumentation (ICCMA'15) », 2015. see http://argumentationcompetition.org/2015/. *2 citations pages 122 and 139*

[VBvdT11] Serena VILLATA, Guido BOELLA, and Leendert van der TORRE. « Attack Semantics for Abstract Argumentation ». In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 406–413, 2011. *Cited page 56*

[Win88] Marianne WINSLETT. « Reasoning about Action Using a Possible Models Approach ». In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI 1988)*, pages 89–93, 1988. *no citation*

[WJP00] Eric WÜRBEL, Robert JEANSOULIN, and Odile PAPINI. « Revision: an Application in the Framework of GIS ». In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR 2000)*, pages 505–515, 2000. *Cited page 63*

[Yag88] Ronald R YAGER. « On Ordered Weighted Averaging Aggregation Operators in Multi-Criteria Decision Making ». *IEEE Transactions on Systems, Man and Cybernetics*, 18:183–190, 1988. *Cited page 83*

Abstract

This thesis tackles the problem of integrating a new piece of information in an abstract argumentation framework. Such a framework is a directed graph such that its nodes represent the arguments, and the directed edges represent the attacks between arguments. There are different ways to decide which arguments are accepted by the agent who uses such a framework to represent her beliefs.
An agent may be confronted with a piece of information such that "this argument should be accepted", which is in contradiction with her current beliefs, represented by her argumentation framework.
In this thesis, we have studied several approaches to incorporate a piece of information in an argumentation framework.
Our first contribution is an adaptation of the AGM framework for belief revision, which has been developed for characterizing the incorporation of a new piece of information when the agent's beliefs are represented in a logical setting. We have adapted the rationality postulates from the AGM framework to characterize the revision operators suited to argumentation frameworks, and we have identified several ways to generate the argumentation frameworks resulting from the revision.
We have also shown how to use AGM revision as a tool for revising argumentation frameworks. Our approach uses a logical encoding of the argumentation framework to take advantage of the classical revision operators, for deriving the expected result.
At last, we have studied the problem of enforcing a set of arguments (how to change an argumentation framework so that a given set of arguments becomes an extension). We have developed a new family of operators which guarantee the success of the enforcement process, contrary to the existing approaches, and we have shown that a translation of our approaches into satisfaction and optimization problems makes possible to develop efficient tools for computing the result of the enforcement.

Keywords: abstract argumentation, belief revision, minimal change, logical encoding.

Résumé

Cette thèse traite du problème de l'intégration d'une nouvelle information dans un système d'argumentation abstrait. Un tel système est un graphe orienté dont les nœuds représentent les arguments, et les arcs représentent les attaques entre arguments. Il existe divers moyen de décider quels arguments sont acceptés par l'agent qui utilise un tel système pour représenter ses croyances.
Il peut arriver dans la vie d'un agent qu'il soit confronté à une information du type "tel argument devrait être accepté", alors que c'est en contradiction avec ses croyances actuelles, représentées par son système d'argumentation.
Nous avons étudié dans cette thèse diverses approches pour intégrer une information à un système d'argumentation.
Notre première contribution est une adaptation du cadre AGM pour la révision de croyances, habituellement utilisé lorsque les croyances de l'agent sont représentées dans un formalisme logique. Nous avons notamment adapté les postulats de rationalité proposés dans le cadre AGM pour pouvoir caractériser des opérateurs de révision de systèmes d'argumentation, et nous avons proposé différents moyens de générer les systèmes d'argumentation résultant de la révision.

Nous avons ensuite proposé d'utiliser la révision AGM comme un outil pour réviser les systèmes d'argumentation. Il s'agit cette fois-ci d'une approche par encodage en logique du système d'argumentation, qui permet d'utiliser les opérateurs de révision usuels pour obtenir le résultat souhaité.
Enfin, nous avons étudié le problème du forçage d'un ensemble d'arguments (comment modifier le système pour qu'un ensemble donné soit une extension). Nous avons proposé une nouvelle famille d'opérateurs qui garantissent le succès de l'opération, contrairement aux opérateurs de forçage existants, et nous avons montré qu'une traduction de nos approches en problèmes de satisfaction ou d'optimisation booléenne permet de développer des outils efficaces pour calculer le résultat du forçage.

Mots-clés: argumentation abstraite, révision de croyances, changement minimal, encodage logique.