Révision de systèmes d'argumentation

MÉMOIRE

soutenu le 3 juillet 2012

pour l'obtention du

Master Recherche de l'Université d'Artois

Spécialité Informatique – Option Sytèmes Intelligents et Applications

par

Jean-Guy Mailly

Encadrants :	Sylvie Coste-Marquis	Maître de Conférences
	Sébastien Konieczny	Directeur de Recherche CNRS
	Pierre Marquis	Professeur des Universités

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

Je souhaite en premier lieu remercier Sylvie Coste-Marquis, Sébastien Konieczny et Pierre Marquis d'avoir accepté de m'encadrer, et pour leur aide précieuse pendant le stage. Il aurait été difficile de réussir ce stage sans leurs conseils pour m'orienter dans mon travail, tout au long de ces quelques mois.
De plus, je tiens à les remercier de m'avoir proposé d'assister aux Journées Francophone de la Programmation par Contraintes, et de l'Intelligence Artificielle Fondamentale, deux conférences organisées conjointement à Toulouse cette année, et qui m'ont permis de découvrir au fil des exposés de nouveaux domaines très intéressants.

Ensuite, je voudrais remercier l'ensemble du corps enseignant. Ces cinq années de Licence et Master à la Faculté Jean Perrin m'ont donné un goût de l'informatique et de la recherche que je ne me connaissais pas quand je suis sorti du lycée.

Enfin, je voudrais remercier mes camarades, étudiants de Master SIA ou doctorants, pour la bonne ambiance qui rend le travail plus agréable encore. Merci en particulier à Samy, mon coéquipier de robotique, et Manu et Long pour l'accueil dans leur bureau.

Table des matières

Table des figures **vi**

Introduction générale **1**

Partie I Etat de l'art

Chapitre 1 Systèmes d'argumentation **5**

1.1 L'argumentation selon Dung 6

1.1.1 Cadre formel 6

1.1.2 Extensions 7

1.1.3 Coïncidence des différentes sémantiques 9

1.2 Une autre façon de voir l'argumentation : les *labellings* 10

1.2.1 Introduction de la notion 10

1.2.2 Des extensions aux *labellings*, et vice-versa 11

1.3 Inférence à partir d'un système d'argumentation 14

1.4 Equivalence entre systèmes d'argumentation 15

1.4.1 Plusieurs critères d'équivalence 15

1.4.2 L'équivalence forte 16

1.4.3 L'équivalence locale 17

Chapitre 2 La dynamique des croyances **19**

2.1 Présentation du cadre AGM 20

2.1.1 Préliminaires 20

2.1.2 Postulats de rationalité 21

2.1.3 Théorèmes de représentation 24

2.2 Révision dans un cadre particulier : la logique propositionelle 27

2.2.1 Postulats pour la révision en logique propositionnelle 28

2.2.2 Théorème de représentation : les assignements fidèles 28

Chapitre 3 Le changement dans les systèmes d'argumentation **31**

3.1 Ajout d'un argument 32

3.1.1 Opérations de changement en argumentation 32

3.1.2 Propriétés de structure 33

3.1.3 Propriétés sur l'état des arguments 34

3.1.4 Liens entre les propriétés 35

3.1.5 L'ajout d'argument sous les sémantiques de base et préférée 36

3.2 Retrait d'un argument 39

3.2.1 Un type de changement adapté au retrait d'un argument 39

3.2.2 Caractériser la monotonie d'un retrait 40

3.2.3 Diverses propriétés sur le changement expansif et le changement restrictif . 40

3.3 Principes d'abstraction et de raffinement 40

3.3.1 Le cadre formel de Baroni et Giacomin 41

3.3.2 Principes d'abstraction d'attaque 42

3.3.3 Principes d'abstraction d'argument 43

3.3.4 Raffinement d'attaque 45

3.4 Ce qu'on attend de la révision 46

Partie II Etude de la révision de systèmes d'argumentation

Chapitre 1 Définition formelle de la révision de systèmes d'argumentation **49**

1.1 Définition de la révision 50

1.1.1 Formules sur les arguments : un langage pour exprimer les informations inférées à partir d'un système d'argumentation . . . 50
1.1.2 Formalisation de la révision de systèmes d'argumentation . . . 52
1.1.3 Partir des *labellings* pour définir la dynamique dans les systèmes d'argumentation . . . 52
1.2 Postulats pour la révision de systèmes d'argumentation . . . 55
1.2.1 Des postulats dans d'autres cadres . . . 55
1.2.2 Traduction des postulats algébriques . . . 55
1.2.3 Une autre notion intéressante : l'indépendance . . . 58
1.2.4 Postulats de rationalité . . . 58

Chapitre 2 Propriétés intéressantes concernant la révision **61**
2.1 Définition d'opérateurs de révision partiels . . . 62
2.1.1 Première idée : α ne doit pas être attaqué . . . 62
2.1.2 Deuxième idée : α ne doit pas être attaqué, sauf s'il est défendu . . . 63
2.1.3 Révision par systèmes candidats . . . 64
2.1.4 Quelques propriétés intéressantes . . . 68
2.2 Relations entre des opérateurs particuliers et les postulats . . . 70
2.2.1 Ajout d'attaques . . . 70
2.2.2 Retrait d'attaques . . . 71
2.2.3 Opérateur drastique . . . 72
2.2.4 Opérateurs mixtes . . . 73
2.3 Différentes distances pour différentes minimalités . . . 74
2.3.1 Distance sur le graphe . . . 75
2.3.2 Distance de conséquence . . . 75
2.3.3 Non-équivalence des distances . . . 75

Conclusion et perspectives **79**

Bibliographie

Table des figures

1.1 Exemple d'attaques entre arguments . . . 6
1.2 Recherche des extensions préférées pour un système d'argumentation . . . 7
1.3 Hiérarchie des sémantiques d'extensions . . . 9
1.4 L'argument d influence l'acceptation de a . . . 10
1.5 Une extension préférée n'implique pas nécessairement un $undec$ minimal . . . 14

2.1 Transitions entre statuts épistémiques . . . 20
2.2 Révision de K par α via un système de sphères centré sur $[K]$ . . . 28
2.3 Révision de K par α . . . 29

3.1 Exemple d'attaques entre arguments . . . 36

1.1 a et c sont acceptés car a défend c . . . 53
1.2 Ici a et c ne sont pas attaqués . . . 53
1.3 Transitions entre statuts d'acceptation . . . 54

2.1 Exemple de révision non minimale . . . 63
2.2 Système d'argumentation à réviser par α . . . 63
2.3 Echec de la révision à cause d'un cycle de longueur impaire . . . 64
2.4 Déroulement de l'algorithme 1 sous forme d'arbre . . . 65
2.5 Système d'argumentation à réviser par α . . . 67
2.6 Un système « candidat » qui n'a que l'extension $\varnothing$ . . . 68
2.7 α ne peut pas changer de statut juste avec des retraits . . . 69
2.8 β ne peut pas être in juste avec des ajouts . . . 69
2.9 α ne peut pas être out juste avec des ajouts . . . 69
2.10 Le système accepte a et rejette b . . . 70
2.11 Le système n'accepte ni a ni b . . . 71
2.12 Une extension du système satisfait $\varphi = b \wedge c$ . . . 73
2.13 Deux systèmes dont le graphe est proche, mais pas les conséquences . . . 76
2.14 Deux systèmes dont les conséquences sont proches, mais pas le graphe . . . 76

Introduction générale

Présentation du thème

L'intelligence artificielle est un domaine d'étude lié à l'informatique et aux mathématiques dont le but est de modéliser et d'implémenter des processus permettant à un programme d'agir « comme un humain ». Un exemple connu permettant de définir ce domaine est le test de Turing : un programme est doté d'une « bonne intelligence » s'il peut tromper un humain, c'est-à-dire si un humain qui interragit avec lui n'est pas capable de se rendre compte que son interlocuteur est une machine.

L'imitation du comportement humain nécessite de modéliser un grand nombre de types de raisonnements que l'homme peut tenir. Les deux qui nous ont intéressé sont l'argumentation et la révision.

L'argumentation est la capacité de confronter des idées, par exemple, pour convaincre un interlocuteur ou pour prendre une décision. Une autre utilisation possible de l'argumentation est la représentation des croyances d'un agent rationnel. En effet, les croyances qu'a un agent sur son environnement peuvent conduire à une contradiction si elles sont toutes considérées comme vraies ; dans ce cas l'argumentation peut être un moyen de sélectionner parmi ces croyances les plus acceptables, les plus fiables pour représenter le monde.

L'argumentation abstraite a été formalisée par Dung dans [Dun95] : un système d'argumentation peut être vu comme un graphe dirigé dont les noeuds représentent les arguments, et les arcs représentent les attaques entre les arguments. Diverses sémantiques permettent de retenir des ensembles d'arguments qui peuvent être acceptés conjointement sans causer d'incohérence, et l'intersection de ces ensembles correspond à l'ensemble des arguments qui peuvent être acceptés à coup sûr.

La révision de croyances est le processus suivi par un agent rationnel lorsqu'il apprend une nouvelle information sur l'état du monde, et que cette information n'est pas cohérente avec ses croyances précédentes.

La révision a déjà été étudiée dans plusieurs cadres, les deux plus classiques étant le cadre AGM ([AGM85]) et la logique propositionnelle ([KM91]), mais pas dans le cadre où les croyances sont exprimées au moyen de systèmes d'argumentation.

Motivations et présentation du travail

Le changement dans les systèmes d'argumentation a été étudié dans un certain nombre de travaux au cours des dernières années, cependant aucun de ces travaux ne correspond à de la révision. Globalement, il s'agit d'études de l'effet de l'ajout ou du retrait d'un argument, en assurant un certain résultat en fonction des caractéristiques de la modification faite. Par exemple, on sait que si l'on ajoute un argument qui attaque certains des arguments déjà présents, mais que ce nouvel argument n'est pas attaqué, alors il sera accepté et ceux qu'il attaque seront refusés.

Cependant, ce genre de raisonnement ne semble pas satisfaisant pour une opération de révision. En effet, pour faire l'analogie avec la révision en logique propositionnelle, l'ajout (ou le retrait) d'un argument semble plus proche d'une modification du langage, c'est-à-dire ajouter à la base de connaissance une formule comportant de nouvelles variables. C'est une extension du monde connu par l'agent, plutôt qu'une révision de ses croyances sur le monde.
De plus, les travaux précédents ne présentent pas de postulats de rationnalité, qui sont des propriétés qu'on souhaite voir vérifiées par tout opérateur de révision, pour assurer que le changement dans la base de croyance de l'agent correspond au changement que ferait un être humain.
Enfin, le coeur du concept de révision est de pouvoir faire accepter une croyance qui était rejetée, ou l'inverse, ce qui n'est absolument pas le propos des travaux existants.

La proposition qui est faite dans ce mémoire est qu'il ne faut pas modifier l'ensemble d'arguments du système d'argumentation, mais plutôt modifier la relation d'attaque. On considère alors que l'information « l'argument α attaque l'argument β » est une croyance qui peut être changée, au même titre qu'on peut réviser une formule en logique propositionnelle. Ainsi, la problématique est « comment modifier la relation d'attaque d'un système d'argumentation pour changer le statut d'un argument particulier ? ».

Organisation du mémoire

Ce mémoire débute par un bref état de l'art. Après une présentation des systèmes d'argumentation de Dung et de notions liées, on présente le cadre classique de la dynamique des croyances : le cadre AGM et la révision en logique propositionnelle. Cette première partie s'achève par une présentation de quelques travaux existants sur le changement dans les systèmes d'argumentation.

La seconde partie est le coeur du travail effectué. Dans un premier chapitre, on définit un langage permettant d'exprimer une information inférée par un système d'argumentation, et on utilise ce langage pour définir formellement la révision de systèmes d'argumentation. Une fois cette définition donnée, on présente des postulats de rationnalité adaptés à la révision de systèmes d'argumentation. La particularité de ces définitions est qu'on peut associer plusieurs sémantiques au langage défini, et on construit autant de familles de postulats qu'il y a de sémantiques.
On s'intéresse ensuite à diverses propriétés liées à la révision de systèmes d'argumentation, d'abord sur des exemples concrets d'opérateurs, puis en comparant des classes d'opérateurs aux postulats. Enfin, on étudie la notion de changement minimal dans le cadre de l'argumentation.

Première partie

Etat de l'art

Chapitre 1

Systèmes d'argumentation

L'argumentation constitue un point essentiel de l'intelligence humaine. Il s'agit d'une aptitude importante pour comprendre un problème dans sa globalité, le résoudre via un raisonnement scientifique, et pour exprimer et défendre ses idées dans la vie quotidienne, et prendre les décisions qui peuvent en découler. L'étude de l'argumentation a donc pleinement sa place en intelligence artificielle.

On peut représenter de manière formelle l'argumentation dans plusieurs buts. Le premier est de décider, parmi les arguments (éventuellement contradictoires) avancés par plusieurs agents, lesquels sont acceptables pour refléter correctement le monde réel. Cette décision peut conduire à un choix d'action à réaliser par un ou des agents. Un autre but peut être de modéliser les croyances d'un seul agent. En effet, celles ci peuvent aussi être contradictoires entre elles, et l'agent peut passer par un système d'argumentation pour décider quelles croyances il va retenir comme plus valables.

Pour un système d'argumentation donné, on associe, selon une certaine sémantique, un ensemble d'extensions, c'est-à-dire d'ensembles d'arguments considérés comme assez bons pour être acceptés. Une autre notion intéressante est celle de *labelling* (étiquette), qui consiste en un étiquetage des arguments par leur valeur d'acceptation.

Nous allons présenter dans ce chapitre les notions de bases de la théorie de l'argumentation abstraite, telles que définies par Dung. Dans une seconde partie, nous introduirons la notion plus récente de *labelling*. Enfin, nous nous intéresserons à la notion d'équivalence dans le cadre de l'argumentation, dont la définition n'est pas évidente.

Sommaire

1.1 L'argumentation selon Dung . . . 6
- 1.1.1 Cadre formel . . . 6
- 1.1.2 Extensions . . . 7
- 1.1.3 Coïncidence des différentes sémantiques . . . 9

1.2 Une autre façon de voir l'argumentation : les *labellings* . . . 10
- 1.2.1 Introduction de la notion . . . 10
- 1.2.2 Des extensions aux *labellings*, et vice-versa . . . 11

1.3 Inférence à partir d'un système d'argumentation . . . 14

1.4 Equivalence entre systèmes d'argumentation . . . 15
- 1.4.1 Plusieurs critères d'équivalence . . . 15
- 1.4.2 L'équivalence forte . . . 16
- 1.4.3 L'équivalence locale . . . 17

1.1 L'argumentation selon Dung

De façon basique, on peut expliquer le fonctionnement de l'argumentation en disant que celui qui gagne est celui qui a le dernier mot, au sens propre comme au figuré. Illustrons cette idée par un exemple.

Exemple 1 (Débat)**.**
Les individus a et b ne sont pas d'accord à propos de la voiture qui est la plus intéressante. a pense que c'est la voiture V_1, b pense que c'est la voiture V_2.
- a : Ma voiture est meilleure que la tienne, car elle est plus confortable.
- b : La mienne est meilleure, car elle est plus écologique.
- a : C'est vrai, mais la tienne coûte beaucoup plus cher que la mienne.

Si la discussion s'arrête ici, il peut sembler que a a imposé son point de vue. Pourtant, si b donne un nouvel argument contre celui de a, alors c'est b qui « remporte le match ».

Ce genre de discussion peut aisément être représenté de façon formelle. Dans cette partie, nous présentons le formalisme de Dung pour représenter l'argumentation, et la notion d'extension qui permet de décider, pour un système d'argumentation, quels arguments peuvent être considérés comme « vainqueurs » du débat.

1.1.1 Cadre formel

Un système d'argumentation permet de représenter des arguments (discussions entre agents, ou croyances conflictuelles d'un agent) au moyen de la notion d'attaque. Un argument « attaque » un autre argument si le premier permet d'invalider le second. Il n'y a pas besoin de connaître la structure d'un argument, il peut être vu comme un simple noeud dans un graphe.

Définition 1 (Système d'argumentation [Dun95])**.**
Un système d'argumentation est une paire $\langle \mathcal{A}, \mathcal{R} \rangle$ où $\mathcal{A}$ est un ensemble d'arguments et $\mathcal{R}$ une relation binaire sur $\mathcal{A}$, c'est-à-dire $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$.

On appelle $\mathcal{R}$ la relation d'attaque : $(a, b) \in \mathcal{R}$ signifie « l'argument a attaque l'argument b ».

Exemple 2 (Attaques entre arguments)**.**
- a : « Il a plu aujourd'hui. »
- b : « C'est faux, Météo France annonçait du soleil. »
- c : « J'ai vu la pluie tomber, Météo France se trompe. »

$$a \longleftarrow b \longleftarrow c$$

FIGURE 1.1 – Exemple d'attaques entre arguments

Comme résultat de cette discussion, on conserve l'argument c et l'argument a (c n'est pas attaqué, a est défendu par c) tandis qu'on rejette l'argument b (qui est défait par c).

Il est essentiel, pour retenir un ensemble d'arguments qu'on accepte, que cet ensemble présente une certaine cohérence, pour cela il ne faut pas qu'un de ses arguments en attaque un autre.

Définition 2 (Ensemble sans conflit)**.**
Un ensemble d'arguments $S \subseteq \mathcal{A}$ est dit sans conflit s'il n'y a pas d'arguments $a, b \in S$ tels que $(a, b) \in \mathcal{R}$.

Exemple 3 (Ensemble avec ou sans conflit)**.**
Si on reprend le système d'argumentation donné par le graphe de l'exemple 2, l'ensemble $\{a, b\}$ n'est pas sans conflit, puisque b attaque a, par contre $\{a, c\}$ est sans conflit : ni (a, c), ni (c, a), ni les boucles (a, a) et (c, c) n'appartiennent à $\mathcal{R}$.

On peut étendre la notion d'attaque par un argument à l'attaque par un ensemble d'arguments. De façon assez évidente, on dit qu'un argument est attaqué par un ensemble s'il est attaqué par un des éléments de l'ensemble.

Définition 3 (Ensemble attaquant un argument)**.**
Un ensemble d'arguments $S \subseteq \mathcal{A}$ attaque $b \in \mathcal{A}$ si et seulement si $\exists a \in S$ tel que $(a, b) \in \mathcal{R}$.

Exemple 4 (Attaque par un ensemble)**.**
Dans le système d'argumentation de l'exemple 2, l'ensemble $\{a, c\}$ attaque b, parce que $(c, b) \in \mathcal{R}$.

Un agent rationnel accepte un argument s'il n'est pas attaqué, ou s'il peut être défendu contre tout argument l'attaquant. Par conséquent, l'ensemble de tous les arguments acceptés par un agent est un ensemble d'arguments qui peut se défendre lui même. On introduit donc les définitions suivantes d'argument *acceptable* et d'ensemble d'arguments *admissible*.

Définition 4 (Argument acceptable)**.**
Un argument $a \in \mathcal{A}$ est acceptable pour un ensemble S d'arguments si et seulement si $\forall b \in \mathcal{A}$, si $(b, a) \in \mathcal{R}$, alors b est attaqué par S.

Définition 5 (Ensemble admissible)**.**
Un ensemble d'arguments $S \subseteq \mathcal{A}$, sans conflit, est admissible si et seulement si tout argument $a \in S$ est acceptable pour S.

Exemple 5.
Dans le système d'argumentation de l'exemple 2, l'argument a est acceptable pour l'ensemble $\{a, c\}$ car c défend a contre b. De plus, c n'est pas attaqué. Donc $\{a, c\}$ est admissible.

1.1.2 Extensions

Dung ([Dun95]) présente plusieurs sémantiques d'acceptabilité, basées sur différentes notions d'extensions, plus ou moins contraintes, qui permettent d'obtenir un ensemble d'ensembles d'arguments qui seront potentiellement acceptés par l'agent.

Extensions préférées

Une première sémantique d'extensions est la sémantique préférée.

Définition 6 (Extension préférée)**.**
Une extension préférée d'un système d'argumentation est un ensemble admissible du système d'argumentation maximal pour l'inclusion.

Exemple 6 (Exemple d'extensions préférées)**.**

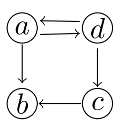

FIGURE 1.2 – Recherche des extensions préférées pour un système d'argumentation

Pour le système d'argumentation représenté à la figure 1.2, les ensembles admissibles sont : $\{a\}$, $\{d\}$, $\{a, c\}$ et $\{d, b\}$. Les extensions préférées sont donc $\{a, c\}$ et $\{d, b\}$.

Dung a montré que la sémantique préférée est définie pour tout système d'argumentation :

Proposition 1.
Tout système d'argumentation possède au moins une extension préférée.

Extensions stables

Dung présente ensuite la notion d'extension stable. Il a montré que cette notion coïncide avec la notion de solution stable de jeux à n-joueurs ([vM80]). Elle coïncide aussi avec plusieurs autres notions, comme les modèles stables en programmation logique, les extensions en logique des défauts de Reiter ([Rei80]) ou les expansions stables dans la logique autoépistémique de Moore ([Moo85]).

Définition 7 (Extension stable)**.**
Un ensemble d'arguments S sans conflit est une extension stable si et seulement si S attaque tout argument $a \notin S$, c'est-à-dire $S = \{a \in A | a$ n'est pas attaqué par $S\}$

Exemple 7 (Exemple d'extensions stables)**.**
En reprenant le système d'argumentation de l'exemple 6, les extensions stables sont $\{a, c\}$ et $\{b, d\}$, car $\{a, c\}$ attaque $\mathcal{A}\backslash\{a, c\} = \{b, d\}$, et $\{b, d\}$ attaque $\mathcal{A}\backslash\{b, d\} = \{a, c\}$.

Proposition 2.
Toute extension stable est préférée, mais l'inverse n'est pas vrai.

Extensions de base et complètes

Dung introduit une fonction qui associe à tout ensemble d'arguments S l'ensemble des arguments acceptables pour S.

Définition 8 (Fonction caractéristique)**.**
Soit AF un système d'argumentation. On définit la fonction caractéristique de AF par :

$$\begin{array}{rccl} \mathcal{F}_{AF} : & 2^{\mathcal{A}} & \rightarrow & 2^{\mathcal{A}} \\ & S & \mapsto & \{a | a \text{ est acceptable pour } S\} \end{array}$$

Pour se référer à un système d'argumentation quelconque mais fixé, on écrira $\mathcal{F}$ au lieu de $\mathcal{F}_{AF}$.

Proposition 3.
Un ensemble d'arguments sans conflit S est admissible si et seulement si $S \subseteq \mathcal{F}(S)$.

Dung définit la notion d'extension de base à partir de la fonction caractéristique du système d'argumentation.

Définition 9 (Extension de base)**.**
L'extension de base d'un système d'argumentation AF, notée GE_{AF}, est le plus petit point fixe de $\mathcal{F}_{AF}$.

La notion suivante fait le lien entre les extensions préférées et de base.

Définition 10 (Extension complète)**.**
Un ensemble d'arguments S, admissible, est une extension complète si et seulement si chaque argument qui est acceptable pour S appartient à S. Un ensemble sans conflit est une extension complète si et seulement si $S = \mathcal{F}(S)$.

Intuitivement, la notion d'extension complète peut être vue comme une forme de confiance en soi de l'agent : s'il est rationnel, il va accepter chaque argument qu'il peut défendre.

Théorème 4 ([Dun95])**.**

- *Chaque extension préférée est une extension complète, mais l'inverse n'est pas vrai.*
- *L'extension de base est la plus petite extension complète, au sens de l'inclusion.*

$$\begin{tikzcd}
\text{Stable} \arrow[dr] & & & \\
& \text{Préférée} \arrow[dr] & & \text{De base} \arrow[dl] \\
& & \text{Complète} &
\end{tikzcd}$$

FIGURE 1.3 – Hiérarchie des sémantiques d'extensions

1.1.3 Coïncidence des différentes sémantiques

Système d'argumentation bien fondé

Des conditions suffisantes ont été identifiées pour que les sémantiques de base, préférée et stable coïncident.

Premièrement, Dung définit la notion de système bien fondé. Un système d'argumentation est bien fondé s'il n'existe pas de suite infinie d'arguments qui attaquent leur prédécesseur.

Définition 11 (Système d'argumentation bien fondé)**.**
Un système d'argumentation est bien fondé si et seulement si il n'existe pas de suite infinie $\alpha_0, \alpha_1, \dots, \alpha_n, \dots$ tel que $\forall i, (\alpha_i, \alpha_{i+1}) \in \mathcal{R}$.

Un tel système présente la propriété intéressante de n'avoir qu'une extension, ce qui rend plus simple le choix des arguments acceptés.

Théorème 5 ([Dun95])**.**
Chaque système d'argumentation bien fondé a exactement une extension qui est de base, préférée et stable.

Système d'argumentation cohérent

Ensuite, Dung a donné une condition pour que les extensions stables et préférées coïncident (indépendemment de l'extension de base). En général, l'existence d'une extension préférée qui n'est pas stable signifie qu'il y a certaines « anomalies » dans le système d'argumentation correspondant. Par exemple, le système $\langle \{\alpha\}, \{(\alpha, \alpha)\} \rangle$ a une extension préférée qui n'est pas stable : $\varnothing$.

Définition 12 (Système d'argumentation cohérent, relativement de base)**.**

1. Un système d'argumentation AF est dit cohérent si chaque extension de AF est stable
2. Un système d'argumentation AF est relativement de base si son extension de base coïncide avec l'intersection de toutes ses extensions préférées.

Reprenons l'exemple 2. Dans ce cas, si l'on ajoute un argument d qui attaque c, d a toute son importance pour déterminer si on accepte a ou pas, et pourtant d n'attaque pas a.

$$a \longleftarrow b \longleftarrow c \longleftarrow d$$

FIGURE 1.4 – L'argument d influence l'acceptation de a

On peut aussi définir des notions d'attaques et de défense indirectes.

Définition 13 (Attaque indirecte)**.**
Un argument β attaque indirectement α si il existe une suite finie de longueur paire $\alpha_0, \alpha_1, \ldots, \alpha_{2n+1}$ telle que :

1. $\alpha = \alpha_0$, $\beta = \alpha_{2n+1}$
2. $\forall i, 0 \leqslant i \leqslant 2n$, $(\alpha_{i+1}, \alpha_i) \in \mathcal{R}$.

Définition 14 (Défense indirecte)**.**
Un argument β défend indirectement α si il existe une suite finie de longueur impaire $\alpha_0, \alpha_1, \ldots, \alpha_{2n}$ telle que :

1. $\alpha = \alpha_0$, $\beta = \alpha_{2n}$
2. $\forall i, 0 \leqslant i \leqslant 2n$, $(\alpha_{i+1}, \alpha_i) \in \mathcal{R}$.

Il est possible qu'il existe deux telles suites, entre α et β, de longueur paire pour l'une et impaire pour l'autre. Auquel cas β est à la fois défenseur et attaquant (indirect) de α.

Définition 15 (Argument controversé)**.**
Un argument β est dit controversé vis-à-vis de α si on a à la fois β attaque indirectement α et β défend indirectement α.
Un argument est controversé s'il est controversé vis-à-vis d'un quelconque argument.

Définition 16 (Système d'argumentation sans controverse, faiblement controversé)**.**

1. Un système d'argumentation est sans controverse si aucun de ses arguments n'est controversé.
2. Un système d'argumentation est faiblement controversé si il n'existe pas de suite infinie $\alpha_0, \alpha_1, \ldots, \alpha_n, \ldots$ telle que α_{i+1} est controversé vis-à-vis de α_i.

Clairement, un système sans controverse est un cas particulier de système faiblement controversé.

Théorème 6 ([Dun95])**.**

1. *Tout système d'argumentation faiblement controversé est cohérent.*
2. *Tout système d'argumentation sans controverse est cohérent et relativement basé.*

1.2 Une autre façon de voir l'argumentation : les *labellings*

1.2.1 Introduction de la notion

Caminada ([Cam06]) définit la notion de *labelling* comme l'association d'une étiquette qui indique si un argument est accepté, rejeté ou indéfini.

Définition 17 (AF-*labelling*)**.**
Soit $\langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation. Un AF-*labelling*est une fonction (totale)

$$\mathcal{L} : \mathcal{A} \longrightarrow \{in, out, undec\}$$

On note
- $in(\mathcal{L}) = \{a \in \mathcal{A} | L(a) = in\}$,
- $out(\mathcal{L}) = \{a \in \mathcal{A} | L(a) = out\}$,
- $undec(\mathcal{L}) = \{a \in \mathcal{A} | L(a) = undec\}$.

Cette simple définition n'impose rien concernant la façon d'étiqueter les arguments. La notion de *reinstatement labelling* permet d'obtenir des étiquetages qui sont cohérents avec les attaques.

Définition 18 (*Reinstatement labelling*)**.**
Soit $\mathcal{L}$ un AF-*labelling*. On dit que $\mathcal{L}$ est un *reinstatement labelling* si et seulement si il satisfait
- $\forall a \in \mathcal{A}, \mathcal{L}(a) = out$ si et seulement si $\exists b \in \mathcal{A} : (b, a) \in \mathcal{R}$ et $\mathcal{L}(b) = in$
- $\forall a \in \mathcal{A}, \mathcal{L}(a) = in$ si et seulement si $\forall b \in \mathcal{A} : (b, a) \in \mathcal{R} \Rightarrow \mathcal{L}(b) = out$

Ainsi, on accepte un argument qui n'est pas attaqué, ou seulement par des arguments qui sont rejetés. Et symétriquement, un argument est rejeté s'il existe un argument accepté qui l'attaque.

1.2.2 Des extensions aux *labellings*, et vice-versa

Caminada définit ensuite deux fonctions qui permettent, pour un système d'argumentation donné, de convertir un ensemble d'arguments en *labelling* et réciproquement. La première fonction, $\mathrm{Ext2Lab}_{(\mathcal{A},\mathcal{R})}$, a en entrée un ensemble d'arguments (éventuellement une extension) et le convertit en *labelling*. La fonction $\mathrm{Lab2Ext}_{(\mathcal{A},\mathcal{R})}$ a en entrée un AF-*labelling* et le convertit en ensemble d'arguments. Dans la définition suivante, le *labelling* ne satisfait pas forcément les propriétés pour être un *reinstatement labelling*, et l'ensemble d'arguments n'est pas forcément une extension.

Définition 19 ([Cam06])**.**
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, $Args \subseteq \mathcal{A}$ tel que $Args$ est sans conflit, et $\mathcal{L} : \mathcal{A} \longrightarrow \{in, out, undec\}$ un AF-*labelling*. On définit
- $\mathrm{Ext2Lab}_{(\mathcal{A},\mathcal{R})}(Args) = \{(a, in) | a \in Args\} \cup \{(a, out) | \exists b \in Args : (b, a) \in \mathcal{R}\} \cup \{(a, undec) | a \notin Args$ et $\nexists b \in Args : (b, a) \in \mathcal{R}\}$
- $\mathrm{Lab2Ext}_{(\mathcal{A},\mathcal{R})}(\mathcal{L}) = \{a | (a, in) \in \mathcal{L}\}$

Pour simplifier l'écriture, on écrira par la suite $\mathrm{Ext2Lab}$ et $\mathrm{Lab2Ext}$ quand le système d'argumentation considéré est clairement identifié.

Dans la suite de cette section, on présente les différents cas où un AF-*labelling* correspond à un extension particulière du système d'argumentation.

Relation avec les extensions complètes

On remarque tout d'abord que la notion de *reinstatement labelling* correspond à la notion d'extension complète de Dung, comme le présentent les deux théorèmes suivants.

Théorème 7.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation et $\mathcal{L}$ un reinstatement labelling. Alors $\mathrm{Lab2Ext}(\mathcal{L})$ est une extension complète.

Théorème 8.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $Args$ *une extension complète. Alors* $\mathrm{Ext2Lab}(Args)$ *est un reinstatement labelling.*

De plus, Caminada observe que si on réduit les domaines de $\mathrm{Lab2Ext}$ et $\mathrm{Ext2Lab}$, respectivement à l'ensemble des *reinstatement labellings*, et des extensions complètes, alors ces deux fonctions sont bijectives et sont réciproques l'une de l'autre.

Théorème 9.
Soient les ensembles
- $Re - Lab = \{\mathcal{L} | \mathcal{L}$ *est un reinstatement labelling de* $AF\}$
- $Ext = \{\varepsilon | \varepsilon$ *est une extension complète de* $AF\}$

Soient les fonctions

$$\begin{array}{rrcl} \mathrm{Lab2Ext}^r_{(\mathcal{A},\mathcal{R})}: & Re - Lab & \longrightarrow & Ext \\ & \mathcal{L} & \longmapsto & \mathrm{Lab2Ext}_{(\mathcal{A},\mathcal{R})}(\mathcal{L}) \end{array}$$

et

$$\begin{array}{rrcl} \mathrm{Ext2Lab}^r_{(\mathcal{A},\mathcal{R})}: & Ext & \longrightarrow & Re - Lab \\ & \mathcal{L} & \longmapsto & \mathrm{Ext2Lab}_{(\mathcal{A},\mathcal{R})}(\mathcal{L}) \end{array}$$

Les fonctions $\mathrm{Lab2Ext}^r$ *et* $\mathrm{Ext2Lab}^r$ *sont bijectives et sont réciproques l'une de l'autre.*

De ce fait, il y a une forte connexion entre les extensions complètes et les *reinstatement labelling*.

Relation avec les extensions stables

Les deux théorèmes suivants soulignent la correspondance entre les extensions stables et les *labellings* n'ayant aucun argument $undec$, c'est-à-dire les *labellings* qui décident pour chaque argument s'il sera accepté ou refusé.

Théorème 10.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $\mathcal{L}$ *un reinstatement labelling tel que* $undec(\mathcal{L}) = \emptyset$. *Alors* $\mathrm{Lab2Ext}(\mathcal{L})$ *est une extension stable.*

Théorème 11.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $Args$ *une extension stable. Alors* $\mathrm{Ext2Lab}(Args)$ *est un reinstatement labelling avec* $undec(\mathcal{L}) = \emptyset$.

Relation avec les extensions préférées

Les extensions préférées peuvent être associées à deux types de *labellings*. Tout d'abord, les deux théorèmes ci-dessous montrent la correspondance avec les *labellings* qui maximisent $in(\mathcal{L})$.

Théorème 12.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $\mathcal{L}$ *un reinstatement labelling tel que* $in(\mathcal{L})$ *est maximal. Alors* $\mathrm{Lab2Ext}(\mathcal{L})$ *est une extension préférée.*

Théorème 13.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $Args$ *une extension préférée. Alors* $\mathrm{Ext2Lab}(Args)$ *est un reinstatement labelling avec* $in(\mathcal{L})$ *maximal.*

La même correspondance existe avec les *labellings* tels que $out(\mathcal{L})$ est maximal.

Théorème 14.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $\mathcal{L}$ *un reinstatement labelling tel que* $out(\mathcal{L})$ *est maximal. Alors* $\mathrm{Lab2Ext}(\mathcal{L})$ *est une extension préférée.*

Théorème 15.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $Args$ *une extension préférée. Alors* $\mathrm{Ext2Lab}(Args)$ *est un reinstatement labelling avec* $out(\mathcal{L})$ *maximal.*

Relation avec l'extension de base

L'extension de base peut quant à elle être associée à trois AF-*labellings* différents.
Tout d'abord, le *reinstatement labelling* qui maximise $undec$ peut être associé à l'extension de base.

Théorème 16.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $\mathcal{L}$ *un reinstatement labelling tel que* $undec(\mathcal{L})$ *est maximal. Alors* $\mathrm{Lab2Ext}(\mathcal{L})$ *est l'extension de base.*

Théorème 17.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $Args$ *l'extension de base. Alors* $\mathrm{Ext2Lab}(Args)$ *est un reinstatement labelling avec* $undec(\mathcal{L})$ *maximal.*

Ensuite, on peut associer l'extension de base au *reinstatement labelling* qui minimise in.

Théorème 18.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $\mathcal{L}$ *un reinstatement labelling tel que* $in(\mathcal{L})$ *est minimal. Alors* $\mathrm{Lab2Ext}(\mathcal{L})$ *est l'extension de base.*

Théorème 19.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $Args$ *l'extension de base. Alors* $\mathrm{Ext2Lab}(Args)$ *est un reinstatement labelling avec* $in(\mathcal{L})$ *minimal.*

Enfin, il existe aussi une correspondance entre l'extension de base et le fait que le *reinstatement labelling* minimise out.

Théorème 20.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $\mathcal{L}$ *un reinstatement labelling tel que* $out(\mathcal{L})$ *est minimal. Alors* $\mathrm{Lab2Ext}(\mathcal{L})$ *est l'extension de base.*

Théorème 21.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $Args$ *l'extension de base. Alors* $\mathrm{Ext2Lab}(Args)$ *est un reinstatement labelling avec* $out(\mathcal{L})$ *minimal.*

Un cas à part, les extensions semi-stables

Un dernier cas présenté est celui des *labellings* qui minimisent $undec$. Ce type de *labelling* peut être associé à des extensions préférées, mais cette fois ci la correspondance ne fonctionne que dans un sens.

Théorème 22.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $\mathcal{L}$ *un reinstatement labelling tel que* $undec(\mathcal{L})$ *est minimal. Alors* $\mathrm{Lab2Ext}(\mathcal{L})$ *est une extension préférée.*

La réciproque n'est pas vérifiée. Si $Args$ est une extension préférée, il n'est pas nécessairement vrai que $\mathrm{Ext2Lab}(Args)$ minimise $undec$.

Exemple 8.
Soit AF le système d'argumentation représenté à la figure 1.5.
AF a deux extensions préférées : $\varepsilon_1 = \{b, d\}$ et $\varepsilon_2 = \{a\}$. Comme ε_1 est aussi une extension stable, $\mathrm{Ext2Lab}(\varepsilon_1)$ est un *labelling*, $\mathcal{L}$, tel que $undec(\mathcal{L}) = \emptyset$, et donc ici $undec$ est bien minimal.
Cependant, $\mathrm{Ext2Lab}(\varepsilon_2) = \mathcal{L}'$ est tel que $undec(\mathcal{L}') = \{c, d, e\}$. Donc, bien que ε_2 soit une extension préférée, $undec$ n'est pas minimal pour le *reinstatement labelling* correspondant.

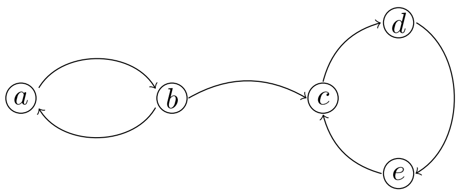

FIGURE 1.5 – Une extension préférée n'implique pas nécessairement un $undec$ minimal

Caminada introduit alors une sémantique d'extension plus fine, qui correspond aux *labellings* qui minimisent $undec$. Ces extensions sont aussi des extensions complètes particulières, il s'agit de celles qui maximisent $Args \cup Args^+$ (où $Args$ est le nom de l'extension, et $Args^+$ est l'ensemble des arguments attaqués par un argument de $Args$).

Définition 20 (Extension semi-stable)**.**
Soit $\langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation et $Args \subseteq \mathcal{A}$. $Args$ est appelé extension semi-stable ssi $Args$ est une extension complète telle que $Args \cup Args^+$ est maximal.

A partir de cette nouvelle définition d'extension, on a cette fois des théorèmes qui indiquent une vraie correspondance avec les *labellings* qui minimisent $undec$.

Théorème 23.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation et $\mathcal{L}$ un reinstatement labelling tel que $undec(\mathcal{L})$ est minimal. Alors $\mathrm{Lab2Ext}(\mathcal{L})$ est une extension semi-stable.

Théorème 24.
Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation et $Args$ une extension semi-stable. Alors $\mathrm{Ext2Lab}(Args)$ est un reinstatement labelling avec $undec(\mathcal{L})$ minimal.

1.3 Inférence à partir d'un système d'argumentation

A partir d'un système d'argumentation et d'une sémantique, un agent peut tirer certaines conclusions (retenir des informations, prendre une décision). Le calcul des extensions (ou des *reinstatement labellings* correspondants) donne en général plusieurs résultats (parmi les sémantiques présentées, seule la sémantique de base garantit l'existence d'une unique extension). Il faut donc que l'agent choisisse de quelle façon il va retenir certains arguments. Plusieurs politiques d'inférences existent, en particulier l'inférence sceptique et l'inférence crédule.

On introduit d'abord des notations pour l'ensemble des extensions et l'ensemble des *labellings* d'un système d'argumentation.

Définition 21 (Ensembles des extensions, ensemble des *labellings*)**.**
Soit AF un système d'argumentation, soit $\mathcal{S}$ une sémantique. On note $Ext(AF, \mathcal{S})$ l'ensemble des extensions de AF suivant la sémantique $\mathcal{S}$. S'il n'y a pas d'ambiguïté sur la sémantique choisie, on note $Ext(AF)$.
De façon équivalente, on note $Labs(AF, \mathcal{S})$ l'ensemble des *reinstatement labellings* construits pour la sémantique $\mathcal{S}$, ou $Labs(AF)$.

L'inférence sceptique (ou universelle) revient pour un agent à faire confiance à une information si toutes les sources d'informations (dans notre cas, toutes les extensions ou tous les *reinstatement labellings* du système d'argumentation) sont d'accord sur cette information.

Définition 22 (Inférence sceptique)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, soit $\alpha \in \mathcal{A}$ un argument. On dit que AF infère sceptiquement α pour la sémantique $\mathcal{S}$, noté $AF \mid\!\sim_{\forall}^{\mathcal{S}} \alpha$, si et seulement si

$$\forall \varepsilon \in Ext(AF, \mathcal{S}), \alpha \in \varepsilon$$

ou, de manière équivalente,

$$\forall \mathcal{L} \in Labs(AF, \mathcal{S}), \mathcal{L}(\alpha) = in$$

S'il n'y a pas d'ambiguïté possible sur la sémantique, on note $AF \mid\!\sim_{\forall} \alpha$.

L'inférence crédule (ou existentielle) impose une condition plus faible que l'inférence sceptique pour accepter un argument. Cette fois, l'agent accepte une information dès qu'une source d'information (ici, au moins une extension ou au moins un *reinstatement labelling*) donne cette information.

Définition 23 (Inférence crédule)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, soit $\alpha \in \mathcal{A}$ un argument. On dit que AF infère crédulement α pour la sémantique $\mathcal{S}$, noté $AF \mid\!\sim_{\exists}^{\mathcal{S}} \alpha$, si et seulement si

$$\exists \varepsilon \in Ext(AF, \mathcal{S}) \text{ tel que } \alpha \in \varepsilon$$

ou, de manière équivalente,

$$\exists \mathcal{L} \in Labs(AF, \mathcal{S}) \text{ tel que } \mathcal{L}(\alpha) = in$$

S'il n'y a pas d'ambiguïté possible sur la sémantique, on note $AF \mid\!\sim_{\exists} \alpha$.

1.4 Equivalence entre systèmes d'argumentation

Si la notion d'équivalence en logique peut se définir de façon assez simple, il n'en est pas de même dans le cadre de l'argumentation. On va présenter des travaux qui ont été proposés pour caractériser cette notion.

1.4.1 Plusieurs critères d'équivalence

Différentes façons de considérer deux systèmes équivalents

Dans [Ves11], la partie sur l'équivalence entre systèmes d'argumentations donne essentiellement des résultats concernant les systèmes d'argumentation instanciés avec une logique tarskienne. Cependant, on peut retenir plusieurs critères selon lesquels deux systèmes d'argumentation abstraits peuvent être considérés comme équivalents.

Une forme d'équivalence basique est l'équivalence de la structure du graphe associé au système d'argumentation. Cette notion présente un inconvénient majeur : elle peut être vue comme l'existence d'un renommage sur les arguments, et ne porte pas vraiment de sens, on préfère donc utiliser des critères d'équivalence qui portent sur le sens attaché à un système d'argumentation, c'est-à-dire les extensions et les arguments acceptés.

On établit d'abord deux notations. Etant donné $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation,
- $Sc(AF) = \{a \in A | AF \mid\!\sim_{\forall} a\}$
- $Cr(AF) = \{a \in A | AF \mid\!\sim_{\exists} a\}$

On définit à partir de cela plusieurs notions d'équivalence.

Définition 24 (Critères d'équivalence entre systèmes d'argumentation ([Ves11])**.**
Soient AF et AF' deux systèmes d'argumentation. On a :
- EQ_1 : AF et AF' sont équivalents si $Ext(AF) = Ext(AF')$
- EQ_2 : AF et AF' sont équivalents si $Sc(AF) = Sc(AF')$
- EQ_3 : AF et AF' sont équivalents si $Cr(AF) = Cr(AF')$

On notera dans la suite $AF \equiv_{EQ_i} AF'$ si et seulement si AF et AF' sont équivalents selon le critère EQ_i.

Des liens entre les critères

Le fait que deux systèmes soient équivalents selon un critère peut impliquer qu'ils soient également équivalents selon d'autres critères. [Ves11] présente un certain nombre d'implications de ce genre, dans le cadre de l'argumentation abstraite, on en retient deux.

Proposition 25.
On note « $EQ_i \Rightarrow EQ_j$ » si et seulement si « $\forall AF, AF'$, $AF \equiv_{EQ_i} AF' \Rightarrow AF \equiv_{EQ_j} AF'$ ».
Suivant cette définition, on a les liens suivants entre critères :
- $EQ_1 \Rightarrow EQ_2$
- $EQ_1 \Rightarrow EQ_3$

1.4.2 L'équivalence forte

L'équivalence forte entre systèmes d'argumentation, telle que définie dans [OW11], est le fait que ces deux systèmes restent équivalents quand ils sont « augmentés » par un même troisième système d'argumentation.

Définition 25 (Equivalence forte entre deux systèmes)**.**
Soient deux systèmes d'argumentation AF et AF'. Ils sont dits fortement équivalents, selon une sémantique $\mathcal{S}$, si et seulement si $Ext(AF \cup AF'') = Ext(AF' \cup AF'')$.
On note alors $AF \equiv_S^{\mathcal{S}} AF'$, ou $AF \equiv_S AF'$ s'il n'y a pas d'ambiguïté possible sur la sémantique.

Par définition, $AF \equiv_S AF'$ implique $Ext(AF) = Ext(AF')$, c'est-à-dire AF et AF' sont EQ_1-équivalents, et donc également EQ_2 et EQ_3-équivalents.

L'équivalence forte en terme de conséquence

On peut définir l'équivalence forte par rapport aux critères d'équivalence donnés précédemment.

Définition 26.
Les systèmes d'argumentation AF et AF' sont fortement EQ_i-équivalents, avec $i \in \{1, 2, 3\}$, selon une sémantique $\mathcal{S}$, si et seulement si $AF \cup AF''$ et $AF' \cup AF''$ sont EQ_i-équivalents.
On note alors $AF \equiv_{S;EQ_i} AF'$.

La EQ_1-équivalence forte revient à l'équivalence forte de base.
On remarque que pour un ensemble de sémantiques habituelles, les concepts d'équivalence forte sont équivalents.

Théorème 26 ([OW11])**.**
Etant donnés AF et AF' deux systèmes d'argumentation, quelle que soit la sémantique parmi $\{$*stable, preférée, complète, de base, idéale* [1], *semi-stable*$\}$,

$$AF \equiv_S AF' \Leftrightarrow AF \equiv_{S;EQ_1} AF' \Leftrightarrow AF \equiv_{S;EQ_2} AF'$$

1.4.3 L'équivalence locale

Après avoir considéré un contexte arbitraire pour l'équivalence, Oikarinen et Woltran présentent une notion plus faible, en considérant seulement les systèmes d'argumentation AF'' qui n'ajoutent pas de nouvel argument par rapport à ceux de AF et AF'.

Définition 27 (Equivalence locale)**.**
Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ et $AF' = \langle \mathcal{A}', \mathcal{R}' \rangle$. AF et AF' sont dits localement équivalents selon une sémantique $\mathcal{S}$ si et seulement si $\forall AF'' = \langle \mathcal{A}'', \mathcal{R}'' \rangle$, avec $\mathcal{A}'' \subseteq \mathcal{A} \cup \mathcal{A}'$, $Ext(AF \cup AF'') = Ext(AF' \cup AF'')$.
On note $AF \equiv^{\mathcal{S}}_{l} AF'$.

Pour certaines sémantiques, l'équivalence forte et l'équivalence locale coïncident.

Théorème 27 ([OW11])**.**
Quels que soient AF et AF' deux systèmes d'argumentation, pour $\mathcal{S} \in \{$admissible, preférée, idéale, semi-stable$\}$, $AF \equiv^{\mathcal{S}}_{l} AF'$ si et seulement si $AF \equiv^{\mathcal{S}}_{S}$.

Pour le cas où les systèmes d'argumentation sont sans boucle, le concept d'équivalence locale revient à l'équivalence de syntaxe pour toutes les sémantiques sauf la sémantique de base.

Théorème 28 ([OW11])**.**
Quels que soient les systèmes d'argumentation sans boucle AF et AF', pour $\mathcal{S} \in \{$stable, admissible, préférée, complète, idéale, semi-stable$\}$, $AF = AF'$ si et seulement si $AF \equiv^{\mathcal{S}}_{l} AF'$.

1. Pour la sémantique idéale, voir [DMT07]

Chapitre 2

La dynamique des croyances

On distingue la notion de croyance de celle de connaissance. Une connaissance d'un agent à propos du monde est une information certaine. La notion de croyance, en revanche, est plus permissive : il s'agit d'une information crue par l'agent, *a priori*, mais qu'il peut faire évoluer sans que le monde ne change, par exemple parce que l'agent a obtenu des précisions sur une information qui était incertaine. Ces évolutions de la base de croyances peuvent être de plusieurs types, on a donc défini différents opérateurs pour chacune de ces évolutions.

Dans ce chapitre, nous allons présenter le cadre AGM, du nom de Carlos Alchourrón, Peter Gärdenfors et David Makinson, qui ont défini différentes propriétés que doivent avoir des opérateurs de changement de croyances afin que leur comportement soit « raisonnable ». Dans une seconde partie, nous introduirons la notion de révision en logique propositionnelle, qui est un cas particulier du cadre AGM, particulièrement intéressant en raison de l'usage courant de la logique propositionnelle.

Sommaire

2.1 Présentation du cadre AGM 20

2.1.1 Préliminaires 20

2.1.2 Postulats de rationalité 21

2.1.3 Théorèmes de représentation 24

2.2 Révision dans un cadre particulier : la logique propositionelle 27

2.2.1 Postulats pour la révision en logique propositionnelle 28

2.2.2 Théorème de représentation : les assignements fidèles 28

2.1 Présentation du cadre AGM

2.1.1 Préliminaires

Notations

On définit un langage logique L dans lequel sont exprimées les croyances :
- la tautologique $\top$ (formule toujours vraie) et la contradiction $\perp$ (formule toujours fausse) sont des formules du langage ;
- soit V l'ensemble des variables, alors chaque élément $v \in V$ est une formule du langage ;
- soit C l'ensemble des connecteurs disponibles, alors pour chaque élément $c \in C$ d'arité n, si $f_1, \ldots, f_n$ sont des formules alors $c(f_1, \ldots, f_n)$ est une formule du langage.

Dans le cadre défini dans [AGM85], on raisonne sur des théories, c'est-à-dire des bases de croyances closes pour la déduction logique : $Cn(K) = K$, avec Cn une opération de conséquence ([Tar30]). Cn doit respecter trois propriétés, $\forall X, Y$ des ensembles de formules :
- $X \subseteq Cn(X)$
- $Cn(X) = Cn(Cn(X))$
- Si $X \subseteq Y$, alors $Cn(X) \subseteq Cn(Y)$

Dans la suite, on notera $K_\perp$ la base triviale, c'est-à-dire l'ensemble des formules du langage : $K_\perp = Cn(\{\perp\}) = L$.

Changement de croyances

Pour un agent dont les croyances sont représentées par la base K, une formule α peut avoir trois statuts épistémiques différents :

1. $\alpha \in K$: l'agent accepte cette information, il croit qu'elle est vraie
2. $\neg\alpha \in K$: l'agent refuse cette information, il croit qu'elle est fausse
3. $\alpha \notin K$ et $\neg\alpha \notin K$: l'agent ne croit rien sur cette information, elle est indéterminée

On peut passer d'un état à un autre via plusieurs opérations, l'expansion, la contraction et la révision, qui seront présentées en détail dans la partie suivante.

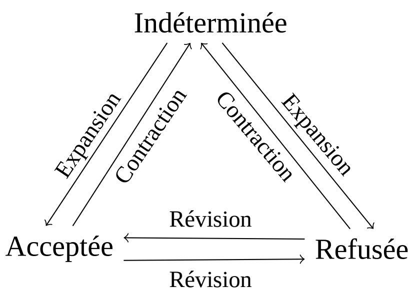

FIGURE 2.1 – Transitions entre statuts épistémiques

De façon générale, on attend qu'une opération de changement de croyances respecte trois propriétés :
- Primauté de la nouvelle information : le changement doit réussir, c'est-à-dire que l'information dont on veut changer le statut doit effectivement avoir le statut voulu
- Cohérence : on veut que la nouvelle base de croyances soit cohérente

– Changement minimal : les croyances de l'agent doivent être modifiées le moins possible, on ne veut pas en enlever trop, ou en ajouter qui ne sont pas nécessaires

2.1.2 Postulats de rationalité

Pour chacune des opérations présentées précédemment, Alchourrón, Gärdenfors et Makinson ont proposé un ensemble de postulats de rationalité : on peut définir un opérateur qui ne respecte pas ces propriétés, mais dans ce cas là cet opérateur n'a pas un comportement raisonnable.

Pour expliquer les notions à venir, nous utilisons cet exemple de base de croyances :

Exemple 9 (Base de croyance d'un individu)**.**
Un individu sort de chez lui. Il ne sait pas s'il pleut, mais il pense que s'il pleut il doit prendre son parapluie, dans le cas contraire le parapluie peut rester à sa place. On peut représenter ses croyances de la façon suivante :

$$C = \{pluie \Rightarrow parapluie, \neg pluie \Rightarrow \neg parapluie\}$$

et donc, pour respecter le formalisme AGM,

$$K = Cn(C)$$

Les propositions $pluie$ et $parapluie$ signifient respectivement « l'individu pense qu'il pleut » et « l'individu prend son parapluie ».

Expansion

Lorsque la base de croyances d'un agent ne contient aucune information à propos d'une formule α (ni α ni $\neg\alpha$ n'appartiennent à la base), l'agent peut ajouter une des alternatives s'il apprend quelque chose.

Exemple 10 (Expansion des croyances de l'individu)**.**
Une fois sorti, l'individu constate qu'il ne pleut pas. S'il effectue une expansion, le résultat est la base suivante :

$$C_{exp} = \{pluie \Rightarrow parapluie, \neg pluie \Rightarrow \neg parapluie, \neg pluie\}$$

Dans ce cas, l'individu peut partir sans son parapluie. Si la base de croyances de départ avait été différente :

$$C' = \{pluie, pluie \Rightarrow parapluie, \neg pluie \Rightarrow \neg parapluie\}$$

l'expansion aurait été incohérente car elle aurait contenu les propositions $pluie$ et $\neg pluie$.

Un opérateur d'expansion $+$ est une fonction de $K \times L$ vers K, c'est-à-dire qu'elle calcule une nouvelle base de croyances à partir d'une base de croyances et d'une croyance α à ajouter, telle que :

– **(K+1)** $K + \alpha$ est une théorie
– **(K+2)** $\alpha \in K + \alpha$
– **(K+3)** $K \subseteq K + \alpha$
– **(K+4)** Si $\alpha \in K$, alors $K + \alpha = K$
– **(K+5)** Si $K' \subseteq K$, alors $K' + \alpha \subseteq K + \alpha$
– **(K+6)** $K + \alpha$ est la plus petite base satisfaisant **(K+1) - (K+5)**

Pour chaque postulat, on peut donner une explication intuitive.
Le premier assure que le résultat de l'opération est bien une théorie, c'est-à-dire que les croyances sont représentées de la même façon après l'expansion qu'avant. **(K+2)** assure que la nouvelle information a bien été ajoutée à la base de croyances (c'est le but principal de cette opération). **(K+3)** impose de ne pas retirer d'anciennes informations de la base, mais seulement d'en ajouter, cela justifie le nom d'expansion. **(K+4)** exprime le fait qu'il n'est pas nécessaire d'apporter de changements à la base si la nouvelle information était déjà dans la base. Le postulat **(K+5)** dit que l'opération d'expansion est monotone. Enfin, le dernier postulat exprime la minimalité du changement : aucune croyance qui n'est pas justifiée par l'ajout d'α ne doit être ajoutée.
De ces postulats on arrive à la conclusion qu'il n'existe qu'un seul opérateur d'expansion raisonnable.

Théorème 29 ([Gä88])**.**
*L'opérateur d'expansion + satisfait les postulats **(K+1) - (K+6)** si et seulement si* $K + \alpha = Cn(K \cup \alpha)$.

Contraction

Dans certaines situations, il peut être utile de retirer une information de la base de croyances, mais sans forcément la remplacer par sa négation : on veut passer de « je crois quelque chose sur ce sujet » à « j'ignore ce qu'il en est ».

Exemple 11 (Contraction des croyances de l'individu)**.**
Supposons que l'individu, avant de partir, entend un bulletin météo qui annonce de la pluie. Ses croyances sont alors

$$C = \{pluie, pluie \Rightarrow parapluie, \neg pluie \Rightarrow \neg parapluie\}$$

S'il se rend compte que ce bulletin météo correspond à des prévisions pour le lendemain, et pas pour le jour même, il faut alors retirer la croyance $pluie$: il ne peut pas raisonnablement continuer à penser qu'il pleut, sans pour autant penser qu'il ne pleut pas. Le résultat de la contraction est donc

$$C_{cont} = \{pluie \Rightarrow parapluie, \neg pluie \Rightarrow \neg parapluie\}$$

Un opérateur de contraction $\div$ est une fonction de $K \times L$ vers K, c'est-à-dire qu'elle calcule une nouvelle base de croyances à partir d'une base de croyances et d'une croyance α à retirer. Elle vérifie les propriétés suivantes :

- **(K÷1)** $K \div \alpha$ est une théorie
- **(K÷2)** $K \div \alpha \subseteq K$
- **(K÷3)** Si $\alpha \notin K$, alors $K \div \alpha = K$
- **(K÷4)** Si $\nvdash \alpha$, alors $\alpha \notin K \div \alpha$
- **(K÷5)** Si $\alpha \in K$, alors $K \subseteq (K \div \alpha) + \alpha$
- **(K÷6)** Si $\vdash \alpha \leftrightarrow \beta$, alors $K \div \alpha = K \div \beta$
- **(K÷7)** $(K \div \alpha) \cap (K \div \beta) \subseteq K \div (\alpha \wedge \beta)$
- **(K÷8)** Si $\alpha \notin K \div (\alpha \wedge \beta)$, alors $K \div (\alpha \wedge \beta) \subseteq K \div \alpha$

De la même façon que pour l'expansion, on peut justifier chaque postulat de façon intuitive.
(K÷1) se justifie de la même façon que **(K+1)**. **(K÷2)** assure qu'aucune nouvelle information n'est ajoutée par la contraction. **(K÷3)** impose de n'effectuer aucun changement à la base de croyance si l'information à retirer était déjà absente de la base. On peut le voir comme une contrepartie à **(K+4)**. **(K÷4)** garantit le succès de l'opération : si la croyance α n'est pas une formule valide, alors la contraction réussit et α n'apparaît pas dans la nouvelle base. Le postulat **(K÷5)** assure que la contraction par α suivie de l'expansion par α redonne la théorie de K, l'inclusion inverse étant une conséquence des quatre premiers

postulats, on voit donc bien la relation entre ces deux opérations. **(K÷6)** exprime l'indépendance de la contraction vis-à-vis de la syntaxe. On appelle ces 6 postulats les postulats de base pour la contraction.
Les deux derniers postulats sont appelés postulats supplémentaires. Certains travaux étudient des méthodes de contraction qui ne satisfont pas ces deux là, mais seulement les postulats de base. **(K÷7)** dit que si une croyance appartient à la base contractée par α et à la base contractée par β, alors elle appartient à la base contractée par la conjonction $\alpha \wedge \beta$. Enfin, le dernier postulat exprime la minimalité du changement pour la conjonction : si on doit retirer $\alpha \wedge \beta$, concrètement cela peut revenir à retirer soit α, soit β. Si α est effectivement retiré, alors on a retiré au moins autant que si on avait juste voulu retirer α.

Révision

Le dernier opérateur de changement de croyances que nous présenterons est la révision. Cette dernière consiste à l'ajout d'une croyance α à la base K, éventuellement telle que $\alpha \wedge K$ n'est pas cohérent. Il faut dans ce cas maintenir la cohérence de la base quand on ajoute la nouvelle croyance, chose qui n'est pas garantie par l'expansion.

Exemple 12 (Révision des croyances de l'individu)**.**
On revient au cas où les croyances de l'individu sont les suivantes :

$$C = \{pluie, pluie \Rightarrow parapluie, \neg pluie \Rightarrow \neg parapluie\}$$

Par exemple à cause d'un bulletin météo. Si en sortant de chez lui, l'agent constate qu'il ne pleut pas, il est raisonnable de remplacer ses croyances celles-ci :

$$C_{rev} = \{\neg pluie, pluie \Rightarrow parapluie, \neg pluie \Rightarrow \neg parapluie\}$$

Ainsi, il intègre la nouvelle information tout en préservant la cohérence logique de la base.

Un opérateur de révision $*$ est une fonction de $K \times L$ vers K, c'est-à-dire qu'elle calcule une nouvelle base de croyances à partir d'une base de croyances à réviser avec une nouvelle croyance α. Elle vérifie les propriétés suivantes :

- **(K$*$1)** $K * \alpha$ est une théorie
- **(K$*$2)** $\alpha \in K * \alpha$
- **(K$*$3)** $K * \alpha \subseteq K + \alpha$
- **(K$*$4)** Si $\neg\alpha \notin K$, alors $K + \alpha \subseteq K * \alpha$
- **(K$*$5)** $K * \alpha = K_\perp$ si et seulement si $\vdash \neg\alpha$
- **(K$*$6)** Si $\vdash \alpha \leftrightarrow \beta$, alors $K * \alpha = K * \beta$
- **(K$*$7)** $K * (\alpha \wedge \beta) \subseteq (K * \alpha) + \beta$
- **(K$*$8)** Si $\neg\beta \notin K * \alpha$, alors $(K * \alpha) + \beta \subseteq K * (\alpha \wedge \beta)$

Comme pour les opérateurs précédents, **(K$*$1)** permet d'assurer que le résultat de la révision est bien une théorie. Le postulat suivant indique que la nouvelle information doit être vraie dans la base de croyances révisée. **(K$*$3)** veut dire que les informations contenues dans la base révisée sont soit des conséquences des croyances d'origine, soit des conséquences de la nouvelle croyance. Ce même postulat, avec le postulat **(K$*$4)**, dit que dans le cas où la nouvelle croyance ne contredit pas la base de croyances d'origine, alors la révision est équivalente à l'expansion. **(K$*$5)** indique qu'on ne peut obtenir une base incohérente qu'en révisant par une information incohérente. **(K$*$6)** garantit l'indépendance de syntaxe de l'opération de révision.
Comme pour la contraction, on parle de postulats de base pour ces six postulats. Les deux postulats supplémentaires garantissent que l'opérateur de révision respecte le principe de minimalité du changement.

(K$*$7) indique que la base révisée par deux informations (conjonction) est incluse dans la base révisée par la première information puis expansée par la seconde. Le dernier postulat indique que l'inclusion inverse doit être vraie si la seconde croyance n'est pas contradictoire avec la base révisée par la première croyance.

Equivalence contraction/révision

Il existe une forte « parenté » entre la contraction et la révision. En effet, il est possible de définir un opérateur de révision à partir d'un opérateur de contraction, ou vice-versa.

Proposition 30 (Identité de Levi)**.**
$K * \alpha = (K \div \neg\alpha) + \alpha$

Cette identité peut se déduire du schéma 2.1, qui permet de supposer qu'une révision se décompose en une contraction (pour passer d'accepté à indéterminé par exemple) et une expansion (pour passer d'indéterminé à refusé).

Théorème 31 ([Gä88])**.**
*Si l'opérateur de contraction $\div$ satisfait **(K$\div$1) - (K$\div$4)** et **(K$\div$6)**, et si l'opérateur d'expansion $+$ satisfait **(K$+$1) - (K$+$6)**, alors l'opérateur $*$ défini par l'identité de Levi satisfait **(K$*$1) - (K$*$6)**.*
*De plus si $\div$ satisfait **(K$\div$7)** (resp. **(K$\div$8)**) alors $*$ satisfait **(K$*$7)** (resp. **(K$*$8)**).*

On remarque que le cinquième postulat de base n'est pas nécessaire pour ce résultat.
De façon réciproque, on peut définir un opérateur de contraction à partir d'un opérateur de révision.

Proposition 32 (Identité de Harper)**.**
$K \div \alpha = K \cap (K * \neg\alpha)$

Si α est acceptée dans la base de croyances de départ, la contraction de K par α est l'ensemble des formules qui sont vraies quel que soit l'état d'acceptation de α, c'est-à-dire l'intersection de la base de départ, dans laquelle α est acceptée et de la base révisée par $\neg\alpha$, dans laquelle α est refusée.

Théorème 33 ([Gä88])**.**
*Si l'opérateur de révision $*$ satisfait **(K$*$1) - (K$*$6)**, alors l'opérateur de contraction $\div$ défini par l'identité de Harper satisfait **(K$\div$1) - (K$\div$6)**.*
*De plus si $*$ satisfait **(K$*$7)** (resp. **(K$*$8)**) alors $\div$ satisfait **(K$\div$7)** (resp. **(K$\div$8)**).*

Ces deux identités montrent qu'il y a un lien très fort entre les opérateurs de révision et les opérateurs de contraction, on peut donc par exemple étudier un seul de ces deux types d'opérateurs en détail, puis définir l'autre grâce à l'identité correspondante. De plus, les deux théorèmes de Gärdenfors confirment le bon sens des postulats, puisqu'on retrouve le même lien entre les postulats correspondants aux opérateurs qu'entre les opérateurs eux-mêmes.

2.1.3 Théorèmes de représentation

Après avoir défini les bonnes propriétés des opérateurs de changement de croyances, il est intéressant de définir concrètement des opérateurs. Les théorèmes de représentation permettent de donner une expression possible d'un opérateur qui satisfait les postulats AGM.
Il existe un certain nombre de théorèmes de représentation, nous n'en présenterons que trois dans cette partie : les intersections partielles, les enracinements épistémiques et les systèmes de sphères.

Contraction par intersection partielle

Le principe de la contraction par intersection est de conserver chaque sous-ensemble maximal (pour l'inclusion) de la base qui ne permet pas de déduire la croyance à retirer. A partir de tous ces sous-ensembles, on construit la nouvelle base de croyances par inférence sceptique, c'est-à-dire qu'on conserve les informations qui se trouvent dans chaque sous-ensemble.

Définition 28 (Sous-théories maximales n'impliquant pas une croyance)**.**
Soient K une théorie et α une proposition. L'ensemble des sous-théories maximales de K n'impliquant pas α, noté $K \perp \alpha$, est l'ensemble de tous les K' tels que :
- $K' \subseteq K$: K' est une sous-théorie
- $K' \nvdash \alpha$: K' n'implique pas α
- $\forall K''$ tel que $K' \subset K'' \subseteq K, K'' \vdash \alpha$: K' est maximale parmi les sous-théories qui n'impliquent pas α

Définition 29 (Contraction par intersection totale)**.**
Une fonction de contraction par intersection totale est définie comme

$$K \div_f \alpha = \begin{cases} \cap(K \perp \alpha) & \text{si } K \perp \alpha \text{ n'est pas vide} \\ K & \text{sinon} \end{cases}$$

La définition de la contraction par intersection totale pose un problème : le résultat est l'ensemble des formules de K qui sont conséquences de $\neg\alpha$.

Théorème 34 ([AM82])**.**
Un opérateur de révision $*$ défini à partir d'une fonction de contraction par intersection totale et de l'identité de Levi revient à $K * \alpha = Cn(\alpha)$ pour chaque α tel que $\neg\alpha \in K$.

Concrètement, ce théorème veut dire qu'un opérateur de révision construit à partir de la contraction par intersection totale va oublier les anciennes croyances, et ne conserver que la nouvelle croyance et ses conséquences. Ce comportement n'est pas souhaitable pour un agent rationnel. Le problème de l'intersection totale est qu'on retire trop d'informations en gardant toutes les sous-théories maximales n'impliquant pas α. Il faut n'en conserver que certaines, « meilleures » que les autres.

Définition 30 (Fonction de sélection)**.**
On appelle fonction de sélection γ une fonction qui associe à chaque proposition α et à chaque théorie K l'ensemble $\gamma(K \perp \alpha)$, qui est un sous-ensemble non vide de $K \perp \alpha$ si celui ci n'est pas vide, et $\{K\}$ sinon.

Définition 31 (Contraction par intersection partielle)**.**
Une fonction de contraction par intersection partielle est définie comme

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

Globalement, la contraction par intersection partielle fonctionne de la même façon que la contraction par intersection totale, mis à part que cette fois ci seules les sous-théories « choisies » par γ interviennent. D'ailleurs, l'intersection totale est un cas particulier d'intersection partielle, celui où γ conserve tous les $K' \in K \perp \alpha$.

On peut maintenant énoncer le théorème de représentation d'Alchourrón, Gärdenfors et Makinson concernant les fonctions de contraction par intersection partielle. Ce théorème indique qu'un tel opérateur de contraction satisfait les propriétés attendues pour la contraction, et réciproquement qu'un opérateur qui satisfait ces propriétés peut être défini via l'intersection partielle.

Théorème 35 ([AGM85])**.**
*Un opérateur $\div$ est une fonction de contraction par intersection partielle si et seulement si il satisfait **(K÷1) - (K÷6)**.*

Il est possible d'ajouter des contraintes sur γ pour satisfaire aussi les postulats **(K÷7)** et **(K÷8)**, on parle alors de contraction par intersection partielle relationnelle transitive.

Définition 32 (Fonction de sélection relationnelle)**.**
Une fonction de sélection γ est relationnelle si et seulement si quelle que soit la base de croyance K, on peut définir une relation $\leqslant$ sur $K \times K$ telle que

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

Si de plus la relation $\leqslant$ est transitive, alors γ et la fonction de contraction par intersection partielle définie à partir de γ sont dites relationnelles transitives.

On peut présenter un théorème de représentation pour ces fonctions de contraction.

Théorème 36 ([AGM85])**.**
*Un opérateur $\div$ est une fonction de contraction par intersection partielle si et seulement si il satisfait **(K÷1) - (K÷8)**.*

Contraction par enracinement épistémique

L'idée de Gärdenfors est d'ordonner les formules par rapport à leur « importance ». Cette importance dépend du contexte, par exemple un degré de fiabilité : un croyance est plus importante qu'une autre si on considère que la première a plus de chances d'être vraie que la deuxième.
On peut alors prendre en compte cet ordre pour définir la contraction, en choisissant d'enlever les formules les moins importantes.

Définition 33 (Contraction par enracinement épistémique)**.**
Un enracinement épistémique est une relation sur les formules, telle que $\alpha \leqslant \beta$ signifie que β est au moins aussi importante (enracinée) que α. Cet ordre doit satisfaire les propriétés suivantes :
- **(EE1)** $\leqslant$ est transitive
- **(EE2)** Si $\alpha \vdash \beta$, alors $\alpha \leqslant \beta$
- **(EE3)** $\alpha \leqslant \alpha \wedge \beta$ ou $\beta \leqslant \alpha \wedge \beta$
- **(EE4)** Si $K \neq K_\bot$, $\alpha \notin K$ ssi $\forall \beta, \alpha \leqslant \beta$
- **(EE5)** Si $\beta \leqslant \alpha \forall \beta$, alors $\vdash \alpha$

Théorème 37 ([Gä88])**.**
*Une fonction de contraction $\div$ satisfait **(K÷1) - (K÷8)** ssi il existe $\leqslant$ satisfaisant **(EE1) - (EE5)**, où $\beta \leqslant \alpha$ ssi $\beta \notin K \div \alpha \wedge \beta$, ou $\vdash \alpha \wedge \beta$.*

De façon intuitive, on peut comprendre ce théorème par le fait que si en retirant $\alpha \wedge \beta$ de la base de croyances, on ne peut plus déduire β, alors β n'était pas plus enraciné que α.

Systèmes de sphères

Une idée pour la révision de croyances est de conserver les interprétations les plus plausibles, et pour cela d'avoir une notion d'ordre sur les interprétations. Cette notion a été étudiée dans le cadre de la logique propositionnelle finie, mais on peut d'abord présenter une définition plus générale, donnée par

Grove via la notion de systèmes de sphères.

Un monde possible d'une base de croyances est un sous-ensemble du langage, maximal parmi les sous-ensembles cohérents, tel que les formules de la base de croyances sont vraies dans ce sous-ensemble. Chaque monde possible est une façon (parmi d'autres) de décrire le monde entièrement qui est cohérente avec les croyances de l'agent.

Définition 34 (Mondes possibles d'une base de croyances)**.**
On appelle monde possible un sous-ensemble maximal cohérent du langage et on note $M_{\mathcal{L}}$ l'ensemble des mondes possibles du langage $\mathcal{L}$.

– Soit une base de croyances K.

$$[K] = \begin{cases} \varnothing & \text{si} \quad K = K_{\perp} \\ \{M \in M_{\mathcal{L}} | K \subseteq M\} & \text{sinon} \end{cases}$$

– Soit un ensemble $S \subseteq M_{\mathcal{L}}$, on définit $K_S = \cap\{M | M \in S\}$

Définition 35 (Système de sphères)**.**
Un système de sphères centré sur $[K]$ est une collection S de sous-ensembles de $M_{\mathcal{L}}$ tels que

– **(S1)** Si $s, s' \in S$ alors $s \subseteq s'$ ou $s' \subseteq s$
– **(S2)** $[K] \in S$
– **(S3)** $\forall s \in S, [K] \subseteq s$
– **(S4)** $M_{\mathcal{L}} \in S$
– **(S5)** Si α est une formule et si $[\alpha]$ intersecte une sphère de S, alors il existe une sphère minimale qui intersecte $[\alpha]$ (on note $C(\alpha) = [\alpha] \cap S_{\alpha}$)

Un sphère est définie comme un ensemble de mondes possibles, et le systèmes de sphères centré sur $[K]$ est construit de la façon suivante :

– Les sphères sont imbriquées les unes dans les autres.
– L'ensemble des mondes possibles de K est la plus petite sphère.
– L'ensemble des mondes possibles, $M_{\mathcal{L}}$, est la plus grande sphère.

Théorème 38 ([Gro88])**.**
*Soit une base de croyances K. Il existe un système de sphères S centré sur $[K]$ tel que pour toute formule α, $K * \alpha = K_{C(\alpha)}$ si et seulement si $*$ est un opérateur de révision satisfaisant **(K$*$1) - (K$*$8)**.*

Graphiquement, la révision au moyen d'un système de sphères peut être représentée comme sur la figure 2.2 : les mondes qui sont cohérents avec les croyances de l'agent sont les mondes les plus plausibles ($[K]$). Ensuite, les mondes possibles sont ordonnés grâce à l'emboîtement des sphères : les mondes de la sphère S_1 sont moins plausibles que ceux de $[K]$, mais plus plausibles que ceux de la sphère S_2. Pour réviser par une nouvelle croyance α, on conserve les mondes possibles de α qui appartiennent à la sphère la plus « basse » possible, c'est-à-dire ceux qui sont les plus plausibles.

2.2 Révision dans un cadre particulier : la logique propositionelle

Les postulats de rationalité exprimés par AGM sont valables dans n'importe quelle logique disposant d'une opération de conséquence au sens de Tarski. Cependant, on peut les exprimer de façon plus simple dans le cadre particulier de la logique propositionnelle. C'est ce que nous allons présenter dans cette section, pour ensuite introduire un théorème de représentation basé sur ces nouveaux postulats.

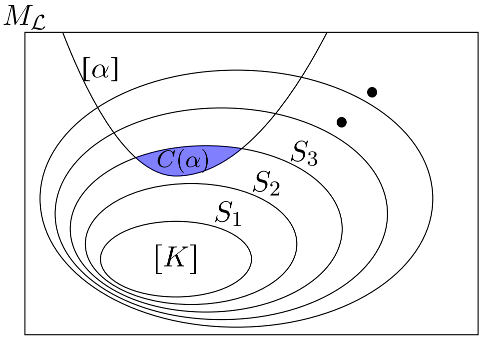

FIGURE 2.2 – Révision de K par α via un système de sphères centré sur $[K]$

2.2.1 Postulats pour la révision en logique propositionnelle

Katsuno et Mendelzon ont donné des postulats pour la révision adaptés à la logique propositionnelle finie. Dans la suite, on note φ la formule représentant les connaissances de l'agent.

- **(R1)** $\varphi \circ \alpha \vdash \alpha$
- **(R2)** Si $\varphi \wedge \alpha$ est cohérent, alors $\varphi \circ \alpha \equiv \varphi \wedge \alpha$
- **(R3)** Si α est cohérent alors $\varphi \circ \alpha$ est cohérent
- **(R4)** Si $\varphi \equiv \psi$ et $\alpha \equiv \beta$ alors $\varphi \circ \alpha \equiv \psi \circ \beta$
- **(R5)** $(\varphi \circ \alpha) \wedge \psi \vdash \varphi \circ (\alpha \wedge \psi)$
- **(R6)** Si $(\varphi \circ \alpha) \wedge \psi$ est cohérent alors $\varphi \circ (\alpha \wedge \psi) \vdash (\varphi \circ \alpha) \wedge \psi$

Relation avec le cadre AGM

En logique propositionnelle finie, on peut représenter une base de croyances dans le formalisme du cadre AGM par une formule propositionnelle φ telle que :

$$\forall \psi \in K, \varphi \vdash \psi$$

et donc, inversement, on déduit K d'une formule propositionnelle de la façon suivante :

$$K = \{\psi | \varphi \vdash \psi\}$$

Dans ce cas, on a une correspondance entre les opérateurs de révision, pour que $\varphi \circ \alpha$ soit la représentation propositionnelle de $K * \alpha$: $K * \alpha = \{\psi | \varphi \circ \alpha \vdash \psi\}$

Théorème 39 ([KM91])**.**
*Soit $*$ un opérateur de révision sur les théories et $\circ$ un opérateur de révision sur les formules propositionnelles correspondant. $*$ satisfait **(K$*$1) - (K$*$6)** si et seulement si $\circ$ satisfait **(R1) - (R4)**.*
*De plus, en présence des postulats précédents, **(K$*$7)** et **(K$*$8)** sont équivalents respectivement à **(R5)** et **(R6)**.*

2.2.2 Théorème de représentation : les assignements fidèles

On peut, dans le cadre de la logique propositionnelle, trouver une contrepartie à la notion de systèmes de sphères. Il s'agit de la notion d'assignement fidèle.
Katsuno et Mendelzon définissent un assignement fidèle comme une fonction qui associe à une base de croyances φ un pré-ordre $\leqslant_\varphi$ sur les interprétations tel que

– Si $\omega \models \varphi$ et $\omega' \models \varphi$, alors $\omega \simeq_\varphi \omega'$
– Si $\omega \models \varphi$ et $\omega' \not\models \varphi$, alors $\omega <_\varphi \omega'$
– Si $\varphi_1 \equiv \varphi_2$, alors $\leqslant_{\varphi_1} = \leqslant_{\varphi_2}$

Théorème 40 ([KM91]).
*Un opérateur de révision $\circ$ satisfait les postulats **(R1) - (R6)** si et seulement si il existe un assignement fidèle qui associe à chaque base de croyances φ un pré-ordre total $\leqslant_\varphi$ tel que*

$$mod(\varphi \circ \alpha) = min(mod(\alpha), \leqslant_\varphi)$$

On a le même fonctionnement qu'avec les systèmes de sphères. On peut l'expliquer graphiquement grâce au schéma 2.3. Soient L_i les différents « niveaux » qui peuvent être définis à partir du pré-ordre $\leqslant_\varphi$: deux interprétations (représentées par des points sur la figure) ω et ω' sont sur le même niveau si et seulement si $\omega \simeq_\varphi \omega'$, et ω est en dessous de ω' si et seulement si $\omega <_\varphi \omega'$.
Les interprétations situés sur le niveau L_0 sont les modèles de φ, les interprétations situés dans la zone bleue sont les modèles de α. La révision de φ par α est la formule dont les modèles sont sur le segment rouge, c'est-à-dire les modèles de α qui sont les plus plausibles (le plus bas possible).

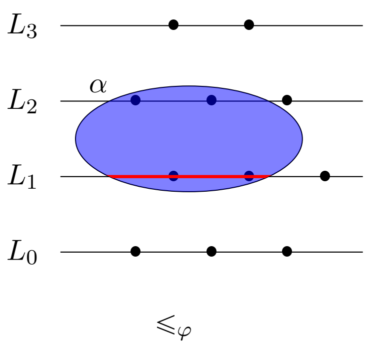

FIGURE 2.3 – Révision de K par α

Chapitre 3

Le changement dans les systèmes d'argumentation

Un certain nombre de travaux ont déjà été conduits sur le changement dans le cadre de l'argumentation. Ce chapitre présente quatre travaux concernant le changement dans les systèmes d'argumentation. Dans un premier temps, nous présentons les travaux de [CdSCLS10], qui consistent à étudier l'effet de l'ajout d'un argument sur un système d'argumentation. Cela permet de dégager des conditions que doit satisfaire un opérateur d'ajout pour que le système résultant ait certaines propriétés.
Ensuite, nous présentons [BCdSCLS11], qui étudie l'intérêt de retirer un argument, par exemple dans le cas d'une procédure légale, certaines informations peuvent se retrouver inutilisables. Sur cet exemple, quelques propriétés sont données, en parallèle à celles concernant l'ajout d'un argument.
Enfin, nous présenterons [BKvdT09a] et [BKvdT09b] qui définissent des principes qui peuvent être (ou non) satisfaits par une sémantique lors de l'ajout ou du retrait d'attaques ou d'arguments. Ces deux papiers se concentrent sur la sémantique de base.

En conclusion, nous expliquerons ce que l'on entend par « révision de systèmes d'argumentation », et comparerons ce but avec les résultats des travaux sus-cités.

Sommaire

3.1 Ajout d'un argument **32**
- 3.1.1 Opérations de changement en argumentation 32
- 3.1.2 Propriétés de structure 33
- 3.1.3 Propriétés sur l'état des arguments 34
- 3.1.4 Liens entre les propriétés 35
- 3.1.5 L'ajout d'argument sous les sémantiques de base et préférée 36

3.2 Retrait d'un argument **39**
- 3.2.1 Un type de changement adapté au retrait d'un argument 39
- 3.2.2 Caractériser la monotonie d'un retrait 40
- 3.2.3 Diverses propriétés sur le changement expansif et le changement restrictif 40

3.3 Principes d'abstraction et de raffinement **40**
- 3.3.1 Le cadre formel de Baroni et Giacomin 41
- 3.3.2 Principes d'abstraction d'attaque 42
- 3.3.3 Principes d'abstraction d'argument 43
- 3.3.4 Raffinement d'attaque 45

3.4 Ce qu'on attend de la révision **46**

3.1 Ajout d'un argument

Un premier problème qui a été traité est de déterminer les conséquences de l'ajout d'un argument sur l'ensemble d'extensions d'un système d'argumentation, ce qui a permis d'identifier diverses propriétés que l'opération de changement induit sur les dites extensions.
Ces propriétés ont deux applications. La première est calculatoire. Il est possible en connaissant ces propriétés de déduire les extensions du nouveau système à partir de celles du système d'origine, sans devoir les calculer entièrement. La seconde application est de pouvoir choisir d'ajouter un « bon » argument pour obtenir le résultat voulu.

On va d'abord caractériser les opérations de changements, avant de donner dans la partie suivante les propriétés qui sont liées à ces changements.

3.1.1 Opérations de changement en argumentation

On peut définir plusieurs types de changements dans un système d'argumentation. Ces changements sont de deux natures, et peuvent porter sur deux parties distinctes du système d'argumentation. On définit donc quatre opérations, qui sont le retrait (ou l'ajout) d'attaques, ou d'un argument avec toutes les attaques qui le concernent.

Définition 36 (Opérations de changement)**.**
Soit $\langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation. On définit les opérations de changement suivantes :
– l'ajout d'une attaque $a = (\alpha, \beta)$ où $\alpha \in \mathcal{A}$ et $\beta \in \mathcal{A}$ est défini par :

$$\langle \mathcal{A}, \mathcal{R} \rangle \oplus_i a = \langle \mathcal{A}, \mathcal{R} \cup \{a\} \rangle$$

– le retrait d'une attaque $a \in \mathcal{R}$ est défini par :

$$\langle \mathcal{A}, \mathcal{R}, \rangle \ominus_i a = \langle \mathcal{A}, \mathcal{R} \backslash \{a\} \rangle$$

– l'ajout d'un argument $\delta \notin \mathcal{A}$ avec un ensemble d'attaques concernant δ, noté $\mathcal{R}_\delta$, est défini par :

$$\langle \mathcal{A}, \mathcal{R} \rangle \oplus_i^a (\delta, \mathcal{R}_\delta) = \langle \mathcal{A} \cup \{\delta\}, \mathcal{R} \cup \mathcal{R}_\delta \rangle$$

Ici, on suppose que $\mathcal{R}_\delta$ est un ensemble non vide d'attaques, c'est-à-dire de couples dans $\mathcal{A} \times \mathcal{A}$, tel qu'un des éléments du couple est δ.
– le retrait d'un argument, et des attaques qui le concernent, est défini par

$$\langle \mathcal{A}, \mathcal{R} \rangle \ominus_i^a (\delta, \mathcal{R}_\delta) = \langle \mathcal{A} \backslash \{\delta\}, \mathcal{R} \backslash \mathcal{R}_\delta \rangle$$

On suppose ici que $\delta \in \mathcal{R}$ et $\mathcal{R}_\delta$ est l'ensemble des attaques $\{(\delta, \gamma) \in \mathcal{R}\} \cup \{(\gamma, \delta) \in \mathcal{R}\}$.

On s'intéresse plus particulièrement au cas où l'ajout (ou retrait) d'un argument concerne un argument qui interagit avec les autres arguments, c'est-à-dire $\mathcal{R}_\delta \neq \emptyset$, le cas restant étant trivial.

Les propriétés présentées dans la suite peuvent être classées dans deux catégories : les propriétés concernant la structure de l'ensemble d'extensions, et les propriétés concernant l'état de certains arguments en particulier. La plupart de ces propriétés peuvent être étudiées pour chacun des changements possibles. Par la suite sont étudiées les conditions nécessaires ou suffisantes pour que les propriétés soient satisfaites pour l'ajout d'un argument, pour deux sémantiques de Dung particulières : l'extension de base, et les extensions préférées.

3.1.2 Propriétés de structure

Un changement est dit décisif s'il permet de désigner explicitement les arguments qui seront acceptés. Pour cela, il faut réduire à une seule extension l'ensemble des extensions du système.

Définition 37 (Changement décisif)**.**
Un changement de AF à AF' est décisif si et seulement si l'ensemble des extensions de AF n'est pas un singleton, et l'ensemble des extensions de AF' est un singleton : $E' = \{\varepsilon\}$.

Une version affaiblie de cette propriété est le cas où le nombre d'extensions est plus petit dans le nouveau système que dans le système de départ : pour accepter sceptiquement un argument par exemple, il suffit qu'il appartienne à p extensions au lieu de n (avec $p < n$). On parle de changement restrictif.

Définition 38 (Changement restrictif)**.**
Le changement de AF à AF' est restrictif si et seulement si les extensions de AF sont $E = \{\varepsilon_1, \ldots, \varepsilon_n\}$ et les extensions de AF' sont $E' = \{\varepsilon'_1, \ldots, \varepsilon'_p\}$, avec $p < n$.

Une propriétée opposée est qu'un changement peut amener plus d'ambiguïté, à cause du nombre d'extensions qui est plus important pour le nouveau système que pour le système de départ. On parle de changement interrogatif.

Définition 39 (Changement interrogatif)**.**
Le changement de AF à AF' est interrogatif si et seulement les extensions de AF' sont $E' = \{\varepsilon'_1, \ldots, \varepsilon'_p\}$ et les extensions de AF sont soit $E = \varnothing$, soit $E = \{\varepsilon_1, \ldots, \varepsilon_n\}$, avec $n < p$.

Si le système de départ a au moins une extension non vide, et que le système résultat n'a qu'une extension vide, ou pas d'extension du tout, le changement est destructif : alors qu'on pouvait accepter des arguments (de façon crédule ou sceptique) avec le système d'origine, le système obtenu après le changement ne permet d'accepter aucun argument.

Définition 40 (Changement destructif)**.**
Le changement de AF à AF' est destructif si et seulement si les extensions de AF sont $E = \{\varepsilon_1, \ldots, \varepsilon_n\}$, $n \geqslant 1$ et l'ensemble des extensions de AF' est soit $\varnothing$, soit $\{\varnothing\}$.

Jusque là, les changements considérés avaient un impact sur le nombre d'extensions. On présente dans la suite divers types de changements qui modifient le contenu des extensions, sans modifier leur nombre. Le premier cas intéressant est le changement expansif : quand chaque extension du nouveau système est incluse dans une extension du système de départ. On a alors au moins autant d'arguments acceptés par l'agent, voire plus.

Définition 41 (Changement expansif)**.**
Le changement de AF à AF' est expansif si et seulement si AF et AF' ont le même nombre d'extensions, et chaque extension de AF est strictement incluse dans une extension de AF'.

Un changement particulier est celui pour lequel les extensions sont exactement les mêmes, dans ce cas l'agent conserve ses croyances sur les arguments acceptés. On parle donc de changement conservatif. On remarque que dans ce cas, on retrouve la notion d'équivalence entre systèmes d'argumentation, il s'agit du premier critère d'équivalence (voir 1.4).

Définition 42 (Changement conservatif)**.**
Le changement de AF à AF' est conservatif si et seulement si ils ont les mêmes extensions, c'est-à-dire $E = E'$.

Enfin, un dernier type de changement est le changement altérant. Il se peut en effet qu'au moins une des extensions soit modifiée, malgré le fait qu'il y en a le même nombre.

Définition 43 (Changement altérant)**.**
Le changement de AF à AF' est altérant si et seulement si AF et AF' ont le même nombre d'extensions, et $\exists \varepsilon \in E$ telle que $\forall \varepsilon' \in E', \varepsilon \nsubseteq \varepsilon'$.

On peut résumer ces définitions via le tableau 3.1. Dans ce tableau, deux cases correspondent à des situations impossibles : $\#1$ et $\#2$. En effet, la seule sémantique d'acceptation dans laquelle un système d'argumentation peut n'avoir aucune extension est la sémantique stable. Cependant, avec cette sémantique, le système ne peut pas avoir une extension vide, sauf si l'ensemble des arguments est vide. Or il est supposé pour les changements étudiés qu'il existe un argument δ ou une attaque a (et donc deux arguments x et y tels que $a = (x, y)$).
Par conséquent, les cas $\#1$ et $\#2$ ne peuvent pas se produire.

<table>
<tr><th></th><th>$E' = \varnothing$</th><th>$E' = \{\varnothing\}$</th><th>$E' = \{\varepsilon'_1\}$</th><th>$E' = \{\varepsilon'_1, \ldots, \varepsilon'_p\}$</th></tr>
<tr><td>$E = \varnothing$</td><td>conservatif</td><td>$\#1$</td><td rowspan="2">décisif</td><td rowspan="2">interrogatif</td></tr>
<tr><td>$E = \{\varnothing\}$</td><td>$\#2$</td><td>conservatif</td></tr>
<tr><td>$E = \{\varepsilon_1\}$</td><td colspan="2" rowspan="2">destructif</td><td>conservatif
expansif
altérant</td><td>interrogatif</td></tr>
<tr><td>$E = \{\varepsilon_1, \ldots, \varepsilon_n\}$</td><td>décisif</td><td>$n < p$:
interrogatif
$n > p$:
restrictif
$n = p$:
conservatif
expansif
altérant</td></tr>
</table>

TABLE 3.1 – Propriétés sur la structure des extensions

Toutes ces propriétés sont mutuellement exclusives : une opération de changement ne peut pas en satisfaire deux à la fois.

3.1.3 Propriétés sur l'état des arguments

Deux propriétés étudiées, concernant l'état d'acceptation des arguments, sont la monotonie et la priorité au plus récent.
La première propriété exprime le fait qu'un argument accepté avant le changement reste accepté après le changement. Trois niveaux de monotonie sont définis.
La deuxième propriété garantit, dans le cas d'un ajout (opération $\oplus_i^a$) que l'argument ajouté est accepté pour le système résultant du changement. Cette propriété rappelle le principe de primauté de la nouvelle information dans le cadre de la dynamique des croyances.

Monotonie

Pour définir cette notion, aucune hypothèse n'est faite concernant le nombre d'extensions et la sémantique. De plus, une propriété adaptée à l'acceptation crédule et une propriété adaptée à l'acceptation sceptique sont présentées.

Dans le cas où il n'y a qu'une extension (ce qui est toujours le cas avec la sémantique de base, par exemple), la notion de monotonie est assez simple. Dire qu'un argument accepté avant le changement reste accepté après le changement revient à dire que tout argument de l'unique extension ε du système de base appartient à l'unique extension ε' du nouveau système.
Dans le cas général où il y a plusieurs extensions, la monotonie peut avoir différentes formes.

Définition 44 (Monotonie, monotonie crédule, monotonie sceptique)**.**

- Le changement de AF à AF' est monotone si et seulement si chaque extension de AF est incluse dans au moins une extension de AF'.
- Le changement de AF à AF' est crédulement monotone si et seulement si l'union des extensions de AF est incluse dans l'union des extensions de AF'.
- Le changement de AF à AF' est strictement monotone si et seulement si l'intersection des extensions de AF est incluse dans l'intersection des extensions de AF'.

Il est évident que dans le cas d'une sémantique garantissant une seule extension, les trois notions coïncident. De plus, la monotonie implique la monotonie crédule. Par contre, la monotonie n'implique pas la monotonie sceptique, et la monotonie sceptique n'implique pas la monotonie.

Ces propriétés sont définies au niveau des extensions, mais une propriété similaire est définie pour les arguments.

Définition 45 (Monotonie partielle pour un argument)**.**
Soit α un argument. Le changement de AF vers AF' est partiellement monotone pour α si et seulement si quand α appartient à une extension de AF, α appartient aussi à au moins une extension de AF'.

Priorité au plus récent

Basée sur les postulats de dynamiques des croyances correspondant au principe de primauté, la propriété de priorité au plus récent permet d'assurer, dans le cas de l'ajout d'un argument (et des attaques qui lui sont associées) que cet argument sera accepté dans le nouveau système d'argumentation.

Définition 46 (Priorité au plus récent)**.**
Le changement $\oplus_i^a$, de AF vers AF' satisfait la priorité au plus récent si et seulement si AF' a au moins une extension et l'argument α ajouté appartient à chaque extension de AF'.

3.1.4 Liens entre les propriétés

Il existe un certain nombre de liens entre les propriétés concernant la structure et les propriétés concernant l'état des arguments. Ces liens sont énoncés par les propositions suivantes.

Proposition 41.
- *Un changement conservatif satisfait la monotonie et la monotonie sceptique.*
- *Un changement expansif satisfait la monotonie et la monotonie sceptique.*
- *Un changement décisif qui satisfait la monotonie satisfait aussi la monotonie sceptique.*
- *Dans le cas particulier d'une sémantique qui assure l'existence d'une unique extension, un changement satisfait la monotonie (et la monotonie sceptique) si et seulement si il est décisif, expansif ou conservatif.*

Proposition 42.
- *Un changement destructif ne satisfait jamais la propriété de monotonie.*
- *Un changement altérant ne satisfait jamais la propriété de monotonie.*
- *Un changement restrictif ne satisfait jamais la propriété de monotonie.*

De plus, on dispose de certains liens entre les propriétés structurelles et la priorité au plus récent dans le cas du changement $\oplus_i^a$.

Proposition 43.
- *Un changement $\oplus_i^a$ conservatif ne satisfait jamais la priorité au plus récent.*
- *Un changement $\oplus_i^a$ destructif ne satisfait jamais la priorité au plus récent.*

Et pour les cas particiliers des sémantiques de base, stable et préférée, on dispose en plus de la propriété suivante :

Proposition 44.
Avec la sémantique de base, la sémantique stable et la sémantique préférée, un changement $\oplus_i^a$ expansif satisfait toujours la priorité au plus récent.

On peut résumer les liens entre propriétés, dans le cas de $\oplus_i^a$, par la figure 3.1.

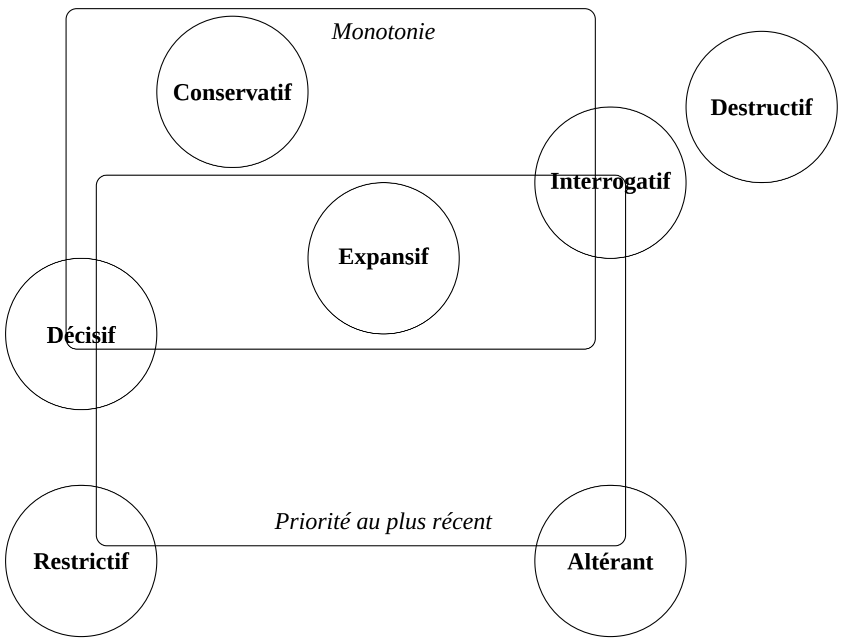

FIGURE 3.1 – Exemple d'attaques entre arguments

3.1.5 L'ajout d'argument sous les sémantiques de base et préférée

Un changement possible dans un système d'argumentation est l'ajout d'exactement un argument, avec les attaques qui le concernent et qui concernent un autre argument de $\mathcal{A}$ (par exemple, l'agent reçoit une nouvelle information α qui contredit une certaine information β que l'agent avait déjà, on ajoute alors l'argument α et l'attaque (α, β)).
Dans la suite, on présente donc les propriétés de l'opération $\oplus_i^a$ sous les sémantiques de base et préférée.

Ajout d'argument sous la sémantique de base

On appelle respectivement $E = \{\varepsilon\}$ et $E' = \{\varepsilon'\}$ les ensembles d'extensions (ici des singletons) de AF et AF'.

Un premier résultat est une condition sous laquelle un argument β, accepté par AF, reste accepté par AF' (c'est-à-dire la propriété de monotonie partielle pour β est vérifiée).

Proposition 45.
Sous la sémantique de base, si $\beta \in \varepsilon$, et l'argument α ajouté n'attaque pas indirectement β, alors $\beta \in \varepsilon'$.

On donne ensuite une condition suffisante pour garantir que l'argument ajouté est accepté (c'est-à-dire la priorité au plus récent).

Proposition 46.
Sous la sémantique de base, si α n'est pas attaqué par un argument de $\mathcal{A}$, alors α est accepté par AF'.

On dispose aussi de propriétés sur le cas particulier où $\varepsilon = \varnothing$.

Proposition 47.
Sous la sémantique de base,
- *Si $\varepsilon = \varnothing$, alors $\varepsilon' = \varnothing$ si et seulement si $\exists \beta \in \mathcal{A}'$ tel que $(\beta, \alpha) \in \mathcal{R}'$.*
- *De plus, si $\varepsilon = \varnothing$ et α n'est pas attaqué par $\mathcal{A}$, alors $\varepsilon' = \{\alpha\} \cup \bigcup\limits_{i \geqslant 1} \mathcal{F}'^i(\{\alpha\})$ (c'est-à-dire α et l'ensemble des arguments qu'il défend indirectement).*

Par conséquent,
- *si $\varepsilon = \varnothing$ et α n'est pas attaqué par $\mathcal{A}$, alors le changement $\oplus_i^a$ est décisif.*
- *Si le changement $\oplus_i^a$ est décisif, alors α n'est pas attaqué par $\mathcal{A}$, et par conséquent α attaque $\mathcal{A}$.*

On s'intéresse maintenant aux cas où $\varepsilon \neq \varnothing$. On présente d'abord une condition suffisante pour que l'opération d'ajout satisfasse la monotonie.

Proposition 48.
Sous la sémantique de base, si $\varepsilon \neq \varnothing$ et α n'attaque pas ε, alors $\oplus_i^a$ satisfait la monotonie (c'est-à-dire $\varepsilon \subseteq \varepsilon'$).

Plus précisément, on peut distinguer deux cas particuliers : celui d'un changement conservatif, et celui d'un changement expansif.

Proposition 49.
Sous la sémantique de base, si $\varepsilon \neq \varnothing$ et α n'attaque pas ε, alors :
- *si ε ne défend pas α, alors $\varepsilon' = \varepsilon$.*
- *si ε défend α, alors $\varepsilon' = \varepsilon \cup \{\alpha\} \cup \bigcup\limits_{i \geqslant 1} \mathcal{F}'^i(\{\alpha\})$.*
  De plus, dans ce cas, si α n'attaque pas $\mathcal{A}$, $\varepsilon' = \varepsilon \cup \{\alpha\}$.

Comme conséquence de la proposition 49, on a une autre condition qui permet de garantir que l'opérateur d'ajout satisfait la propriété de priorité au plus récent.

Proposition 50.
Sous la sémantique de base, si $\varepsilon \neq \varnothing$, α n'attaque pas ε, et ε défend α, alors $\oplus_i^a$ satisfait la propriété de priorité au plus récent, c'est-à-dire $\alpha \in \varepsilon'$.

On constate aussi que certaines propriétés ne peuvent pas être satisfaites par l'opération d'ajout sous la sémantique de base.

Proposition 51.
Sous la sémantique de base, un changement $\oplus_i^a$ ne peut pas être interrogatif ou restrictif.

On note enfin le cas d'un changement destructif, qui peut être obtenu en ajoutant une attaque contre tout argument qui n'était pas attaqué dans AF.

Proposition 52.
Sous la sémantique de base, si $\varepsilon \neq \varnothing$, si α attaque chaque argument de $\mathcal{A}$ non attaqué dans AF, et si α est attaqué dans AF', alors $\oplus_i^a$ est destructif. La réciproque est vraie.

Ajout d'argument sous la sémantique préférée

Contrairement à la sémantique de base, la sémantique préférée ne garantit pas l'existence d'une unique extension. Elle garantit cependant l'existence d'au moins une extension, qui peut être vide. On a donc $E = \{\varepsilon\}$ (avec ε possiblement vide) ou $E = \{\varepsilon_1, \ldots, \varepsilon_n\}$. Pareillement, $E' = \{\varepsilon'\}$ (où ε' peut être vide) ou $E' = \{\varepsilon'_1, \ldots, \varepsilon'_n\}$.

On peut, comme pour la sémantique de base, énoncer des conditions suffisantes ou nécessaires pour que le changement $\oplus_i^a$ vérifie certaines propriétés. Premièrement, on a la condition pour garantir que le changement satisfait la priorité au plus récent.

Proposition 53.
Sous la sémantique préférée, si α n'est pas attaqué par $\mathcal{A}$, alors $\oplus_i^a$ satisfait la priorité au plus récent.

On dispose aussi de propriétés pour identifier les ensembles d'arguments admissibles pour AF'.

Proposition 54.
Sous la sémantique préférée,
- *si α n'attaque pas ε_i, alors ε_i reste admissible pour AF'.*
- *si α n'attaque pas ε_i et ε_i défend α, alors $\varepsilon_i \cup \{\alpha\}$ est admissible pour AF'.*

En conséquence de cette proposition, on a une autre condition qui permet de garantir que l'ajout d'un argument satisfait la priorité au plus récent.

Proposition 55.
Sous la sémantique préférée, si α n'attaque aucune extension de AF, et si ε_i défend α, $\forall i$, alors $\oplus_i^a$ satisfait la priorité au plus récent.

On peut identifier des cas où l'ajout d'un argument est décisif.

Proposition 56.
Sous la sémantique préférée, si $E = \{\varnothing\}$, α n'est pas attaqué par $\mathcal{A}$, et il n'y a pas de cycle de longueur paire dans le graphe d'attaques, alors $E' = \{\varepsilon'\}$ et $\alpha \in \varepsilon'$.

On présente ensuite une condition nécessaire pour que le changement soit décisif.

Proposition 57.
Sous la sémantique préférée, si α n'attaque aucun argument de $\mathcal{A}$ et $E = \{\varnothing\}$, alors $E' = \{\varnothing\}$.
De façon équivalente, si $E = \{\varnothing\}$, alors le changement $\oplus_i^a$ est décisif si et seulement si α attaque $\mathcal{A}$.

La proposition suivante concerne le cas où il existe une extension non vide de AF, et donne une condition pour que $\oplus_i^a$ soit un changement conservatif ou un changement expansif.

Proposition 58.
Sous la sémantique préférée, si α n'attaque aucun argument de $\mathcal{A}$, et $E \neq \{\varnothing\}$, alors $\forall i$:
- *si ε_i défend α, alors $\varepsilon_i \cup \{\alpha\}$ est une extension de AF'.*
- *si ε_i ne défend pas α, alors ε_i est une extension de AF'.*

De plus, AF et AF' ont le même nombre d'extensions.

On en déduit donc une condition qui garantit que le changement soit monotone.

Proposition 59.
Sous la sémantique préférée, si α n'attaque aucune extension de AF, alors le changement $\oplus_i^a$ est monotone.

Dans le cas particulier des systèmes d'argumentation sans controverse, on obtient une condition sous laquelle le changement $\oplus_i^a$ satisfait la monotonie sceptique.

Proposition 60.
Sous la sémantique préférée, on suppose que AF ne contient aucun argument controversé. Si α n'attaque pas $\bigcap_{i \leqslant 1} \varepsilon_i$, alors le changement $\oplus_i^a$ satisfait la monotonie sceptique, c'est-à-dire $\bigcap_{i \leqslant 1} \varepsilon_i \subseteq \bigcap_{i \leqslant 1} \varepsilon'_i$

Enfin, comme pour la sémantique de base, on peut donner une condition pour que le changement soit destructif.

Proposition 61.
Sous la sémantique préférée, si $E \neq \{\varnothing\}$, qu'il n'y a pas de cycle de longueur paire dans AF', si chaque argument non attaqué $\beta_i \in \mathcal{A}$ est attaqué par $\mathcal{A}'$, et si α est attaqué par $\mathcal{A}'$, alors le changement $\oplus_i^a$ est destructif.

3.2 Retrait d'un argument

Des résultats sur le retrait d'un argument du système ont été proposés ([BCdSCLS11]), pour arriver à des propriétés similaires à celles présentées sur l'ajout d'un argument. La définition utilisée pour le retrait d'un argument est la même que celle présentée dans la partie 3.1.1, c'est-à-dire

$$\langle \mathcal{A}, \mathcal{R} \rangle \ominus_i^a (\delta, \mathcal{R}_\delta) = \langle \mathcal{A} \backslash \{\delta\}, \mathcal{R} \backslash \mathcal{R}_\delta \rangle$$

avec $\delta \in \mathcal{R}$ et $\mathcal{R}_\delta$ est l'ensemble des attaques $\{(\delta, \gamma) \in \mathcal{R}\} \cup \{(\gamma, \delta) \in \mathcal{R}\}$.

3.2.1 Un type de changement adapté au retrait d'un argument

Le changement considéré ici peut satisfaire les propriétés définies dans la partie précédente, mais dans le cas particulier du retrait d'un argument, qui est en quelque sort le « dual » de l'ajout d'un argument, il est intéressant de définir un type de changement dual du changement expansif.

Définition 47 (Changement restrictif)**.**
Le changement d'un système AF vers un système AF' est restrictif si et seulement si :
- $E \neq \varnothing$, $|E| = |E'|$
- $\forall \varepsilon'_j \in E'$, $\exists \varepsilon_i \in E$ tel que $\varepsilon'_j \subset \varepsilon_i$
- $\forall \varepsilon_i \in E$, $\exists \varepsilon'_j \in E'$ tel que $\varepsilon'_j \subset \varepsilon_i$

3.2.2 Caractériser la monotonie d'un retrait

Il est possible de donner sur le retrait des conditions suffisantes pour que le changement soit monotone, au sens donné dans la partie précédente.

Proposition 62 (Conditions suffisantes pour la monotonie/non-monotonie)**.**
Quand on retire un argument δ du système d'argumentation AF, sous la sémantique préférée, stable ou de base,
- *si* $\exists \varepsilon_i \in E$ *tel que* $\delta \in \varepsilon_i$, *alors* $\exists \varepsilon_j \in E$ *tel que* $\forall \varepsilon' \in E'$, $\varepsilon_j \not\subseteq \varepsilon'$
- *si* $\nexists \varepsilon_i \in E$ *tel que* $\delta \in \varepsilon_i$, *alors* $\forall \varepsilon_j \in E$, $\exists \varepsilon' \in E'$ *tel que* $\varepsilon_j \subseteq \varepsilon'$.

On peut aussi établir une propriété concernant la monotonie faible, c'est-à-dire la conservation d'une extension sans prendre en compte l'argument retiré (cf [BKvdT09a]).

Proposition 63.
Quand on retire un argument δ du système d'argumentation AF, si δ n'attaque aucun argument alors
- $\forall \varepsilon$ *extension préférée de* AF, $\varepsilon \backslash \{\delta\}$ *est admissible pour le système* AF', *et par conséquent* $\exists \varepsilon'$ *extension préférée de* AF' *telle que* $\varepsilon \backslash \{\delta\} \subseteq \varepsilon'$.
- $\forall \varepsilon$ *extension stable de* AF, $\varepsilon \backslash \{\delta\}$ *est une extension stable de* AF'.

3.2.3 Diverses propriétés sur le changement expansif et le changement restrictif

Le cas du changement expansif

On rappelle qu'un changement expansif accroit la taille des extensions. On peut énoncer des propriétés qui indiquent quand le retrait d'un argument peut être expansif.

Proposition 64.
Il est impossible d'avoir une changement $\ominus_i^a$ expansif sous la sémantique stable.

Proposition 65.
Quand on retire l'argument δ du système AF sous la sémantique préférée ou de base, si le changement est expansif alors
- δ *n'appartient à aucune extension de* AF,
- *et* δ *attaque au moins un argument de* AF.

Le cas du changement restrictif

Le changement restrictif peut être vu comme le dual du changement expansif, puisqu'il restreint la taille des extensions. Cela peut être désirable pour un agent dans un débat, pour réduire les possibilités d'argumentation de la partie adverse. La propriété suivante donne une condition nécessaire pour obtenir un changement restrictif.

Proposition 66.
Quand on retire l'argument δ du système d'argumentation AF, sous la sémantique préférée, stable ou de base, si le changement est restrictif alors il existe une extension ε de AF telle que $\delta \in \varepsilon$.

3.3 Principes d'abstraction et de raffinement

Le travail de Boella, Kaci et van der Torre ([BKvdT09a]) qu'on souhaite maintenant présenter se concerne l'abstraction. Cette notion est définie comme suit.

Définition 48 (Abstraction)**.**
Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ et $AF' = \langle \mathcal{A}', \mathcal{R}' \rangle$ deux systèmes d'argumentation.
- AF est une abstraction d'argument de AF' si et seulement si $\mathcal{A} \subseteq \mathcal{A}'$, et $\forall a \in \mathcal{A}, \forall b \in \mathcal{A}, (a,b) \in \mathcal{R}$ si et seulement si $(a,b) \in \mathcal{R}'$.
- AF est une abstraction d'attaque de AF' si et seulement si $\mathcal{A} = \mathcal{A}'$ et $\mathcal{R} \subseteq \mathcal{R}'$.
- AF est une abstraction d'argument et d'attaque de AF' si et seulement si $\mathcal{A} \subseteq \mathcal{A}'$ et $\mathcal{R} \subseteq \mathcal{R}'$.

Concrètement, une abstraction est le retrait d'arguments de $\mathcal{A}$ (avec conservation des attaques pour les arguments restants), ou le retrait d'attaques (sans toucher aux arguments), ou enfin le retrait d'arguments et d'attaques.

Ce travail se base sur le cadre d'argumentation défini par Baroni et Giacomin ([BG07]). On commence donc par présenter ce cadre.

3.3.1 Le cadre formel de Baroni et Giacomin

Le cadre de Baroni et Giacomin est basé sur une fonction $\mathcal{E}$ qui associe à un système d'argumentation $\langle \mathcal{A}, \mathcal{R} \rangle$ ses extensions, c'est-à-dire à un ensemble d'ensembles d'arguments. Cette fonction est appelée fonction d'acceptation dans le papier de Boella, Kaci et van der Torre.

Définition 49 (Fonction d'acceptation)**.**
Soit $\mathcal{N}$ l'univers des arguments. Une fonction d'acceptation de multiples extensions $\mathcal{E} : \mathcal{N} \times 2^{\mathcal{N} \times \mathcal{N}} \to 2^{2^{\mathcal{N}}}$ est
- une fonction partielle qui est définie pour chaque système d'argumentation $\langle \mathcal{A}, \mathcal{R} \rangle$ avec $\mathcal{A} \subseteq \mathcal{N}$ fini et $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$, et
- qui associe à un système d'argumentation $\langle \mathcal{A}, \mathcal{R} \rangle$ un ensemble de sous-ensembles de $\mathcal{A}$: $\mathcal{E}(\langle \mathcal{A}, \mathcal{R} \rangle) \subseteq 2^{\mathcal{A}}$.

La généralité de ce cadre formel vient de la définition de divers principes qui sont intégrés au cadre de Dung, par exemple le principe d'indépendance du langage et le principe de liberté de conflit.

Définition 50 (Indépendance du langage)**.**
Deux systèmes d'argumentation $AF_1 = \langle \mathcal{A}_1, \mathcal{R}_1 \rangle$ et $AF_2 = \langle \mathcal{A}_2, \mathcal{R}_2 \rangle$ sont isomorphes si et seulement s'il existe une bijection $m : \mathcal{A}_1 \to \mathcal{A}_2$ tel que $(\alpha, \beta) \in \mathcal{R}_1$ si et seulement si $(m(\alpha), m(\beta)) \in \mathcal{R}_2$. On écrit alors $AF_1 \doteq_m AF_2$.
Une sémantique $\mathcal{S}$ satisfait le principe d'indépendance du langage si et seulement si $\forall AF_1 = \langle \mathcal{A}_1, \mathcal{R}_1 \rangle$, $\forall AF_2 = \langle \mathcal{A}_2, \mathcal{R}_2 \rangle$ tel que $AF_1 \doteq_m AF_2$, on a $\mathcal{E}_{\mathcal{S}}(AF_2) = \{M(E) | E \in \mathcal{E}_{\mathcal{S}}(AF_1)\}$, où $M(E) = \{\beta \in \mathcal{A}_2 | \exists \alpha \in E, \beta = m(\alpha)\}$.

Définition 51 (Absence de conflit)**.** Étant donné un système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, un ensemble $S \subseteq \mathcal{A}$ est sans conflit, noté $cf(S)$ (pour *conflict free*) si et seulement si $\nexists a, b \in S$ tels que a attaque b.
Une sémantique $\mathcal{S}$ satisfait le principe de liberté de conflit si et seulement si $\forall AF, \forall E \in \mathcal{E}_{\mathcal{S}}(AF)$, E est sans conflit.

Le cas d'une extension unique L'article de Boella, Kaci et van der Torre traite uniquement le cas des sémantiques qui garantissent l'existence d'une unique extension, comme la sémantique de base. Dans ce cas, la définition de la fonction d'acceptation est adaptée.

Définition 52 (Fonction d'acceptation)**.**
Soit $\mathcal{N}$ l'univers des arguments. Une fonction d'acceptation d'une unique extension

$$\mathcal{E} : \mathcal{N} \times 2^{\mathcal{N}\times\mathcal{N}} \to 2^{2^{\mathcal{N}}}$$

est
- une fonction totale qui est définie pour chaque système d'argumentation $\langle \mathcal{A}, \mathcal{R} \rangle$ avec $\mathcal{A} \subseteq \mathcal{N}$ fini et $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$, et
- qui associe à un système d'argumentation $\langle \mathcal{A}, \mathcal{R} \rangle$ un sous-ensemble de $\mathcal{A}$: $\mathcal{E}(\langle \mathcal{A}, \mathcal{R} \rangle) \subseteq \mathcal{A}$.

Les principes définis pour les fonctions d'acceptation multiples sont toujours définis, puisque le cas d'une seule extension est un cas particulier du cas des extensions multiples. Par exemple, une sémantique $\mathcal{S}$ satisfait le principe de liberté de conflit quand l'unique extension est sans conflit.

Avant de présenter les principes d'abstraction, on introduit une notation basée sur les *labellings* de Caminada.

Définition 53 (Arguments rejetés et non décidés)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation tel que $\mathcal{E}(AF)$ est l'unique extension, sans conflit, du système, alors $\mathcal{A}$ est partitionné en $\mathbf{A}(AF), \mathbf{R}(AF)$ et $\mathbf{U}(AF)$, avec
- $\mathbf{A}(AF) = \mathcal{E}(AF)$ est l'ensemble des arguments acceptés,
- $\mathbf{R}(AF) = \{a \in \mathcal{A} | \exists x \in \mathbf{A}(AF) : (x, a) \in \mathcal{R}\}$ est l'ensemble des arguments rejetés,
- $\mathbf{U}(AF) = \mathcal{A} \backslash (\mathbf{A}(AF) \cup \mathbf{R}(AF))$ est l'ensemble des arguments non décidés.

Ces trois notations correspondent respectivement à l'ensemble des arguments qui sont in pour chaque *labelling* dans la sémantique choisie, à l'ensemble des arguments qui sont out pour chaque *labelling* dans la sémantique choisie, et à l'ensemble des arguments qui ne sont ni toujours in, ni toujours out.

3.3.2 Principes d'abstraction d'attaque

On présente dans cette partie la situation où l'ensemble d'arguments reste le même, mais où la relation d'attaque peut être réduite. En l'occurrence, on retire une unique attaque (a, b) de la relation, avec une distinction selon le statut de a et b.

Définition 54 (Principe d'abstraction d'attaque)**.**
Une fonction d'acceptation $\mathcal{E}$ satisfait le principe d'abstraction d'attaque XY, où $X, Y \in \{\mathbf{A}, \mathbf{R}, \mathbf{U}\}$, si quel que soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, $\forall a \in X(AF), \forall b \in Y(AF)$, on a $\mathcal{E}(\langle \mathcal{A}, \mathcal{R} \backslash \{(a, b)\} \rangle) = \mathcal{E}(AF)$.

Ce principe signifie que le retrait d'une attaque ne change pas l'ensemble des arguments acceptés si l'attaque a lieu entre des arguments dont l'état d'acceptation est déterminé par XY. On remarque que le changement décrit ici est conservatif, c'est-à-dire que le système de départ et le système après retrait d'une attaque sont équivalents selon le premier critère.

Les principes satisfaits par la sémantique de base sont connus.

Proposition 67.
*La sémantique de base satisfait les principes d'abstraction d'attaque **AA**, **AU**, **UA**, **UR**, **R**, **A**, **RU** et **RR**, mais ne satisfait pas les principes d'abstraction d'attaque **AR** et **UU**.*

Deux cas sont intéressants, celui du retrait d'une attaque d'un argument accepté vers un argument rejeté (**AR**), et celui du retrait d'une attaque entre deux arguments non décidés (**UU**). Pour ces deux cas, il faut trouver sous quelles conditions les extensions restent les mêmes.
Pour cela, la notion d'abstraction d'attaque conditionnelle est introduite. L'idée est que si l'on retire une attaque (a, b) de type **AR** ou **UU**, alors il y a une autre raison qui empêche b d'être accepté.

Définition 55 (Principe d'abstraction d'attaque conditionnelle)**.**
Une fonction d'acceptation $\mathcal{E}$ satisfait le principe d'abstraction d'attaque conditionnelle $XY(Z)$, où $X, Y, Z \in \{\mathbf{A}, \mathbf{R}, \mathbf{U}\}$, si pour tout système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$,

$$\forall a \in X(AF), \forall b \in Y(AF), \text{ si } \exists c \in Z(AF) \text{ tel que } c \neq a \text{ et } (c, b) \in \mathcal{R},$$
$$\text{alors } \mathcal{E}(\langle \mathcal{A}, \mathcal{R} \backslash \{(a, b)\} \rangle) = \mathcal{E}(AF)$$

Proposition 68.
*La sémantique de base satisfait les principes d'abstraction **UU**(**A**) et **UU**(**U**). Elle ne satisfait pas les principes **AR**(**A**), **AR**(**U**), **AR**(**R**) et **UU**(**R**).*

Deux autres principes peuvent être définis pour traiter les cas **AR**(**A**) et **AR**(**U**). Pour le premier, il faut envisager le cas où en retirant une attaque (a, b), b reste rejeté. Cela signifie qu'il y a un argument accepté c, autre que a, qui attaque b. Cependant, un contre-exemple pourrait illustrer une faiblesse de cette idée : le cas où c est accepté grâce au rejet de b !
Il faut donc éviter qu'il y ait un chemin de longueur impaire de b à c, c'est-à-dire éviter que le rejet de b soit la cause de l'acceptation de c.

Définition 56 (Principe d'abstraction d'attaque conditionnelle acyclique)**.**
Une fonction d'acceptation $\mathcal{E}$ satisfait le principe d'abstraction d'attaque conditionnelle acyclique $XY(Z)$, avec $X, Y, Z \in \{\mathbf{A}, \mathbf{R}, \mathbf{U}\}$, si quel que soit le système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$,

$$\forall a \in X(AF), \forall b \in Y(AF), \text{ si } \exists c \in Z(AF) \text{ tel que } c \neq a, (c, b) \in \mathcal{R}$$
$$\text{et il n'y a pas de suite d'attaques de longueur impaire de } b \text{ à } c, \text{ alors } \mathcal{E}(\langle \mathcal{A}, \mathcal{R} \backslash \{(a, b)\} \rangle) = \mathcal{E}(AF)$$

A présent, considérons le principe **AR**(**U**). Dans ce cas, un argument accepté a attaque un argument rejeté b, qui est aussi attaqué par un argument non décidé c. Par conséquent, l'argument b peut devenir non décidé, il n'appartient toujours pas à une extension, mais ce changement peut avoir des conséquences sur d'autres parties du système d'argumentation. b ne doit donc pas être la cause d'acceptation d'un autre argument.

Définition 57 (Principe d'abstraction d'attaque conditionnelle fort)**.**
Une fonction d'acceptation $\mathcal{E}$ satisfait le principe d'abstraction $XY(Z, W)$, avec $X, Y, Z, W \in \{\mathbf{A}, \mathbf{R}, \mathbf{U}\}$, si pour tout système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$

$$\forall a \in X(AF), \forall b \in Y(AF), \text{ si } \exists c \in Z(AF) \text{ tel que } c \neq a, (c, b) \in \mathcal{R} \text{ et}$$
$$\forall d \in W(AF), (b, d) \notin \mathcal{R}, \text{ alors } \mathcal{E}(\langle \mathcal{A}, \mathcal{R} \backslash \{(a, b)\} \rangle) = \mathcal{E}(AF)$$

Proposition 69.
*La sémantique de base satisfait le principe d'abstraction **AR**(**U**, **A**).*

3.3.3 Principes d'abstraction d'argument

Après avoir défini plusieurs principes pour le retrait d'une attaque, sans toucher à l'ensemble $\mathcal{A}$, Boella, Kaci et van der Torre ont étudié le retrait d'un argument (et des attaques qui le concernent).

Comme dans la partie précédente, plusieurs cas doivent être considérés, en fonction de l'état de l'argument supprimé : accepté, rejeté ou non décidé.

On rappelle la notation $\mathcal{R}_a$ pour désigner les attaques qui concernent l'argument a. Ceci étant dit, on peut introduire le principe d'abstraction d'argument.

Définition 58 (Principe d'abstraction d'argument)**.**
Une fonction d'acceptation $\mathcal{E}$ satisfait le principe d'abstraction d'argument $X \in \{\mathbf{A}, \mathbf{R}, \mathbf{U}\}$ si quel que soit le système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$,

$$\text{si } a \in X(AF), \text{ alors } \mathcal{E}(\langle \mathcal{A} \backslash \{a\}, \mathcal{R} \backslash \mathcal{R}_a \rangle) = \mathcal{E}(AF) \backslash \{a\}$$

On remarque que lorsque ce principe est respecté, le changement est conservatif, alors que les retraits d'arguments caractérisés dans la partie précédente étaient expansif et restrictif.
Ce principe n'est respecté que dans le cas **R**. En effet, si on retire un argument accepté par exemple, alors les arguments qu'il attaquait peuvent devenir acceptés, et donc appartenir à $\mathcal{E}(AF')$, où AF' est le système issu de l'abstraction.

Proposition 70.
*L'extension de base satisfait le principe d'abstraction d'argument **R**, et ne satisfait pas les principes d'abstraction d'argument **A** et **U**.*

Pour les deux cas restants, les auteurs ont proposé des principes d'abstraction adaptés. Présentons d'abord ceux qui concernent la suppression d'arguments non décidés, qu'on appelle abstraction d'**U**-argument.
Contrairement à un argument rejeté, dont les attaques sont sans effet, un argument non décidé peut empêcher un argument d'être accepté, et par conséquent le retrait d'un de ces arguments peut changer l'extension. On peut garantir que l'extension ne changera pas en retirant un argument non décidé seulement si les arguments qu'il attaque sont dans l'extension, c'est-à-dire les attaques en question ont été « inefficaces », ou s'ils sont hors de l'extension à cause d'autres arguments.
Les principes suivants sont définis :

Définition 59 (Principe d'abstraction d'**U**-argument 1)**.**
Une fonction d'acceptabilité $\mathcal{E}$ satisfait le principe d'abstraction d'**U**-argument 1 si pour tout système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$,

$$\text{si } a \in \mathbf{U}(AF) \text{ et } a \text{ attaque seulement des arguments de } \mathbf{A}(AF), \text{ alors}$$
$$\mathcal{E}(\langle \mathcal{A} \backslash \{a\}, \mathcal{R} \backslash \mathcal{R}_a \rangle) = \mathcal{E}(AF)$$

Définition 60 (Principe d'abstraction d'**U**-argument 2)**.**
Une fonction d'acceptabilité $\mathcal{E}$ satisfait le principe d'abstraction d'**U**-argument 2 si pour tout système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$,

$$\text{si } a \in \mathbf{U}(AF) \text{ et } a \text{ attaque seulement des arguments de } \mathbf{R}(AF), \text{ alors}$$
$$\mathcal{E}(\langle \mathcal{A} \backslash \{a\}, \mathcal{R} \backslash \mathcal{R}_a \rangle) = \mathcal{E}(AF)$$

Définition 61 (Principe d'abstraction d'**U**-argument 3)**.**
Une fonction d'acceptabilité $\mathcal{E}$ satisfait le principe d'abstraction d'**U**-argument 3 si pour tout système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$,

$$\text{si } a \in \mathbf{U}(AF) \text{ et } \forall b \in \mathbf{U}(AF) \text{ tel que } (a, b) \in \mathcal{R}, \exists c \in \mathbf{U}(AF), c \neq b \text{ et } (c, b) \in \mathcal{R}, \text{ alors}$$
$$\mathcal{E}(\langle \mathcal{A} \backslash \{a\}, \mathcal{R} \backslash \mathcal{R}_a \rangle) = \mathcal{E}(AF)$$

Proposition 71.
*La sémantique de base satisfait les principes d'abstraction d'**U**-argument* 1, 2 *et* 3.

Enfin, deux principes sont donnés pour le retrait d'un argument accepté (abstraction d'**A**-argument) : si on supprime un argument qui n'attaque aucun autre argument, alors l'extension reste inchangée, de même si l'argument qu'on supprime n'attaque que des arguments qui sont attaqués par un autre argument.

Définition 62 (Principe d'abstraction d'**A**-argument 1)**.**
Une fonction d'acceptabilité $\mathcal{E}$ satisfait le principe d'abstraction d'**U**- argument 1 si pour tout système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$,

$$\text{si } a \in \mathbf{A}(AF) \text{ et } a \text{ n'attaque aucun argument } b \neq a, \text{ alors}$$
$$\mathcal{E}(\langle \mathcal{A} \backslash \{a\}, \mathcal{R} \backslash \mathcal{R}_a \rangle) = \mathcal{E}(AF) \backslash \{a\}$$

Définition 63 (Principe d'abstraction d'**A**-argument 2)**.**
Une fonction d'acceptabilité $\mathcal{E}$ satisfait le principe d'abstraction d'**U**- argument 2 si pour tout système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$,

$$\text{si } a \in \mathbf{A}(AF) \text{ et } \forall b \in \mathbf{R}(AF) \text{ tel que } (a,b) \in \mathcal{R}, \exists c \in \mathbf{A}(AF), c \neq a, (c,b) \in \mathcal{R} \text{ alors}$$
$$\mathcal{E}(\langle \mathcal{A} \backslash \{a\}, \mathcal{R} \backslash \mathcal{R}_a \rangle) = \mathcal{E}(AF) \backslash \{a\}$$

Proposition 72.
*La sémantique de base satisfait les principes d'abstraction d'**A**-argument* 1 *et* 2.

3.3.4 Raffinement d'attaque

La dernière partie des travaux de Boella, Kaci et van der Torre qu'on souhaite présenter concerne le raffinement d'attaque ([BKvdT09b]). Cette notion, duale de l'abstraction, consiste en l'ajout (au lieu de la suppression) d'attaques à la relation $\mathcal{R}$, dans le cas qui nous intéresse. De façon générale, le raffinement est défini ainsi :

Définition 64 (Raffinement)**.**
Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ et $AF' = \langle \mathcal{A}', \mathcal{R}' \rangle$ deux systèmes d'argumentation.
- AF est un raffinement d'argument de AF' si et seulement si $\mathcal{A}' \subseteq \mathcal{A}$, et $\forall a, b \in \mathcal{A}'$, $(a,b) \in \mathcal{R}$ si et seulement si $(a,b) \in \mathcal{R}'$.
- AF est un raffinement d'attaque de AF' si et seulement si $\mathcal{A} = \mathcal{A}'$ et $\mathcal{R}' \subseteq \mathcal{R}$.
- AF est un raffinement d'argument et d'attaque de AF' si et seulement si $\mathcal{A}' \subseteq \mathcal{A}$ et $\mathcal{R}' \subseteq \mathcal{R}$.

On énonce brièvement trois principes présentés pour le raffinement d'attaque, en particulier ici l'ajout d'une attaque (a,b) au système d'argumentation.

Définition 65 (Principe de raffinement d'attaque)**.**
Une fonction d'acceptation $\mathcal{E}$ satisfait le principe de raffinement d'attaque XY, avec $X, Y \in \{\mathbf{A}, \mathbf{R}, \mathbf{U}\}$ si quel que soit le système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$,

$$\forall a \in X(AF), \forall b \in Y(AF), \mathcal{E}(\langle \mathcal{A}, \mathcal{R} \cup \{(a,b)\} \rangle) = \mathcal{E}(AF)$$

Définition 66 (Principe de raffinement d'attaque acyclique)**.**
Une fonction d'acceptation $\mathcal{E}$ satisfait le principe de raffinement d'attaque acyclique XY, avec $X, Y \in \{\mathbf{A}, \mathbf{R}, \mathbf{U}\}$ si quel que soit le système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$,

$$\forall a \in X(AF), \forall b \in Y(AF), \text{ s'il n'y a pas de suite d'attaques de } b \text{ à } a \text{ de longueur impaire,}$$
$$\text{alors } \mathcal{E}(\langle \mathcal{A}, \mathcal{R} \cup \{(a,b)\} \rangle) = \mathcal{E}(AF)$$

Définition 67 (Principe de raffinement d'attaque conditionnel)**.**
Une fonction d'acceptation $\mathcal{E}$ satisfait le principe de raffinement d'attaque conditionnel $XY(ZT)$, avec $X, Y, Z, T \in \{\mathbf{A}, \mathbf{R}, \mathbf{U}\}$ si quel que soit le système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$,

$$\forall a \in X(AF), \forall b \in Y(AF), \text{ si} \forall c \in Z(AF) \text{ tel que } (b, c) \in \mathcal{R}, \exists d \neq b \in T(AF), (d, c) \in \mathcal{R}, \\ \text{alors } \mathcal{E}(\langle \mathcal{A}, \mathcal{R} \cup \{(a, b)\} \rangle) = \mathcal{E}(AF)$$

La proposition suivante fusionne trois propositions données dans le papier de Boella, Kaci et van der Torre, qui indiquent les principes de raffinement d'attaque qui sont satisfaits par la sémantique de base.

Proposition 73.
La sémantique de base satisfait
- *les principes de raffinement d'attaque **AR**, **RR**, **UR**, **RU** et **UU**,*
- *le principe de raffinement d'attaque acyclique **RA**,*
- *et le principe de raffinement d'attaque conditionnel **AU**(**UU**).*

3.4 Ce qu'on attend de la révision

On émet deux principales critiques sur les travaux mentionnés dans les parties précédentes.

La première est le fait que ces changements sont des changements sur l'ensemble d'arguments du système : à notre sens, rajouter ou supprimer un argument revient à modifier le langage dans lequel sont exprimées les croyances de l'agent, on lui donne la possibilité d'exprimer une croyance (le nouvel argument) qu'il n'aurait pas pu exprimer précédemment.
Pour comparer avec la révision en logique propositionnelle, on rappelle qu'une révision de formule logique ne rajoute pas de nouvelles variables au langage. Plus formellement, si V est l'ensemble des variables appartenant au langage logique avec lequel raisonne l'agent, V ne change pas lors d'une révision. On préfère donc, autant que possible, définir la révision comme un changement dans la relation d'attaque, considérant ainsi que les attaques entre les arguments sont des croyances qui peuvent aussi être remises en cause. On peut se permettre ce point de vue en argumentation abstraite, car la relation d'attaque n'est pas guidée ici par un raisonnement explicite, contrairement à l'argumentation basée sur la logique par exemple.

L'autre critique est qu'il n'y a pas de proposition de postulats pour régir les opérations de changements, et assurer qu'elles sont raisonnables. Ce qui est fait, globalement, est de dire « en effectuant tel type de changement, on va obtenir tel type de résultat ». De plus, les seules considérations quant au statut des arguments sont de vérifier si les arguments qui étaient acceptés le restent, ou si le nouvel argument est accepté, mais ces travaux n'envisagent pas de méthode pour changer le statut d'un argument. Or typiquement, la révision est le changement de statut d'une croyance de l'agent, avec un changement minimal sur la base de croyances. Par conséquent, la question à se poser est « comment changer de façon minimale le graphe du système d'argumentation pour pouvoir changer le statut d'acceptation de certains arguments ? ». De plus, dans la suite de ce mémoire, les travaux présentés portent sur des systèmes d'argumentation sans boucle. On considère qu'un argument qui s'attaque lui même n'a pas de sens, et par conséquent on décide de les interdire.
Il convient donc de définir un langage et une sémantique associée pour exprimer le fait qu'un agent accepte ou refuse une croyance exprimée au moyen d'arguments, et d'utiliser ce langage pour proposer des postulats auxquels on associera des opérateurs.

Deuxième partie

Etude de la révision de systèmes d'argumentation

Chapitre 1

Définition formelle de la révision de systèmes d'argumentation

On rappelle que la dynamique des croyances est le fait de changer le statut d'une croyance. Si on considère qu'un argument est une croyance de l'agent, il faut donc voir ce type de changement comme une modification de la structure du graphe pour qu'un certain argument qui n'était pas accepté le soit, par exemple. On souhaite définir des opérations qui correspondent à celles présentées à la figure 2.1. Il faut pour cela définir, comment, dans un système d'argumentation, signifier qu'un argument est accepté, rejeté ou indéterminé. On va définir un langage qui permet d'exprimer le fait qu'un système d'argumentation infère une certaine information « complexe », en se basant sur l'inférence sceptique d'un argument par le système (information « basique »). Selon le cadre utilisé, on peut associer trois sémantiques différentes aux formules de ce langage.

Une fois ce langage défini, on définit formellement ce qu'est la révision de systèmes d'argumentation, exprimée dans trois cadres différents, et on approfondit le cas de *labellings* qui permet de définir simplement les autres opérations de dynamique sur les systèmes d'argumentation (expansion et contraction).

Enfin, il est intéressant de disposer de postulats de rationalité pour une opération telle que la révision de systèmes d'argumentation. On rappelle que de tels postulats sont des propriétés qui doivent être respectées par un opérateur pour qu'on juge son comportement raisonnable. On définit de tels postulats adaptés à la révision de systèmes d'argumentation, exprimés dans les trois cadres qui ont été évoqués précédemment.

Sommaire

1.1 Définition de la révision **50**

1.1.1 Formules sur les arguments : un langage pour exprimer les informations inférées à partir d'un système d'argumentation 50

1.1.2 Formalisation de la révision de systèmes d'argumentation 52

1.1.3 Partir des *labellings* pour définir la dynamique dans les systèmes d'argumentation 52

1.2 Postulats pour la révision de systèmes d'argumentation **55**

1.2.1 Des postulats dans d'autres cadres 55

1.2.2 Traduction des postulats algébriques 55

1.2.3 Une autre notion intéressante : l'indépendance 58

1.2.4 Postulats de rationalité 58

1.1 Définition de la révision

1.1.1 Formules sur les arguments : un langage pour exprimer les informations inférées à partir d'un système d'argumentation

On a défini dans la partie précédente de ce mémoire ce que sont l'inférence sceptique et l'inférence crédule dans le cadre des systèmes d'argumentation. On a vu qu'il est possible d'exprimer le fait qu'un système infère un argument, c'est-à-dire que cet argument appartient à chaque (ou au moins une si on utilise l'inférence crédule) extension du système d'argumentation. On souhaite cependant pouvoir inférer des informations plus complexes qu'un argument, par exemple on souhaite pouvoir exprimer le fait que le système infère « l'argument α est accepté, et l'argument β est refusé », ou encore « l'argument α est accepté ou l'argument β est accepté ». On définit pour cela un langage adapté, doté de plusieurs sémantiques possibles.

Définition 68 (Formule sur les arguments)**.**
On définit une formule sur les arguments comme un élément du langage donné par la grammaire suivante :

$$\Phi ::= \alpha | \neg \Phi | \Phi \wedge \Phi | \Phi \vee \Phi$$

On peut associer plusieurs sémantiques aux formules de ce langage, qui dépendent du sens qu'on donne à la négation. Pour définir ces sémantiques, on se concentre sur l'inférence sceptique. Par soucis de simplicité des notations, on utilisera à présent $\mid\!\sim$ pour représenter l'inférence sceptique.
Une notion intéressante est celle d'ensemble de conséquences d'un système d'argumentation. Les conséquences d'un système AF sont tous les arguments α tels que $AF \mid\!\sim \alpha$.

Définition 69 (Conséquences sceptiques d'un système d'argumentation)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation. On définit l'ensemble des conséquences sceptiques de AF comme étant l'ensemble des arguments inférés sceptiquement par AF :

$$Cn(AF) = \{\alpha \in \mathcal{A} | AF \mid\!\sim \alpha\} = \bigcap_{\varepsilon \in Ext(AF)} \varepsilon$$

On parlera dans la suite simplement de conséquences de AF.

On dispose à présent de trois cadres pour associer une sémantique aux formules sur les arguments : les *labellings*, les extensions et les conséquences. Il suffit d'associer un sens aux expressions $AF \mid\!\sim \alpha$ et $AF \mid\!\sim \neg\alpha$ pour ensuite donner la sémantique des formules « complexes ».

Définition 70 (Sémantique associée aux formules sur les arguments dans le cadre des *labellings*)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, soit $Labs$ l'ensemble des *labellings* associés à AF pour une certaine sémantique, et soit $\alpha \in \mathcal{A}$.

- $AF \mid\!\sim_{\mathcal{L}} \alpha$ si et seulement si $\forall \mathcal{L} \in Labs, \mathcal{L}(\alpha) = in$
- $AF \mid\!\sim_{\mathcal{L}} \neg\alpha$ si et seulement si $\forall \mathcal{L} \in Labs, \mathcal{L}(\alpha) = out$

Définition 71 (Sémantique associée aux formules sur les arguments dans le cadre des extensions)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, soit E l'ensemble des extensions de AF pour une certaine sémantique, et soit $\alpha \in \mathcal{A}$.

- $AF \mid\!\sim_{Ext} \alpha$ si et seulement si $\forall \varepsilon \in E, \alpha \in \varepsilon$
- $AF \mid\!\sim_{Ext} \neg\alpha$ si et seulement si $\forall \varepsilon \in E, \alpha \notin \varepsilon$

Définition 72 (Sémantique associée aux formules sur les arguments dans le cadre des conséquences)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation et soit $\alpha \in \mathcal{A}$.
- $AF \mid\!\sim_{Cn} \alpha$ si et seulement si $\alpha \in Cn(AF)$
- $AF \mid\!\sim_{Cn} \neg\alpha$ si et seulement si $\alpha \notin Cn(AF)$

On notera $\mid\!\sim$ lorsqu'il n'y a pas d'ambiguïté possible sur la sémantique chosie, ou lorsqu'une remarque s'applique aux trois sémantiques.

Il est intéressant de remarquer que pour ces trois sémantiques, la condition pour obtenir $AF \mid\!\sim \alpha$ est équivalente, par contre on a trois conditions distinctes pour inférer $\neg\alpha$.

Proposition 74.
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, soit $\alpha \in \mathcal{A}$.
- $AF \mid\!\sim_{\mathcal{L}} \alpha \Leftrightarrow AF \mid\!\sim_{Ext} \alpha \Leftrightarrow AF \mid\!\sim_{Cn} \alpha$.
- $AF \mid\!\sim_{\mathcal{L}} \neg\alpha \Rightarrow AF \mid\!\sim_{Ext} \neg\alpha \Rightarrow AF \mid\!\sim_{Cn} \neg\alpha$*, les implications inverses ne sont pas vraies.*

Démonstration.

- $AF \mid\!\sim_{\mathcal{L}} \alpha \Leftrightarrow AF \mid\!\sim_{Ext} \alpha$ s'explique par la bijection entre les *labellings* et les extensions pour une sémantique donnée. On rappelle qu'on peut construire une extension à partir d'un *reinstatement labelling* en conservant les arguments étiquetés in, et inversement on peut construire un *reinstatement labelling* à partir d'une extension en étiquetant in les arguments de l'extension, puis en « propageant » (les arguments attaqués par ceux de l'extension sont out, les arguments qui attaquent l'extension sont out, les autres sont $undec$).
- $AF \mid\!\sim_{Ext} \alpha \Leftrightarrow AF \mid\!\sim_{Cn} \alpha$ découle de la définition des conséquences. En effet,
$$\begin{aligned} AF \mid\!\sim_{Ext} \alpha &\Leftrightarrow \forall \varepsilon \in Ext(AF), \alpha \in \varepsilon \\ &\Leftrightarrow \alpha \in \bigcap_{\varepsilon \in Ext(AF)} \varepsilon \\ &\Leftrightarrow \alpha \in Cn(AF) \\ &\Leftrightarrow AF \mid\!\sim_{Cn} \alpha \end{aligned}$$
- $AF \mid\!\sim_{\mathcal{L}} \neg\alpha \Rightarrow AF \mid\!\sim_{Ext} \neg\alpha$: de la même façon que précédemment, on rappelle que les arguments in pour un *reinstatement labelling* sont ceux qui sont dans l'extension correspondante. Donc si α est out pour chaque *labelling*, il n'est pas in. Et donc il n'est dans aucune extension. La réciproque n'est pas vraie, car un argument peut n'être dans aucune extension, mais ne pas être out à chaque fois : s'il n'est pas attaqué par chacune des extensions, il pourra être $undec$.
- $AF \mid\!\sim_{Ext} \neg\alpha \Rightarrow AF \mid\!\sim_{Cn} \neg\alpha$: si α n'est dans aucune extension, alors il n'est pas dans l'intersection des extensions. Mais la réciproque n'est pas vraie : il suffit que α soit absent d'une extension pour qu'on puisse inférer $\neg\alpha$ dans le cadre des conséquences, mais cela n'implique pas qu'on infère $\neg\alpha$ dans le cadre des extensions.

□

Quelle que soit la sémantique choisie pour inférer α et $\neg\alpha$, on construit la sémantique d'une formule complexe de la même façon, similaire à ce qui se fait en logique propositionnelle.

Définition 73 (Sémantique associée aux formules sur les arguments)**.**
On définit la satisfaction d'une formule φ sur les arguments par un système d'argumentation AF, notée $AF \mid\!\sim \varphi$, de la façon suivante :
- Si $\varphi = \varphi_1 \wedge \varphi_2$, $AF \mid\!\sim \varphi$ si et seulement si $AF \mid\!\sim \varphi_1$ et $AF \mid\!\sim \varphi_2$.
- Si $\varphi = \varphi_1 \vee \varphi_2$, $AF \mid\!\sim \varphi$ si et seulement si $AF \mid\!\sim \varphi_1$ ou $AF \mid\!\sim \varphi_2$.
- Si $\varphi = \neg(\varphi_1 \wedge \varphi_2)$, $AF \mid\!\sim \varphi$ si et seulement si $AF \mid\!\sim (\neg\varphi_1 \vee \neg\varphi_2)$.
- Si $\varphi = \neg(\varphi_1 \vee \varphi_2)$, $AF \mid\!\sim \varphi$ si et seulement si $AF \mid\!\sim (\neg\varphi_1 \wedge \neg\varphi_2)$.

– Si $\varphi = \neg\neg\varphi_1$, $AF {\mid\!\sim} \varphi$ si et seulement si $AF {\mid\!\sim} \varphi_1$.

On définit alors la notion de cohérence d'une formule sur les arguments.

Définition 74 (Cohérence d'une formule)**.**
Une formule sur les arguments φ est cohérente si et seulement s'il existe un système d'argumentation qui satisfait φ.

Dans la suite de ce mémoire, certains résultats seront présentés avec trois variantes, correspondant à ces trois sémantiques. A terme, il faudra choisir parmi les trois sens de la négation lequel doit être retenu :
- inférer $\neg\alpha$ si α est attaqué par chaque extension (*labellings*) ;
- inférer $\neg\alpha$ si α n'appartient à aucune extension, sans forcément être attaqué par chacune (extensions) ;
- inférer $\neg\alpha$ si α n'appartient pas à au moins une extension.

1.1.2 Formalisation de la révision de systèmes d'argumentation

On dispose maintenant des éléments pour formaliser la notion de révision d'un système d'argumentation. On présente d'abord le cas particulier de la révision par un argument ou sa négation.

Définition 75 (Révision par un argument, par la négation d'un argument)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, et soit $\alpha \in \mathcal{A}$ un argument.
- La révision de AF par α est la transformation du graphe associé à AF afin de produire un ensemble de systèmes d'argumentation qui acceptent sceptiquement α.
- La révision de AF par $\neg\alpha$ est la transformation du graphe associé à AF afin de produire un ensemble de systèmes d'argumentation qui refusent (au sens de la sémantique choisie, parmi les *labellings*, les extensions et les conséquences) α.

En toute généralité, la révision par une formule sur les arguments φ est un changement sur le graphe de façon que les systèmes résultats de ce changement infèrent φ (au sens de la sémantique choisie).

Définition 76 (Révision par une formule sur les arguments)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, et soit φ une formule sur les arguments de $\mathcal{A}$.
La révision de AF par φ est la transformation du graphe associé à AF afin de produire un ensemble de systèmes d'argumentation qui infèrent φ. Formellement, un opérateur de révision de système d'argumentation $\dot{\times}$ est un changement sur le graphe du système tel que

$$AF \mathbin{\dot{\times}} \varphi \subseteq \{AF' | AF' {\mid\!\sim} \varphi\}$$

Cette définition est totalement indépendante de la sémantique choisie pour « $AF {\mid\!\sim} \varphi$ », et permet de choisir par la suite laquelle semble la mieux adaptée.

1.1.3 Partir des *labellings* pour définir la dynamique dans les systèmes d'argumentation

Les *labellings* offrent la possibilité de décrire de façon fine le statut des arguments. On se propose d'utiliser ce cadre pour définir plus précisément la révision, mais aussi les autres opérations de la dynamique : expansion et contraction.

Association d'un AF-*labelling* à un système d'argumentation

L'idée de base est de définir un *labelling* associé non pas à une extension du système d'argumentation, mais au système lui-même, représentant le statut d'un argument selon l'inférence sceptique. En effet, on considère qu'un argument est accepté par l'agent dont les croyances sont modélisées par le système d'argumentation si et seulement si l'argument en question est étiqueté in dans tous les *labellings* associés aux extensions du système.
On définit de la même façon un argument rejeté par l'agent : tous les *labellings* marquent cet argument out. Les arguments pour lesquels l'agent est indécis sont ceux restants.

Définition 77 (*Labelling* sceptique associé à un système d'argumentation)**.**
Soit AF un système d'argumentation et soit une sémantique $\mathcal{S}$, avec $E = \{\varepsilon_1, \ldots, \varepsilon_n\}$ l'ensemble des extensions de AF pour cette sémantique.
On appelle $Labs$ l'ensemble des *labellings* associés aux extensions de AF, c'est-à-dire
$Labs = \{\mathrm{Ext2Lab}(\varepsilon_i) | \varepsilon_i \in E\}$. On définit le *labelling* sceptique associé à AF de la façon suivante :

$$\begin{aligned} Lab = &\{(\alpha, in) | \mathcal{L}(\alpha) = in, \forall \mathcal{L} \in Labs\} \\ &\cup \{(\alpha, out) | \mathcal{L}(\alpha) = out, \forall \mathcal{L} \in Labs\} \\ &\cup \{(\alpha, undec) | \exists \mathcal{L}_1 \in Labs \text{ tel que } \mathcal{L}_1(\alpha) \neq in \\ &\text{et } \exists \mathcal{L}_2 \in Labs \text{ tel que } \mathcal{L}_2(\alpha) \neq out\} \end{aligned}$$

Plus simplement, on associe à α l'étiquette in si $AF \mid\!\sim_{\mathcal{L}} \alpha$, l'étiquette out si $AF \mid\!\sim_{\mathcal{L}} \neg\alpha$, et l'étiquette $undec$ autrement.

A un système d'argumentation correspond un seul *labelling* sceptique, mais la relation entre systèmes d'argumentation et *labellings* sceptiques n'est pas bijective : un même *labelling* peut être associé à plusieurs systèmes d'argumentation différents.

Exemple 13 (Exemple de systèmes d'argumentation différents mais avec le même *labelling* sceptique)**.**

Par exemple, les deux systèmes ci-dessous peuvent être associés au même *labelling* : $\{(a, in)$, (b, out), $(c, in)\}$ alors qu'ils sont évidemment différents.

$$a \longrightarrow b \longrightarrow c$$

FIGURE 1.1 – a et c sont acceptés car a défend c

$$a \longrightarrow b \longleftarrow c$$

FIGURE 1.2 – Ici a et c ne sont pas attaqués

Cela implique qu'un opérateur de révision défini comme une opération sur les *labellings* ne produira pas un unique système d'argumentation, alors que dans le cas de la logique propositionnelle par exemple, une opération qui définit les modèles de la formule révisée fournit une base révisée unique à l'équivalence logique près.

Définition des opérations par les *labellings*

La notion de *labelling* sceptique associé à un système d'argumentation permet d'obtenir une correspondance entre le statut d'un argument (sceptiquement in, sceptiquement out, ou $undec$) et l'état d'une croyance (acceptée, refusée, indéterminée), et donc de définir ce qu'on attend des opérations de dynamique des croyances :

Expansion Opération qui modifie le graphe d'un système d'argumentation de façon à ce qu'un argument passe de $undec$ à in ou out dans le *labelling* sceptique.

Contraction Opération qui modifie le graphe d'un système d'argumentation de façon à ce qu'un argument passe de out à $undec$ ou de in à $undec$ dans le *labelling* sceptique.

Révision Opération qui modifie le graphe d'un système d'argumentation de façon à ce qu'un argument passe de in à out ou de out à in dans le *labelling* sceptique.

On peut donc reproduire le schéma 2.1 dans le cadre de l'argumentation :

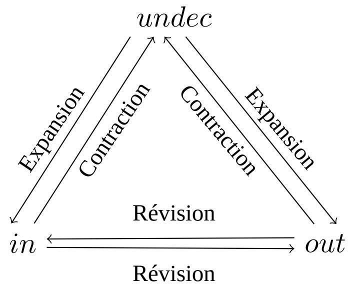

FIGURE 1.3 – Transitions entre statuts d'acceptation

On peut à présent définir formellement les trois opérations de la façon suivante.

Définition 78 (Opérateur d'expansion de systèmes d'argumentation)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, Lab le *labelling* sceptique associé à AF, et $\alpha \in \mathcal{A}$.
$\dotplus$ est un opérateur d'expansion de système d'argumentation si

- si $Lab(\alpha) = undec$,
  - $AF \dotplus \alpha$ est un ensemble de systèmes d'argumentations qui acceptent α sceptiquement ($\forall AF' \in AF \dotplus \alpha$, $Lab'(\alpha) = in$ avec Lab' le *labelling* sceptique associé à AF').
  - $AF \dotplus \neg\alpha$ est un ensemble de systèmes d'argumentations qui rejettent α sceptiquement ($\forall AF' \in AF \dotplus \neg\alpha$, $Lab'(\alpha) = out$ avec Lab' le *labelling* sceptique associé à AF').
- si $Lab(\alpha) = in$,
  - $AF \dotplus \alpha = \{AF\}$
  - $AF \dotplus \neg\alpha = \varnothing$
- si $Lab(\alpha) = out$,
  - $AF \dotplus \alpha = \varnothing$
  - $AF \dotplus \neg\alpha = \{AF\}$

Les deux situations pour lesquelles on obtient $\varnothing$ sont les cas pour lesquels l'expansion est incohérente. Cela revient à expanser une formule logique par l alors que la formule de départ infère $\neg l$ (plus formellement, c'est calculer $\varphi + l$ alors que $\varphi \vdash \neg l$).

Définition 79 (Opérateur de contraction de systèmes d'argumentation)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, Lab le *labelling* sceptique associé à AF, et $\alpha \in \mathcal{A}$.
$\dot{-}$ est un opérateur de contraction de système d'argumentation si

- si $Lab(\alpha) = undec$, $AF \dot{-} \alpha = \{AF\}$
- sinon, $AF \dot{-} \alpha$ est un ensemble de systèmes d'argumentations qui ne rejettent ni n'acceptent α sceptiquement ($\forall AF' \in AF \dot{-} \alpha$, $Lab'(\alpha) = undec$ avec Lab' le *labelling* sceptique associé à AF').

Définition 80 (Opérateur de révision de systèmes d'argumentation)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, Lab le *labelling* sceptique associé à AF, et $\alpha \in \mathcal{A}$.
$\dot{\times}$ est un opérateur de révision de système d'argumentation si

- si $Lab(\alpha) = undec$, $AF \dot{\times} \alpha = AF \dotplus \alpha$ et $AF \dot{\times} \neg\alpha = AF \dotplus \neg\alpha$
- si $Lab(\alpha) = in$, $AF \dot{\times} \neg\alpha$ est un ensemble de systèmes d'argumentations qui rejettent α sceptiquement ($\forall AF' \in AF \dot{\times} \neg\alpha$, $Lab'(\alpha) = out$ avec Lab' le *labelling* sceptique associé à AF').
- si $Lab(\alpha) = in$, $AF \dot{\times} \alpha = \{AF\}$
- si $Lab(\alpha) = out$, $AF \dot{\times} \alpha$ est un ensemble de systèmes d'argumentations qui acceptent α sceptiquement ($\forall AF' \in AF \dot{\times} \alpha$, $Lab'(\alpha) = in$ avec Lab' le *labelling* sceptique associé à AF').
- si $Lab(\alpha) = out$, $AF \dot{\times} \neg\alpha = \{AF\}$

1.2 Postulats pour la révision de systèmes d'argumentation

Dans cette section, on construit un ensemble de postulats adaptés à la révision dans le cadre de l'argumentation abstraite. On procède en deux étapes. Après une présentation de postulats exprimés pour une logique quelconque en utilisant des relations ensemblistes sur les ensembles de modèles, on adapte ceux-ci aux notions propres aux systèmes d'argumentation : les *labellings*, les extensions et les conséquences. La deuxième étape sera une critique des postulats adaptés, et la proposition de notions supplémentaires, afin de retenir un bon ensemble de postulats.

1.2.1 Des postulats dans d'autres cadres

[QLB06] a présenté une adaptation des postulats classiques qui permet par exemple de se passer de la négation, et qui exprime les relations entre les formules par des relations ensemblistes entre leurs modèles, qu'on note $\mathrm{Mod}(f)$, où f est la formule dont on considère les modèles.

- **(G1)** $\mathrm{Mod}(\Psi \dotplus M) \subseteq \mathrm{Mod}(M)$
- **(G2)** Si $\mathrm{Mod}(\Psi) \cap \mathrm{Mod}(M) \neq \emptyset$, alors $\mathrm{Mod}(\Psi \dotplus M) = \mathrm{Mod}(\Psi) \cap \mathrm{Mod}(M)$
- **(G3)** Si $\mathrm{Mod}(M) \neq \emptyset$, alors $\mathrm{Mod}(\Psi \dotplus M) \neq \emptyset$
- **(G4)** Si $\mathrm{Mod}(\Psi_1) = \mathrm{Mod}(\Psi_2)$ et $\mathrm{Mod}(M_1) = \mathrm{Mod}(M_2)$,
  alors $\mathrm{Mod}(\Psi_1 \dotplus M_1) = \mathrm{Mod}(\Psi_2 \dotplus M_2)$
- **(G5)** $\mathrm{Mod}(\Psi \dotplus M_1) \cap \mathrm{Mod}(M_2) \subseteq \mathrm{Mod}(\Psi \dotplus (M_1 \cup M_2))$
- **(G6)** Si $\mathrm{Mod}(\Psi \dotplus M_1) \cap \mathrm{Mod}(M_2) \neq \emptyset$,
  alors $\mathrm{Mod}(\Psi \dotplus (M_1 \cup M_2)) \subseteq \mathrm{Mod}(\Psi \dotplus M_1) \cap \mathrm{Mod}(M_2)$

Ces postulats ont été proposés à l'origine pour les logiques de description, mais peuvent être utilisés pour n'importe quel cadre présentant la notion de modèles d'une formule.

1.2.2 Traduction des postulats algébriques

Traduction de (G1) - (G6) vers les extensions

Les postulats **(G1)-(G6)** s'appuient sur la notion de modèle. Si on peut faire la correspondance entre modèles et extensions d'un système d'argumentation, on ne peut pas procéder tout à fait de la même façon pour les modèles de la formule par laquelle on révise. En effet, dans le cadre qui nous intéresse, on révise par une formule sur les arguments, pas par un système d'argumentation. Les « modèles » de la formule φ seront donc tous les sous-ensembles de $\mathcal{A}$ qui « satisfont » φ, c'est-à-dire les ensembles d'arguments qui, s'ils sont une extension pour une certaine sémantique, permettent au système d'inférer φ.

Définition 81 (Ensembles d'arguments correspondants à une formule sur les arguments)**.**
Soit φ une formule construite sur un sous-ensemble de l'ensemble $\mathcal{A}$, A_φ est l'ensemble des sous-ensembles de $\mathcal{A}$ qui peuvent être une extension d'un système $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ tel que $AF \mid\!\sim \varphi$. Formellement,

$$A_\varphi = \bigcup_{AF_i \mid\sim \varphi} Ext(AF_i)$$

On traduit donc les postulats **(G1)-(G6)** dans le formalisme des extensions :
- **(G1E)** $Ext(AF \dot{\times} \varphi) \subseteq A_\varphi$ $\forall S \in Ext(AF \dot{\times} \varphi), \varphi \in S$
- **(G2E)** Si $Ext(AF) \cap A_\varphi \neq \emptyset$ alors $Ext(AF \dot{\times} \varphi) = Ext(AF) \cap A_\varphi$
- **(G3E)** $\exists \varepsilon \in Ext(AF \dot{\times} \varphi)$ tel que $\varepsilon \neq \emptyset$
- **(G4E)** Si $Ext(AF_1) = Ext(AF_2)$ alors $Ext(AF_1 \dot{\times} \varphi) = Ext(AF_2 \dot{\times} \varphi)$
- **(G5E)** $Ext(AF \dot{\times} \varphi) \cap A_\psi \subseteq Ext(AF \dot{\times} \varphi \wedge \psi)$
- **(G6E)** Si $Ext(AF \dot{\times} \varphi) \cap A_\psi \neq \emptyset$ alors $Ext(AF \dot{\times} \varphi \wedge \psi) \subseteq Ext(AF \dot{\times} \varphi) \cap A_\psi$

Traduction de (G1) - (G6) vers les *labellings*

De la même façon que pour les extensions, on a besoin d'une notion de « modèle » d'une formule sur les arguments, exprimé en terme de *labelling*.

Définition 82 (*Labellings* correspondants à une formule sur les arguments)**.**
Soit φ une formule construite sur un sous-ensemble de l'ensemble $\mathcal{A}$, $\mathcal{L}_\varphi$ est l'ensemble des *reinstatement labellings* qui peuvent être associés à un système $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ tel que $AF \mid\!\sim \varphi$. Formellement,

$$\mathcal{L}_\varphi = \bigcup_{AF_i \mid\sim \varphi} Labs(AF_i)$$

Les postulats suivants, exprimés avec les *labellings*, sont donc équivalents à ceux donnés en terme d'extensions :
- **(G1L)** $Labs(AF \dot{\times} \varphi) \subseteq \mathcal{L}_\varphi$ $\forall \mathcal{L} \in Labs(AF \dot{\times} \varphi), \mathcal{L}(\varphi) = in$
- **(G2L)** Si $Labs(AF) \cap \mathcal{L}_\varphi \neq \emptyset$ alors $Labs(AF \dot{\times} \varphi) = Labs(AF) \cap \mathcal{L}_\varphi$
- **(G3L)** $Labs(AF \dot{\times} \varphi) \neq \emptyset$
- **(G4L)** Si $Labs(AF_1) = Labs(AF_2)$ alors $Labs(AF_1 \dot{\times} \varphi) = Labs(AF_2 \dot{\times} \varphi)$
- **(G5L)** $Labs(AF \dot{\times} \varphi) \cap \mathcal{L}_\psi \subseteq Labs(AF \dot{\times} \varphi \wedge \psi)$
- **(G6L)** Si $Labs(AF \dot{\times} \varphi) \cap \mathcal{L}_\psi \neq \emptyset$ alors $Labs(AF \dot{\times} \varphi \wedge \psi) \subseteq Labs(AF \dot{\times} \varphi) \cap \mathcal{L}_\psi$

Traduction de (G1) - (G6) vers les conséquences

Pour traduire les postulats ensemblistes des modèles vers la notion de conséquences des systèmes d'argumentation, on définit le couple Cn_φ pour symboliser les « modèles » de la formule φ. Cn_φ est composé d'une première partie correspondant à l'ensemble des arguments qui doivent faire partie des conséquences d'un système pour qu'il infère φ, et d'une seconde partie qui correspond à l'ensemble des arguments qui doivent ne pas faire partie des conséquences d'un système pour qu'il infère φ.

Définition 83 (Ensembles d'arguments correspondants à une formule sur les arguments)**.**
Soit φ une formule construite sur un sous-ensemble de l'ensemble d'arguments $\mathcal{A}$, Cn_φ est un couple constitué du sous-ensemble d'arguments de $\mathcal{A}$ qui doivent être déductibles de tout AF tel que $AF \mid\!\sim \varphi$, et du sous-ensemble d'arguments de $\mathcal{A}$ qui ne doivent pas être déductibles de tout AF tel que $AF \mid\!\sim \varphi$.

On illustre cette définition sur un exemple.

Exemple 14.
Soient $\alpha, \beta \in \mathcal{A}$ deux arguments, et $\varphi = \alpha \wedge \neg\beta$.
$Cn_\varphi = (\{\alpha\}, \{\beta\})$ car tout système qui satisfait φ doit accepter α et rejeter β. Les arguments de $\mathcal{A}\backslash\{\alpha, \beta\}$ n'ont pas à appartenir à une des composantes de Cn_φ.

Définition 84 (Composantes de Cn_φ)**.**
On note respectivement Req_φ et For_φ les deux composantes de Cn_φ (pour *required* et *forbidden*).

Concrètement, soit α un argument et soient φ, φ_1 et φ_2 des formules construites sur les arguments, alors
- $Cn_\alpha = (\{\alpha\}, \varnothing)$
- $Cn_{\neg\alpha} = (\varnothing, \{\alpha\})$
- $Cn_{\varphi_1 \wedge \varphi_2} = (Req_{\varphi_1} \cup Req_{\varphi_2}, For_{\varphi_1} \cup For_{\varphi_2})$
- $Cn_{\varphi_1 \vee \varphi_2} = (Req_{\varphi_1} \cap Req_{\varphi_2}, For_{\varphi_1} \cap For_{\varphi_2})$
- si $\varphi = \varphi_1 \wedge \varphi_2$, alors $Cn_{\neg\varphi} = Cn_{\neg\varphi_1 \vee \neg\varphi_2}$
- si $\varphi = \varphi_1 \vee \varphi_2$, alors $Cn_{\neg\varphi} = Cn_{\neg\varphi_1 \wedge \neg\varphi_2}$
- si $\varphi = \neg\varphi_1$, alors $Cn_{\neg\varphi} = Cn_{\varphi_1}$.

Proposition 75.
Soit φ une formule sur des arguments. Si φ est cohérente (c'est-à-dire il existe un système d'argumentation AF tel que $AF \mid\!\sim \varphi$) alors $Req_\varphi \cap For_\varphi = \varnothing$.

Démonstration.
S'il existe un système d'argumentation qui satisfait la formule φ, alors à partir de ce système on ne peut pas à la fois accepter et refuser un argument α, donc l'intersection de Req_φ et For_φ est vide.
□

La traduction se fait de la façon suivante :
- **(G1C)** $Req_\varphi \subseteq Cn(AF \dot{\times} \varphi)$ et $For_\varphi \subseteq \mathcal{A}\backslash Cn(AF \dot{\times} \varphi)$
- **(G2C)** Si $Req_\varphi \subseteq Cn(AF)$ et $For_\varphi \subseteq \mathcal{A}\backslash Cn(AF)$ alors $AF \dot{\times} \varphi = AF$
- **(G3C)** Si φ est cohérente, $Cn(AF \dot{\times} \varphi) \neq \varnothing$
- **(G4C)** Si $Cn(AF_1) = Cn(AF_2)$ alors $Cn(AF_1 \dot{\times} \alpha) = Cn(AF_2 \dot{\times} \alpha)$
- **(G5C)** $Cn((AF \dot{\times} \varphi) + \psi) \subseteq Cn(AF \dot{\times} \varphi \wedge \psi)$
- **(G6C)** Si $Cn((AF \dot{\times} \varphi) + \psi) \neq \varnothing$ alors $Cn(AF \dot{\times} \varphi \wedge \psi) \subseteq Cn((AF \dot{\times} \varphi) + \psi)$

Critique des traductions

On peut émettre un certain nombre de critiques concernant les postulats présentés ci-dessus, issus de « traductions » à partir des postulats pour les logiques de description.

Une première critique, générale, est la traduction du postulat **(G4)** qui ici n'a pas forcément de sens. En effet, deux systèmes équivalents peuvent ne pas avoir du tout la même structure, et par conséquent l'effet de la révision sur ces systèmes peut ne pas être le même.

Ensuite, dans le cas des postulats exprimés à l'aide des conséquences, les deux derniers postulats nécessitent la définition explicite d'une opération d'expansion, ce qui n'est pas un problème en théorie, mais l'est en pratique tant qu'on n'a pas défini explicitement l'opération d'expansion.

1.2.3 Une autre notion intéressante : l'indépendance

Il est possible d'avoir, dans le graphe qui représente un système d'argumentation, des arguments qui ne sont pas connectés entre eux, c'est-à-dire que ces deux arguments ne s'attaquent ni ne se défendent, directement ou indirectement, mais aussi ne sont pas attaqués ou défendus (directement ou indirectement) par un même troisième argument. Dans ce cas, si l'on révise le système d'argumentation par un de ces deux arguments, il n'y a pas de raison de changer d'une quelconque façon le statut de l'autre. Cette idée est liée à celle de Parikh pour la révision AGM ([Par99]).
Pour proposer un postulat allant en ce sens, on définit d'abord formellement la notion d'indépendance entre arguments.

Définition 85 (Arguments indépendants)**.**
Deux arguments α et β dans un système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ sont indépendants si et seulement s'il n'existe dans le graphe correspondant à AF aucune chaîne de α vers β.

On parle de chaîne, au lieu de chemin, dans la définition, car on ne tient pas compte de l'orientation des attaques, comme on le montre sur l'exemple 15.

Exemple 15 (Arguments indépendants)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ avec $\mathcal{A} = \{a, b, c\}$ et $\mathcal{R} = \{(b, a), (b, c)\}$. Dans cet exemple, a et c ne sont pas indépendants car il existe la chaîne $[a, b, c]$.

Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ avec $\mathcal{A} = \{a, b, c, d\}$ et $\mathcal{R} = \{(b, a), (d, c)\}$. Dans cet exemple, a et c sont indépendants car il n'existe pas de chaîne entre eux.

On peut proposer le postulat suivant, sous les trois formalismes considérés :
- Si $\beta \in \bigcap\limits_{e \in Ext(AF)} e$, et α et β sont indépendants, alors $\beta \in \bigcap\limits_{e \in Ext(AF) \ \alpha} e$
- Si $\mathcal{L}(\beta) = in, \forall \mathcal{L} \in Labs(AF)$, et α et β sont indépendants, alors $\mathcal{L}(\beta) = in \forall \mathcal{L} \in Labs(AF \dot{\times} \alpha)$
- Si $\beta \in Cn(AF)$, et α et β sont indépendants, alors $\beta \in Cn(AF \dot{\times} \alpha)$

Bien entendu, ces idées sont valides si on révise par $\neg\alpha$, et si le statut de β n'est pas « accepté ».

1.2.4 Postulats de rationalité

Expression au moyen des *labellings*

On reformule les postulats en prenant en compte les critiques et remarques énoncées précédemment.
- **(AL1)** $Labs(AF \dot{\times} \varphi) \subseteq \mathcal{L}_\varphi$
- **(AL2)** Si $Labs(AF) \cap \mathcal{L}_\varphi \neq \varnothing$ alors $Labs(AF \dot{\times} \varphi) = Labs(AF) \cap \mathcal{L}_\varphi$
- **(AL3)** Si φ est cohérente, $Labs(AF \dot{\times} \varphi) \neq \varnothing$
- **(AL4)** $Labs(AF \dot{\times} \varphi) \cap \mathcal{L}_\psi \subseteq Labs(AF \dot{\times} \varphi \wedge \psi)$
- **(AL5)** Si $Labs(AF \dot{\times} \varphi) \cap \mathcal{L}_\psi \neq \varnothing$ alors $Labs(AF \dot{\times} \varphi \wedge \psi) \subseteq Labs(AF \dot{\times} \varphi) \cap \mathcal{L}_\psi$
- **(AL6)** Si α et β sont indépendants dans AF, alors $\mathcal{L}(\beta) = \mathcal{L}_i(\beta)$, où $\mathcal{L}$ est le *labelling* sceptique associé au système AF, et $\mathcal{L}_i$ est le *labelling* sceptique associé au système AF_i dans la révision de AF par α (ou $\neg\alpha$).

Expression au moyen des extensions

On peut reformuler les postulats **(AL$_i$)** au moyen des extensions.

- **(AE1)** $Ext(AF \dot{\times} \varphi) \subseteq A_\varphi$
- **(AE2)** Si $Ext(AF) \cap A_\varphi \neq \emptyset$ alors $Ext(AF \dot{\times} \varphi) = Ext(AF) \cap A_\varphi$
- **(AE3)** Si φ est cohérente, $Ext(AF \dot{\times} \varphi) \neq \emptyset$
- **(AE4)** $Ext(AF \dot{\times} \varphi) \cap A_\psi \subseteq Ext(AF \dot{\times} \varphi \wedge \psi)$
- **(AE5)** Si $Ext(AF \dot{\times} \varphi) \cap A_\psi \neq \emptyset$ alors $Ext(AF \dot{\times} \varphi \wedge \psi) \subseteq Ext(AF \dot{\times} \varphi) \cap A_\psi$
- **(AE6)** Si α et β sont indépendants dans AF, alors $\beta \in e$ (resp. $\beta \notin e$), $\forall e \in Ext(AF)$ si et seulement si $\beta \in e'$ (resp. $\beta \notin e'$), $\forall e' \in Ext(AF \dot{\times} \alpha)$ (ou $\forall e' \in Ext(AF \dot{\times} \neg\alpha)$.

Expression au moyen des conséquences

Enfin, pour compléter cette partie, on reformule les postulats au moyen du formalisme des conséquences.

- **(AC1)** $Req_\varphi \subseteq Cn(AF \dot{\times} \varphi)$ et $For_\varphi \subseteq \mathcal{A} \backslash Cn(AF \dot{\times} \varphi)$
- **(AC2)** Si $Req_\varphi \subseteq Cn(AF)$ et $For_\varphi \subseteq \mathcal{A} \backslash Cn(AF)$ alors $AF \dot{\times} \varphi = AF$
- **(AC3)** Si φ est cohérente, $Cn(AF \dot{\times} \varphi) \neq \emptyset$
- **(AC4)** $Cn((AF \dot{\times} \varphi) + \psi) \subseteq Cn(AF \dot{\times} \varphi \wedge \psi)$
- **(AC5)** Si $Cn((AF \dot{\times} \varphi) + \psi) \neq \emptyset$ alors $Cn(AF \dot{\times} \varphi \wedge \psi) \subseteq Cn((AF \dot{\times} \varphi) + \psi)$
- **(AC6)** Si α et β sont indépendants dans AF, alors $\beta \in Cn(AF)$ si et seulement si $\beta \in Cn(AF \dot{\times} \alpha)$ (ou $\beta \in Cn(AF \dot{\times} \neg\alpha)$).

Chapitre 2

Propriétés intéressantes concernant la révision

Après avoir défini formellement la révision de systèmes d'argumentation et établi des familles de postulats pour cette opération, on se propose de mettre en évidence un certain nombre de propriétés.

Dans un premier temps, on définit quelques opérateurs « jouets », qui permettent d'avoir l'intuition sur des propriétés et des problèmes liés à la révision de systèmes d'argumentation.
Ensuite, on approfondit cette réflexion en étudiant la correspondance entre plusieurs types d'opérateurs et les postulats définis précédemment.
Enfin, une dernière étude faite dans ce chapitre concerne la notion de minimalité de changement, centrale dans la dynamique des croyances. On montre qu'il existe au moins deux façons de considérer cette notion dans le cadre des systèmes d'argumentation.

Sommaire

2.1 Définition d'opérateurs de révision partiels 62
- 2.1.1 Première idée : α ne doit pas être attaqué 62
- 2.1.2 Deuxième idée : α ne doit pas être attaqué, sauf s'il est défendu 63
- 2.1.3 Révision par systèmes candidats 64
- 2.1.4 Quelques propriétés intéressantes 68

2.2 Relations entre des opérateurs particuliers et les postulats 70
- 2.2.1 Ajout d'attaques 70
- 2.2.2 Retrait d'attaques 71
- 2.2.3 Opérateur drastique 72
- 2.2.4 Opérateurs mixtes 73

2.3 Différentes distances pour différentes minimalités 74
- 2.3.1 Distance sur le graphe 75
- 2.3.2 Distance de conséquence 75
- 2.3.3 Non-équivalence des distances 75

2.1 Définition d'opérateurs de révision partiels

On se propose de comparer plusieurs méthodes pour réviser un système d'argumentation. On parle ici d'opérateurs partiels car ceux-ci ne font que faire accepter un argument, et ne gèrent pas le cas où on veut faire rejeter un argument, encore moins le cas plus complexe des formules sur les arguments.
Ces différentes méthodes permettent de mettre en lumière des propriétés qu'une opération a selon le type de changements qui composent cette opération, mais aussi de mettre en lumière des problèmes à prendre en compte pour définir par la suite des opérateurs plus « concrets », qui satisfont la définition donnée dans la partie prédédente (possibilité de faire accepter un argument refusé, et de faire refuser un argument accepté, puis de travailler avec des formules).

Dans les idées proposées ci dessous, on ne traite pas le cas où α est déjà accepté dans le système AF. Dans ce cas, la révision consiste à laisser AF inchangé.

2.1.1 Première idée : α ne doit pas être attaqué

La première idée est de retirer de la relation d'attaque toute attaque contre l'argument que l'on souhaite accepter. Formellement :

Définition 86 (Retrait d'attaques systématique)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, et $\alpha \in \mathcal{A}$. On définit la révision par retrait systématique de AF par α de la façon suivante :

$$AF \dot{\times} \alpha = \langle \mathcal{A}, \mathcal{R} \backslash \{(\beta, \alpha) | \beta \in \mathcal{A}\} \rangle$$

De cette façon, on garantit que l'argument α appartiendra à chaque extension complète du système révisé, et donc à chaque extension quelle que soit la sémantique $\mathcal{S}$. Par conséquent, l'argument α sera accepté dans le système révisé.

Proposition 76.
Le système d'argumentation résultat de la révision par retrait systématique accepte l'argument α, quelle que soit la sémantique considérée (parmi celles de Dung).

Démonstration. Soit AF' le système résultat de la révision par retrait systématique, et soit E l'ensemble des extensions complètes de ce système.
Soit $\varepsilon \in E$, deux cas sont possibles :

1. $\alpha \in \varepsilon$: on n'a rien à montrer
2. $\alpha \notin \varepsilon$: comme α n'est pas attaqué, on a en particulier « chaque β qui attaque α est attaqué par ε », c'est-à-dire α est acceptable pour ε. Par définition des extensions complètes de Dung, $\alpha \in \varepsilon$ ou ε n'est pas une extension. C'est une contradiction.

Ce raisonnement est vrai $\forall \varepsilon \in E$, donc α est accepté dans le système révisé. □

Problème : Ce genre d'opérateur de révision ne garantit pas le respect de propriétés intéressantes, comme la minimalité du changement, qu'on définit plus tard dans ce chapitre.

On explique l'intuition liée à ce problème sur un exemple.

Exemple 16.
Voici un exemple de révision d'un système AF par l'opérateur précédemment défini (AF'), et un autre système (AF'') qu'on peut considérer comme révisé à partir de AF (par un autre opérateur), mais avec un changement « plus minimal » que pour obtenir AF', c'est-à-dire plus proche du système de départ.

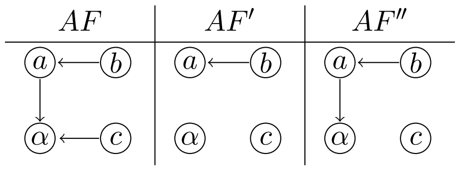

FIGURE 2.1 – Exemple de révision non minimale

Dans le système de départ, la seule extension complète est $\{b, c\}$, donc α est rejeté.
Le système révisé par notre opérateur, AF', a pour seule extension complète $\{b, c, \alpha\}$, donc α est effectivement accepté. Cependant, le troisième système, AF'', a aussi pour seule extension complète $\{b, c, \alpha\}$, donc il accepte aussi l'argument α, mais avec une « perte d'information » moins importante, puisqu'il n'a perdu qu'une attaque par rapport à AF, tandis que AF' en a perdu deux.
On souhaiterait donc avoir, dans ce cas particulier, un opérateur de révision dont le comportement permet d'obtenir AF'', et non pas AF'.

2.1.2 Deuxième idée : α ne doit pas être attaqué, sauf s'il est défendu

Comme le laisse penser l'exemple 16, il n'est pas nécessaire de retirer toutes les attaques qui portent sur α. Si β, qui attaque α, est attaqué par un argument δ qui est accepté, alors l'attaque (β, α) peut être conservée. Comme δ est accepté, toute extension de AF défend α contre β. Formellement :

Définition 87 (Retrait d'attaques nécessaire)**.**
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, et $\alpha \in \mathcal{A}$. On définit la révision par retrait nécessaire de AF par α de la façon suivante :

$$AF \dot{\times} \alpha = \langle \mathcal{A}, \mathcal{R}' \rangle$$

avec $\mathcal{R}' = \mathcal{R} \backslash \{(\beta, \alpha) | \beta \in \mathcal{A} \text{ et } \nexists \delta \in \bigcap_{\varepsilon \in E} \varepsilon \text{ tel que } (\delta, \beta) \in \mathcal{R}\}$

Ainsi, on obtient un résultat similaire à la première idée, mais en apportant moins de changements « inutiles » que précédemment. Bien entendu, dans le cas où $\bigcap_{\varepsilon \in E} \varepsilon = \varnothing$, cet opérateur est identique au premier.

Cet opérateur a permis de mettre en lumière un problème intéressant : celui des cycles de longueur impaire. En effet, dans le cas où l'argument qu'on veut accepter attaque un de ses défenseurs (ou un défenseur indirect), on obtient comme résultat de la révision un système qui n'accepte pas α. Voici un exemple concret.

Exemple 17 (Exemple d'un argument attaquant son défenseur)**.**
Dans le cas où on souhaite accepter l'argument α dans le système représenté par le graphe ci-dessous, on ne retire pas l'attaque de γ vers α car elle est défendue par β, qui est un argument sceptiquement accepté dans le système de départ.

Attaques	$(\alpha;\beta)(\beta;\gamma)(\gamma;\alpha)(\delta;\alpha)$
Extensions	$\{\{\beta, \delta\}\}$
Arguments acceptés	$\{\beta, \delta\}$

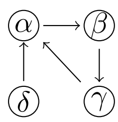

FIGURE 2.2 – Système d'argumentation à réviser par α

Le non-retrait de cette attaque conduit à la situation suivante : β n'est plus accepté, et α n'est pas accepté car il n'est pas défendu par un argument accepté.

Attaques	$(\alpha;\beta)(\beta;\gamma)(\gamma;\alpha)$
Extensions	$\{\{\delta\}\}$
Arguments acceptés	$\{\delta\}$

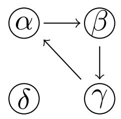

FIGURE 2.3 – Echec de la révision à cause d'un cycle de longueur impaire

La correction de ce problème est une tâche très complexe, elle nécessite en effet de détecter les cycles qui passent par α et ses défenseurs, pour soit retirer l'attaque de α vers son défenseur, soit retirer l'attaque vers α (comme s'il n'était pas défendu).

Globalement, ce type de révision présente un avantage qui peut être intéressant : un seul système d'argumentation est construit, il est donc ensuite aisé pour un agent de travailler avec le nouveau système d'argumentation de la même façon qu'il travaillait avec le système d'origine. De plus, le seul problème du point de vue de la complexité est le calcul des extensions.
Cependant, on souhaite souvent pouvoir effectuer plusieurs révisions successives. Or avec cette méthode, on risque de finir avec un système d'argumentation (presque) sans attaque, ce qui n'est pas forcément souhaitable.

2.1.3 Révision par systèmes candidats

Systèmes candidats

On définit à présent la notion de système d'argumentation candidat pour appartenir à la révision. Il s'agit de systèmes obtenus en modifiant la relation d'attaque pour que α appartienne à toutes les extensions du nouveau système. Par la suite, une fonction de choix sera appliquée pour sélectionner parmi les candidats les systèmes qui sont une révision possible du système de départ.

Tout d'abord, on utilise la fonction $\mathrm{Ext2Lab}$ définie par Caminada, qui associe un *reinstatement-labelling* à une extension du système. On définit $\mathrm{AF2Lab}$ comme étant la fonction qui associe à un système d'argumentation l'ensemble des *labellings* correspondant chacun à une extension du système.

Définition 88.
Soit AF un système d'argumentation, soit E l'ensemble de ses extensions pour une sémantique donnée. On note $\mathrm{AF2Lab}(AF)$ l'ensemble des *reinstatement labellings* correspondant.

$$\mathrm{AF2Lab}(AF) = \{\mathcal{L}_\varepsilon | \mathcal{L}_\varepsilon = \mathrm{Ext2Lab}(\varepsilon), \varepsilon \in E\}$$

On construit un ensemble de systèmes d'argumentation, tels que dans chacun d'eux les β qui attaquent α sont out pour chaque extension du nouveau système. Ainsi, pour chaque système d'argumentation construit, α est in pour chaque *labelling*, c'est-à-dire α est accepté.

Algorithme 1 candidats

ENTRÉES : AF, α

$LLab = \{\text{AF2Lab}(AF)\}$, $LAF = \{AF\}$
pour $\beta \in \mathcal{A}$ tel que $(\beta, \alpha) \in \mathcal{R}$ **faire**
  $n = \text{length}(LAF)$
  pour i de 0 à $n - 1$ **faire**
    $F = \text{retirer\_tete}(LAF)$
    $Lab = \text{retirer\_tete}(LLab)$
    si $\exists \mathcal{L}_\varepsilon \in Lab$ tel que $\mathcal{L}_\varepsilon(\beta) = in$ **alors**
      $\text{traiter}(F, Lab)$ // On va ajouter les fils de F, cad les différentes façons de défendre α
    sinon
      $\text{ajouter\_queue}(LAF, F)$// On remet F dans la liste pour le traiter au niveau suivant
      $\text{ajouter\_queue}(LLab, Lab)$
    fin si
  fin pour
fin pour
retourner LAF

SORTIE : la liste des systèmes candidats pour appartenir à la révision

On peut voir l'algorithme 1 comme la construction d'un arbre, niveau par niveau. La liste LAF correspond à la liste des systèmes d'argumentation intermédiaires construits à un certain niveau dans l'arbre, la racine étant AF.
Clairement, la liste est considérée comme une liste de systèmes à traiter : au niveau de profondeur i dans l'arbre, on souhaite défendre α contre β_i. Pour cela, on construit les systèmes « fils » du système en tête de file, c'est-à-dire toutes les façons possibles de défendre α contre β_i.
Les systèmes d'argumentation candidats pour la révision sont les feuilles de l'arbre.

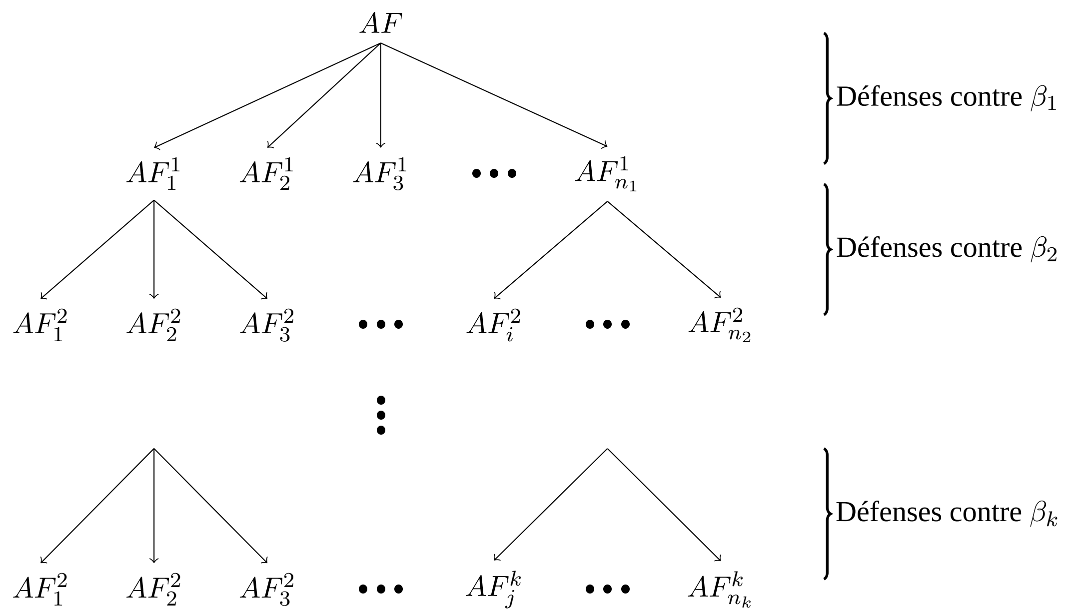

FIGURE 2.4 – Déroulement de l'algorithme 1 sous forme d'arbre

L'algorithme 2 permet de construire chaque fils possible d'un système intermédiaire F, en assurant que pour chaque extension de F chaque façon possible de défendre α est considérée.

Algorithme 2 traiter

ENTRÉES : F, Lab : un système d'argumentation et les *labellings* correspondants

1: $Liste\_AF = \{F\}$
2: **pour** $\mathcal{L}_\varepsilon \in Lab$ tel que $\mathcal{L}_\varepsilon(\beta) = in$ **faire**
3:   $n = \text{length}(Liste\_AF)$
4:   **pour** i de 0 à $n-1$ **faire**
5:     $F' = \text{retirer\_tete}(Liste\_AF)$
6:     **pour** γ tel que $(\gamma, \alpha) \notin \mathcal{R}$ et $\mathcal{L}_\varepsilon(\gamma) = in$ **faire**
7:       $\text{ajouter\_queue}(LAF, \langle \mathcal{A}, \mathcal{R}_{F'} \cup \{(\gamma, \beta)\} \rangle)$
8:     **fin pour**
9:   **fin pour**
10: **fin pour**
11: **pour** $F' \in Liste\_AF$ **faire**
12:   $\text{ajouter\_queue}(LAF, F')$
13:   $\text{ajouter\_queue}(LLab, \text{AF2Lab}(F'))$
14: **fin pour**

Problème : S'il existe une extension ε telle que $(\beta, \alpha) \in \mathcal{R}$, ou $(\alpha, \beta) \in \mathcal{R}, \forall \beta \in \varepsilon$, alors le comportement de l'algorithme n'est pas satisfaisant.
En effet, si c'est le cas, on ne peut ajouter aucun moyen de défendre α comme prévu à la ligne 7 de l'algorithme 2.

Une meilleure construction des systèmes candidats

Pour corriger le problème de l'algorithme de construction de systèmes candidats, on propose une version améliorée de cet algorithme : l'algorithme 3.

On introduit d'abord la notion de conflit unanime. Un argument et un ensemble d'arguments sont en conflit unanime s'il y a une attaque entre l'argument et chaque argument de l'ensemble (dans un sens ou dans l'autre).

Définition 89 (Conflit unanime)**.**
Soient $\alpha \in \mathcal{A}$ et $S \subseteq \mathcal{A}$. On dit que S est en conflit unanime avec α si et seulement si $\forall \beta \in S, (\beta, \alpha) \in \mathcal{R}$ ou $(\alpha, \beta) \in \mathcal{R}$.

On note alors Ext_U l'ensemble des extensions qui sont en conflit unanime avec l'argument α par lequel on révise le système. Ce sont ces extensions qui sont à l'origine du problème posé pour l'opérateur précédemment défini.
L'idée de l'algorithme proposé est maintenant, pour l'ensemble des extensions de Ext_U, de choisir pour chacune un argument β_i de façon à retirer l'attaque (β_i, α) de $\mathcal{R}$ (ou l'attaque (α, β_i)). De cette façon, l'algorithme peut ensuite construire les systèmes candidats, quel que soit le système de départ et quel que soit l'argument par lequel on révise.

Algorithme 3 candidats2

ENTRÉES : AF, α

$LAF = \{AF\}$
$Ext_U = \{\varepsilon \in E | \forall \beta \in \varepsilon, (\beta, \alpha) \in \mathcal{R}\}$
pour $\varepsilon \in Ext_U$ **faire**
    $n = length(LAF)$
    pour i de 0 à $n-1$ **faire**
        $F = \text{retirer\_tete}(LAF)$
        pour $\beta \in \varepsilon$ **faire**
            $\text{ajouter\_queue}(LAF, \langle \mathcal{A}, \mathcal{R}_F \backslash \{(\beta, \alpha)\} \rangle)$
        fin pour
    fin pour
fin pour
$LLab = \{\text{AF2Lab}(F) | F \in LAF\}$
pour $\beta \in \mathcal{A}$ tel que $(\beta, \alpha) \in \mathcal{R}$ **faire**
    $n = \text{length}(LAF)$
    pour i de 0 à $n-1$ **faire**
        $F = \text{retirer\_tete}(LAF)$
        $Lab = \text{retirer\_tete}(LLab)$
        si $\exists \mathcal{L}_\varepsilon \in Lab$ tel que $\mathcal{L}_\varepsilon(\beta) = in$ **alors**
            $\text{traiter}(F, Lab)$ // On va ajouter les fils de F, cad les différentes façons de défendre α
        sinon
            $\text{ajouter\_queue}(LAF, F)$// On remet F dans la liste pour le traiter au niveau suivant
            $\text{ajouter\_queue}(LLab, Lab)$
        fin si
    fin pour
fin pour
retourner LAF

SORTIE : la liste des systèmes candidats pour appartenir à la révision

On note deux problèmes avec cet algorithme :

- Comme pour l'opérateur par retraits nécessaires d'attaques, le cas où l'argument α attaque un de ses défenseurs est problématique.
- L'algorithme peut donner comme résultat des systèmes qui n'ont pas d'extension non vide. Pour illustrer ceci, voir l'exemple 18.

Exemple 18 (Exemple de révision qui conduit à des systèmes sans extension non vide)**.**
On révise le système présenté par la figure suivante, en voulant accepter l'argument α.

Attaques	$(\alpha;\beta)(\beta;\gamma)(\gamma;\zeta)(\delta;\beta)(\delta;\zeta)(\delta;\eta)(\zeta;\delta)$ $(\eta;\alpha)(\eta;\zeta)$
Extensions	$\{\{\alpha,\gamma,\delta\}\}$
Arguments acceptés	$\{\alpha,\gamma,\delta\}$

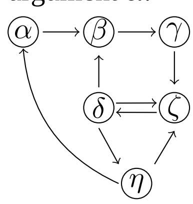

FIGURE 2.5 – Système d'argumentation à réviser par α

Un des systèmes candidats construits est présenté ci-dessous.

Attaques	$(\alpha;\delta)(\beta;\gamma)(\gamma;\zeta)(\delta;\beta)(\delta;\zeta)(\delta;\eta)(\zeta;\delta)$ $(\eta;\alpha)(\eta;\zeta)$
Extensions	$\{\varnothing\}$
Arguments acceptés	$\varnothing$

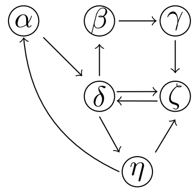

FIGURE 2.6 – Un système « candidat » qui n'a que l'extension $\varnothing$

On retrouve ici les problèmes de cycles : $\alpha \to \delta \to \eta \to \alpha$ par exemple.

2.1.4 Quelques propriétés intéressantes

Les opérateurs partiels définis ci-dessus, malgré leurs défauts évidents, nous ont permis d'arriver à quelques propriétés intéressantes. On suppose, pour chacune de ces propriétés, que le graphe de départ, et le graphe résultat, sont sans boucle.

Pour parvenir à réviser un système d'argumentation (qu'il s'agisse de faire accepter un argument, de faire refuser un argument, ou plus généralement de satisfaire une formule), on peut s'autoriser plusieurs types de changements :
- ne faire que des retraits d'attaques ;
- ne faire que des ajouts d'attaques ;
- retirer et ajouter des attaques.

Selon ce que l'on s'autorise, il est possible qu'un des changements de la dynamique de croyances ne soit pas réalisable, ou au contraire un certain changement de statut peut être garanti par un certain type de changement sur le graphe.

Un première constatation, issue de la définition de l'opérateur de révision par retraits systématiques, est qu'on peut faire accepter un argument avec un opérateur qui n'effectue que des retraits d'attaques.

Proposition 77.
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, et $\alpha \in \mathcal{A}$. On peut assurer que $Lab(\alpha) = in$, où Lab est le labelling sceptique associé à AF, juste en retirant des attaques de la relation $\mathcal{R}$.

Démonstration. Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, et $\alpha \in \mathcal{A}$.
L'application de l'opérateur de révision par retraits systématiques défini précédemment peut être vue comme une suite de retraits d'attaques de la relation $\mathcal{R}$. Une fois toutes les attaques contre α retirées de $\mathcal{R}$, $\forall \mathcal{L} \in \mathrm{AF2Lab}(AF), \mathcal{L}(\alpha) = in$ puisque α n'est pas attaqué. Cette suite de retraits garantit donc que $Lab(\alpha) = in$, avec Lab le *labelling* sceptique associé à AF. □

Cependant, ce genre d'opérateur ne peut pas assurer de faire passer le statut d'un argument à $undec$ ou à out.

Proposition 78.
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, et $\alpha \in \mathcal{A}$. On ne peut pas assurer que $Lab(\alpha) = out$ ou que $Lab(\alpha) = undec$, où Lab est le labelling sceptique associé à AF, juste en retirant des attaques de la relation $\mathcal{R}$.

Démonstration. Dans le système d'argumentation décrit par le graphe de la figure 2.7, il est impossible de changer le statut de l'argument α juste en retirant des attaques. En effet, la seule attaque qui peut être retirée est celle de α vers β, α ne sera pas attaqué pour autant, et restera donc in dans le système résultant de ces changements.

$$\textcircled{\alpha} \longrightarrow \textcircled{\beta}$$

FIGURE 2.7 – α ne peut pas changer de statut juste avec des retraits

On ne peut donc pas assurer qu'un argument devienne $undec$ ou out juste avec des retraits d'attaques. □

On s'intéresse donc maintenant à un opérateur qui ne ferait qu'ajouter des attaques. Avec ce genre d'opérateur, on peut garantir qu'un argument est $undec$ dans le système résultant de la transformation, mais on ne peut pas assurer qu'il sera in ou out.

Proposition 79.
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, et $\alpha \in \mathcal{A}$. On ne peut pas assurer que $Lab(\alpha) = in$ ou que $Lab(\alpha) = out$, où Lab est le labelling sceptique associé à AF, juste en ajoutant des attaques de la relation $\mathcal{R}$.

Démonstration. Dans le système d'argumentation décrit par le graphe ci-dessous, il est impossible de changer le statut de l'argument β en in juste en ajoutant des attaques. En effet, si on ajoute juste une attaque (de β vers α ou de β vers γ), β est quand même attaqué par un argument in (celui qui n'est pas attaqué par β). Et si β attaque les deux autres arguments, alors par exemple les extensions préférées du nouveau système sont $\{\beta\}$ et $\{\alpha, \gamma\}$.
L'argument β est donc $undec$, on ne peut pas le rendre in juste en ajoutant des attaques.

$$\textcircled{\alpha} \longrightarrow \textcircled{\beta} \longleftarrow \textcircled{\gamma}$$

FIGURE 2.8 – β ne peut pas être in juste avec des ajouts

De façon similaire, dans le système suivant, comme α attaque tous les autres arguments, en ajoutant des attaques il appartiendra quand même à au moins une extension ($\{\alpha\}$), et ne pourra donc pas être out.

$$\begin{tikzcd} \textcircled{\alpha} \arrow[r] \arrow[d] \arrow[dr] & \textcircled{\beta} \\ \textcircled{\gamma} & \textcircled{\delta} \end{tikzcd}$$

FIGURE 2.9 – α ne peut pas être out juste avec des ajouts

On ne peut donc pas assurer qu'un argument devienne in ou out juste avec des ajouts d'attaques. □

Proposition 80.
Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, et $\alpha \in \mathcal{A}$. On peut assurer que $Lab(\alpha) = undec$ où Lab est le labelling sceptique associé à AF, juste en ajoutant des attaques à la relation $\mathcal{R}$.

Démonstration. Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, et $\alpha \in \mathcal{A}$. Il suffit, pour garantir que α devienne $undec$, d'ajouter des attaques entre chaque argument du système d'argumentation.
Supposons que $\mathcal{A} = \{\alpha, \beta_1, \ldots, \beta_n\}$, alors les extensions préférées de AF', le système résultant de l'ajout d'attaques à AF, seront $\{\{\alpha\}, \{\beta_1\}, \ldots, \{\beta_n\}\}$. Par conséquent, α aura bien un statut $undec$. □

Puisqu'un opérateur de révision, tel qu'on le souhaite, doit pouvoir faire passer un argument in au statut out, et vice-versa, il est évident qu'un opérateur n'effectuant que des retraits, ou que des ajouts, n'est pas suffisant, étant donné les propriétés précédentes.
On va donc étudier les opérateurs effectuant des ajouts et des retraits d'attaques.

2.2 Relations entre des opérateurs particuliers et les postulats

Une fois les postulats définis, on souhaite comparer plusieurs types d'opérateurs de révision de systèmes d'argumentation en répondant à la question « Quels postulats ce type d'opérateur satisfait-il à coup sûr ? » La réponse à cette question peut permettre d'orienter la forme à donner à un opérateur « concret », voire pourquoi pas être une piste vers un théorème de représentation.

On se propose donc, dans ce chapitre, d'étudier trois familles d'opérateurs : ceux qui n'opèrent que par ajout d'attaques à la relation $\mathcal{R}$, ceux qui n'opèrent que par retrait d'attaques, et enfin les opérateurs « mixtes », c'est-à-dire ceux qui combinent ajouts et retraits. Un opérateur mixte particulier est l'opérateur drastique, qui consiste à « oublier » le graphe de départ pour retenir comme résultat de la révision tous les graphes qui permettent de déduire la formule φ par laquelle on révise.
La problématique de cette section se résume en quelques questions.
- Quels postulats peuvent être satisfaits juste en ajoutant des attaques ? Juste en retirant des attaques ? En mixant les deux ?
- Quels postulats peuvent être satisfaits en oubliant les anciennes croyances pour ne retenir que la nouvelle ?
- Que peut-on en déduire ?

2.2.1 Ajout d'attaques

On traite d'abord le cas où l'opérateur $\dot{\times}$ n'autorise que des ajouts d'attaques à la relation $\mathcal{R}$, c'est-à-dire

$$AF \dot{\times} \varphi \subseteq \{AF_i | \forall i, \mathcal{R} \subseteq \mathcal{R}_i\}$$

La question est donc « un opérateur peut-il satisfaire les postulats juste en ajoutant des attaques ? ». On montre dans cette partie qu'il ne peut pas tous les satisfaire, certains cas particuliers de graphes et de formules φ falsifiant les postulats.

Proposition 81.
On peut garantir qu'un opérateur satisfasse les postulats **(AE3)** *et* **(AE4)** *juste en ajoutant des attaques. On ne peut pas garantir qu'un opérateur satisfasse les postulats* **(AE1)** *et* **(AE2)** *juste en ajoutant des attaques.*

Démonstration.
(AE1) $Ext(AF \dot{\times} \varphi) \subseteq A_\varphi$ On montre sur un exemple simple qu'on ne peut pas garantir le succès de la révision juste en ajoutant des attaques. Soit AF le système d'argumentation représenté par le graphe suivant.

$$a \longrightarrow b$$

FIGURE 2.10 – Le système accepte a et rejette b

Il n'y a que deux moyens d'ajouter des attaques à ce système : le laisser tel quel, ou ajouter une attaque de b vers a. Dans les deux cas, b n'est pas accepté par le nouveau système.
Si $\varphi = b$, quel que soit l'opérateur par ajout d'attaques, $AF \dot{\times} \varphi \nsubseteq A_\varphi$.

(AE2) Si $Ext(AF) \cap A_\varphi \neq \emptyset$ **alors** $Ext(AF \dot{\times} \varphi) = Ext(AF) \cap A_\varphi$ On montre sur un exemple simple qu'un opérateur par ajout ne peut pas garantir que le postulat **(AE2)** soit satisfait. Soit AF le système d'argumentation représenté par le graphe suivant.

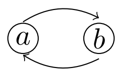

FIGURE 2.11 – Le système n'accepte ni a ni b

Quel que soit l'opérateur par ajout d'attaques choisi, $AF \dot{\times} b = AF$ puisque qu'on ne peut plus ajouter d'attaques.

$Ext(AF, st) = \{\{a\}, \{b\}\}$, donc $Ext(AF, st) \cap A_b = \{\{b\}\} \neq \emptyset$. Cependant, $Ext(AF \dot{\times} b, st) = Ext(AF, st) = \{\{a\}, \{b\}\} \neq Ext(AF, st) \cap A_b$: le postulat n'est pas satisfait.

(AE3) Si φ **est cohérente,** $Ext(AF \dot{\times} \varphi) \neq \emptyset$ En toute généralité, on peut garantir que $Ext(AF \dot{\times} \varphi) \neq \emptyset$ pour les sémantiques complète, de base et préférée. En effet, on rappelle que Dung a montré l'existence d'au moins une extension préférée quel que soit le système d'argumentation AF. Les extensions préférées étant des extensions complètes particulière, on en déduit qu'il existe des extensions complètes, et donc une extension de base (la plus petite extension complète au sens de l'inclusion).
Seule la sémantique stable, parmi les sémantiques de Dung, reste en suspens. Pour celle ci, on peut garantir la satisfaction du postulat juste ajoutant des attaques : si le nouveau graphe est une clique, le système correspondant a pour extensions $E = \{\{\alpha\} | \alpha \in \mathcal{A}\}$, donc le postulat peut être satisfait juste en ajoutant des attaques.

(AE4) $Ext(AF \dot{\times} \varphi) \cap A_\psi \subseteq Ext(AF \dot{\times} \varphi \wedge \psi)$ On peut utiliser la même méthode pour garantir qu'un ajout d'attaques suffise à satisfaire **(AE4)** : en transformant le graphe en clique, $AF \dot{\times} \varphi$ et $AF \dot{\times} \varphi \wedge \psi$ ont le même ensemble d'extensions E, et donc

$$Ext(AF \dot{\times} \varphi) \cap A_\psi = E \cap A_\psi \subseteq E = Ext(AF \dot{\times} \varphi)$$

Le postulat est satisfait.

□

2.2.2 Retrait d'attaques

On étudie à présent ce qui se passe si on ne s'autorise que des retraits d'attaques, c'est-à-dire qu'on choisit un opérateur $\dot{\times}$ tel que

$$AF \dot{\times} \varphi \subseteq \{AF_i | \forall i, \mathcal{R}_i \subseteq \mathcal{R}\}$$

De façon similaire à la partie précédente, on s'intéresse ici à la question « Peut-on garantir de satisfaire les postulats juste en retirant des attaques ? ». La proposition suivante indique clairement que ce n'est pas le cas : certains postulats ne sont pas satisfaits.

Proposition 82.
*On peut garantir qu'un opérateur satisfasse les postulats **(AE3)**, **(AE4)** et **(AE5)** juste en retirant des attaques.*
*On ne peut pas garantir qu'un opérateur satisfasse le postulat **(AE1)** juste en retirant des attaques.*

Démonstration.
(AE1) $Ext(AF \dot{\times} \varphi) \subseteq A_\varphi$ On montre que ce postulat ne peut pas être satisfait dans toutes les situations, par un exemple simple. Soit $AF = \langle \{a, b\}, \emptyset \rangle$, et soit $\varphi = \neg b$.
Si on ne s'autorise que des retraits d'attaque, on a trivialement $AF \dot{\times} \neg b = AF$, puisqu'il n'y a aucune attaque à retirer, et donc $Ext(AF \dot{\times} \neg b) = \{\{a, b\}\} \nsubseteq A_{\neg b}$.

(AE3) Si φ est cohérente, $Ext(AF \dot{\times} \varphi) \neq \emptyset$ On peut procéder de façon similaire à ce qui a été présenté pour le cas des opérateurs de révision par ajout d'attaque : au lieu de former une clique pour construire le nouveau graphe, on laisse le graphe sans attaque, et par conséquent, l'unique extension du nouveau système est $\mathcal{A}$, donc le postulat peut être satisfait.

(AE4) $Ext(AF \dot{\times} \varphi) \cap A_\psi \subseteq Ext(AF \dot{\times} \varphi \wedge \psi)$ En procédant de la même façon que pour le postulat précédent, on peut garantir la satisfaction de celui-ci.
En effet, si la révision consiste à « vider » le graphe de ses attaques, alors $Ext(AF \dot{\times} \varphi) \cap A_\psi$ est égal soit à $\{\mathcal{A}\}$ (si $A_\psi = \{\mathcal{A}\}$), soit à $\emptyset$. Dans les deux cas, le postulat est vérifié, donc on peut bel et bien garantir sa satisfaction juste par des retraits d'attaque.

(AE5) Si $Ext(AF \dot{\times} \varphi) \cap A_\psi \neq \emptyset$**, alors** $Ext(AF \dot{\times} \varphi \wedge \psi) \subseteq Ext(AF \dot{\times} \varphi) \cap A_\psi$ En utilisant le même cas « extrême » que pour les postulats précédents, on rappelle qu'on a $Ext(AF \dot{\times} \varphi) \cap A_\psi$ est égal soit à $\{\mathcal{A}\}$, soit à $\emptyset$. Si cette intersection est non vide, alors $Ext(AF \dot{\times} \varphi \wedge \psi) = \{\mathcal{A}\} \subseteq Ext(AF \dot{\times} \varphi) \cap A_\psi = \{\mathcal{A}\}$: le postulat est vérifié.

□

2.2.3 Opérateur drastique

Le drastique est l'opérateur $\dot{\times}_D$ tel que

$$AF \dot{\times}_D \varphi = \begin{cases} AF & \text{si} \quad AF \mid\!\sim \varphi \\ \{AF_i \mid AF_i \mid\!\sim \varphi\} & \text{sinon} \end{cases}$$

Pour comparer à la révision en logique propositionnelle, cela revient à oublier les anciennes croyances et à conserver uniquement la nouvelle croyance si celle ci n'est pas conséquence de l'ancienne base de croyances. Dans le cas contraire, rien ne change.

Proposition 83.
*L'opérateur drastique satisfait les postulats **(AE1)** et **(AE3)** jusque **(AE5)***
*L'opérateur drastique ne satisfait pas les postulats **(AE2)** et **(AE6)**.*

Démonstration.
(AE1) $Ext(AF \dot{\times} \varphi) \subseteq A_\varphi$ Deux cas sont possibles.
- Si $AF \mid\!\sim \varphi$, alors le postulat est trivialement satisfait.
- Sinon, par définition de l'opérateur drastique, $\forall AF' \in AF \dot{\times}_D \varphi$, $AF' \mid\!\sim \varphi$, donc $Ext(AF') \subseteq A_\varphi$.
  De là, $Ext(AF \dot{\times}_D \varphi) = \bigcup_{AF' \mid\sim \varphi} Ext(AF') \subseteq A_\varphi$.

Le premier postulat est donc satisfait.

(AE2) Si $Ext(AF) \cap A_\varphi \neq \emptyset$ **alors** $Ext(AF \dot{\times} \varphi) = Ext(AF) \cap A_\varphi$ Dans le cas où $AF \not\mid\!\sim \varphi$, on a $Ext(AF \dot{\times}_D \varphi) = A_\varphi$. Or, si $Ext(AF) \cap A_\varphi \neq \emptyset$, et $Ext(AF) \cap A_\varphi \neq A_\varphi$, alors le postulat n'est pas satisfait.
On le montre sur un exemple. Soit $\varphi = b \wedge c$, soit AF le système représenté par le graphe de la figure 2.12.

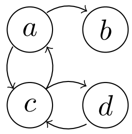

FIGURE 2.12 – Une extension du système satisfait $\varphi = b \wedge c$

Les extensions stables de AF sont $Ext(AF) = \{\{a, d\}, \{b, c\}\}$. De toute évidence, $Ext(AF) \cap A_\varphi = \{\{b, c\}\} \neq \emptyset$. Cependant, $AF \not\mid\!\sim \varphi$, on est donc dans le second cas de la définition de $\dot{\times}_D$: $AF \dot{\times}_D \varphi = \{AF_i | AF_i \mid\!\sim \varphi\}$, et par conséquent $Ext(AF \dot{\times}_D \varphi) = A_\varphi \neq Ext(AF) \cap A_\varphi$.

(AE3) Si φ **est cohérente,** $Ext(AF \dot{\times} \varphi) \neq \emptyset$ Par définition de l'opérateur, le postulat est satisfait. En effet, si φ est cohérente, c'est-à-dire s'il existe un système d'argumentation qui satisfait la formule, alors les extensions de la révision sont (au moins) les extensions de ce système (ainsi que les extensions des éventuels autres systèmes).

(AE4) $Ext(AF \dot{\times} \varphi) \cap A_\psi \subseteq Ext(AF \dot{\times} \varphi \wedge \psi)$ **et (AE5) Si** $Ext(AF \dot{\times} \varphi) \cap A_\psi \neq \emptyset$**, alors** $Ext(AF \dot{\times} \varphi \wedge \psi) \subseteq Ext(AF \dot{\times} \varphi) \cap A_\psi$ On a établi $Ext(AF \dot{\times} \varphi) = A_\varphi$, par conséquent

$$\begin{aligned} Ext(AF \dot{\times} \varphi) \cap A_\psi &= A_\varphi \cap A_\psi \\ &= A_{\varphi \wedge \psi} \\ &= Ext(AF \dot{\times} \varphi \wedge \psi) \end{aligned}$$

Par conséquent les deux postulats sont vérifiés.

(AE6) Si α **et** β **sont indépendants dans** AF**, alors** $\beta \in e, \forall e \in Ext(AF)$ **si et seulement si** $\beta \in e', \forall e' \in Ext(AF \dot{\times} \star\alpha)$**.** On montre sur un exemple que ce postulat n'est pas satisfait.
Si β est accepté dans AF, mais pas α, alors $AF \dot{\times}_D \alpha$ aura pour résultat l'ensemble des systèmes d'argumentations qui acceptent α. Parmi ceux-ci, il est évident que certains n'acceptent pas β (par exemple ceux qui comportent l'attaque (α, β)).
Donc le statut d'acceptation sceptique de β ne peut pas être « accepté ». $\square$

2.2.4 Opérateurs mixtes

Dans cette dernière partie, on s'intéresse aux opérateurs « mixtes », c'est-à-dire dont le résultat est un ensemble de systèmes d'argumentation qui ont été obtenus en s'autorisant des ajouts et des retraits d'attaques par rapport au système de départ. L'opérateur drastique en est un cas particulier.

La question est donc « Peut-on, par une suite quelconque de retraits et d'ajouts d'attaques, contruire un ensemble de systèmes comme résultat qui satisfasse les postulats ? ».

Une première constatation : on peut assurer le succès des postulats **(AE3)**, **(AE4)** et **(AE5)**, puisqu'on peut les satisfaire juste en retirant des attaques, on peut les satisfaire en autorisant retraits et ajouts.
De plus, le drastique étant un opérateur mixte particulier, **(AE1)** peut être satisfait. On montre tout de même que de façon générale, **(AE1)** peut être satisfait.

Proposition 84.
*En s'autorisant des ajouts et des retraits d'attaque, il est possible de satisfaire les postulats **(AE1)** et **(AE3)** jusque **(AE5)**.*

Démonstration.
(AE1) $Ext(AF \dot{\times} \varphi) \subseteq A_\varphi$ Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation, et soit φ une formule sur les arguments. Deux cas se présentent :

- Si φ est incohérente, c'est-à-dire si $\nexists AF'$ un système d'argumentation construit sur $\mathcal{A}$ tel que $AF' \mid\!\sim \varphi$, alors $AF \dot{\times} \varphi = \varnothing$ (aucun système ne peut être construit), et par conséquent $Ext(AF \dot{\times} \varphi) = \varnothing \subseteq A_\varphi$.
- Si φ est cohérente, $\exists E = \{AF_i = \langle \mathcal{A}, \mathcal{R}_i \rangle | AF' \mid\!\sim \varphi\}$, et donc $AF \dot{\times} \varphi \subseteq E$, et par conséquent $Ext(AF \dot{\times} \varphi) \subseteq A_\varphi$, puisque chaque AF_i satisfait φ.
  Or chaque AF_i peut être construit à partir de AF par une suite de retraits et d'ajouts d'attaque (trivialement, tout retirer, puis ajouter les attaques de $\mathcal{R}_i$).

Donc le postulat est bien satisfait. □

On résume dans le tableau 2.1 les postulats qui peuvent être satisfaits par les différents types d'opérateurs décrits.

Postulat	Ajout	Retrait	Drastique	Mixte
(AE1)	✗	✗	✓	✓
(AE2)	✗		✗	
(AE3)	✓	✓	✓	✓
(AE4)	✓	✓	✓	✓
(AE5)		✓	✓	✓
(AE6)			✗	

TABLE 2.1 – Résumé des postulats satisfaits par les différents types d'opérateurs

Il est donc évident que pour parvenir à un résultat, il faut s'autoriser des ajouts et des retraits d'attaques. Le but est de montrer qu'un tel opérateur peut satisfaire tous les postulats.

2.3 Différentes distances pour différentes minimalités

Un des principes les plus importants en révision de croyances est la minimalité du changement. On souhaite que la nouvelle base de croyances de l'agent soit le plus proche possible de ses croyances précédentes. Dans le cas des systèmes d'argumentation, on peut envisager les choses selon au moins deux points de vue. Le premier est de considérer que l'essentiel est le graphe lui-même, et donc la minimalité s'exprime par le fait de rajouter ou retirer le moins d'attaques possible de la relation $\mathcal{R}$. On peut aussi souhaiter que la minimalité porte sur les conséquences du système, c'est-à-dire qu'il y ait le moins de changements possibles dans l'ensemble d'arguments acceptés sceptiquement pour le système.

Quelle que soit la minimalité qu'on souhaite appliquer, pour choisir quels systèmes candidats seront conservés dans la révision, on peut utiliser une notion de distance entre systèmes d'argumentation. On choisira alors les systèmes qui sont les plus proches du système de départ. Cela garantit la minimalité du changement de croyances : parmi toutes les façons possibles de changer ses croyances pour accepter le nouvel argument, l'agent conserve celles qui sont le moins différentes de ses croyances d'origine.

2.3.1 Distance sur le graphe

Pour obtenir la minimalité sur le changement dans le graphe, on définit une distance qui mesure la proximité entre les graphes qui représentent deux systèmes d'argumentation.

Définition 90 (Distance sur le graphe)**.** Soient $AF_1 = \langle \mathcal{A}, \mathcal{R}_1 \rangle$ et $AF_2 = \langle \mathcal{A}, \mathcal{R}_2 \rangle$ deux systèmes d'argumentation construits sur le même ensemble d'arguments $\mathcal{A}$. La distance sur le graphe entre les deux systèmes se définit ainsi :

$$dg(AF_1, AF_2) = |\mathcal{R}_1 \Delta \mathcal{R}_2|$$

Cette distance entre deux systèmes d'argumentation correspond au nombre d'attaques qu'il faut ajouter ou supprimer pour les rendre identiques. On peut définir une notion d'ordre entre systèmes à partir de cette distance.

Définition 91 (Ordre entre systèmes d'argumentation)**.**
On peut utiliser la distance sur le graphe pour ordonner les systèmes d'argumentation de la façon suivante :

$$AF_1 \leqslant^{dg}_{AF} AF_2 \Leftrightarrow dg(AF_1, AF) \leqslant dg(AF_2, AF)$$

2.3.2 Distance de conséquence

On rappelle que les conséquences d'un système d'argumentation sont les arguments qui peuvent être acceptés sceptiquement par ce système, c'est-à-dire qui appartiennent à chaque extension.

On définit la distance de conséquence entre deux systèmes d'argumentation AF et AF' comme le cardinal de la différence symétrique entre les conséquences de AF et les conséquences de AF'.

Définition 92 (Distance de conséquence entre deux systèmes)**.**
Soient AF et AF' deux systèmes d'argumentation. On définit la distance entre AF et AF' par

$$d_{\mid\sim}(AF, AF') = |Cn(AF) \Delta Cn(AF')|$$

De la même façon qu'avec la distance sur le graphe, on peut définir une notion d'ordre entre systèmes en se basant sur la distance définie.

Définition 93 (Ordre entre systèmes d'argumentation)**.**
On définit l'ordre entre systèmes d'argumentation basé sur la distance de conséquence de la façon suivante :

$$AF_1 \leqslant^{\mid\sim}_{AF} AF_2 \Leftrightarrow d_{\mid\sim}(AF, AF_1) \leqslant d_{\mid\sim}(AF, AF_2)$$

2.3.3 Non-équivalence des distances

Contrairement à la distance sur le graphe, la distance de conséquence ne porte pas sur les attaques, mais directement sur les croyances acceptées par l'agent. Avec cette notion, deux graphes peuvent être très différents mais avoir une distance de conséquence très faible, ou inversement deux graphes très proches peuvent avoir des conséquences très différentes.

Proposition 85.
Soient AF et AF' deux systèmes d'argumentation. La minimalité de différence sur les graphes de AF et AF' ne garantit pas la minimalité de différence sur les conséquences de AF et AF'.

Démonstration. On peut démontrer cette propriété par un exemple.
Soient AF et AF' les deux systèmes d'argumentation ci-dessous.

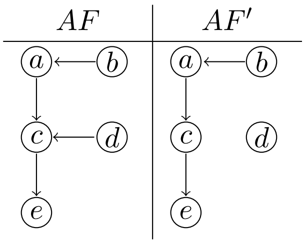

FIGURE 2.13 – Deux systèmes dont le graphe est proche, mais pas les conséquences

Supposons que le but est de réviser le système de façon à rejeter e. On a $dg(AF, AF') = 1$, c'est-à-dire le changement sur le graphe est minimal. Cependant, il n'est pas minimal sur les conséquences.
En effet, l'unique extension de AF est $\{b, d, e\}$, donc $Cn(AF) = \{b, d, e\}$, tandis que l'unique extension de AF' est $\{b, c, d\}$, et donc $Cn(AF') = \{b, c, d\}$. De là, $d_{\mid\!\sim}(AF, AF') = 2$.
Or il est possible d'obtenir un système d'argumentation qui rejette e avec une distance de conséquence par rapport à AF de 1 : il suffit d'ajouter une attaque de d vers e, dans ce cas les arguments acceptés sont b et d, et la distance est strictement plus petite que $d_{\mid\!\sim}(AF, AF')$.
On a donc bien AF' qui est un résultat minimal pour le changement sur le graphe, mais qui n'est pas minimal pour le changement sur les conséquences. □

Proposition 86.
Soient AF et AF' deux systèmes d'argumentation. La minimalité de différence sur les conséquences de AF et AF' ne garantit pas la minimalité de différence sur les graphes de AF et AF'.

Démonstration. De la même façon que pour la propriété précédente, on peut démontrer cette propriété par un exemple.
Soient AF et AF' les deux systèmes d'argumentation ci-dessous.

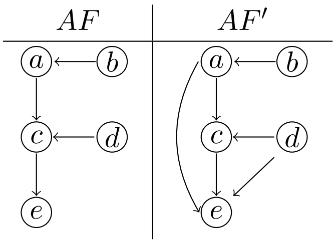

FIGURE 2.14 – Deux systèmes dont les conséquences sont proches, mais pas le graphe

Supposons que le but est de réviser le système de façon à rejeter e. Le changement sur les conséquences est alors minimal, en effet l'unique extension de AF est $\{b, d, e\}$, donc $Cn(AF) = \{b, d, e\}$, tandis que l'unique extension de AF' est $\{b, d\}$, et donc $Cn(AF') = \{b, d\}$. De là, $d_{\mid\!\sim}(AF, AF') = 1$.
On a $dg(AF, AF') = 2$, c'est-à-dire le changement sur le graphe est minimal si et seulement s'il n'existe pas de graphe AF'' tel que $AF'' \mid\!\sim \neg e$ et $dg(AF, AF'') = 1$. Or un tel graphe existe : il suffit d'ajouter une attaque de d vers e, sans ajouter l'attaque de a vers e qui est inutile.
On a donc bien AF' qui est un résultat minimal pour le changement sur les conséquences, mais qui n'est pas minimal pour le changement sur le graphe. □

Il est donc nécessaire, pour définir un opérateur de révision de systèmes d'argumentation, de déterminer quel critère est le plus important. A notre sens, l'information portée par un système d'argumentation est l'ensemble de ses conséquences, il semble donc plus adapté d'utiliser la distance $d_{\mid\!\sim}$ pour filtrer les

candidats non minimaux.
Cependant, pour limiter encore plus le nombre de systèmes dans l'ensemble correspondant à la révision, on peut utiliser la distance sur le graphe comme un second critère.

Un opérateur de révision construit à partir des notions précédentes s'écrit de la façon suivante :

$$AF \dot{\times} \alpha = \min(\min(candidats(AF, \alpha), \leqslant^{\mid\sim}_{AF}), \leqslant^{de}_{AF})$$

Il reste donc à définir quel est l'ensemble $candidats(AF, \alpha)$.
On peut bien entendu définir d'autres ordres, basés sur d'autres types de distance.
Ce type d'opérateur peut rappeler les assignements fidèles proposés par Katsuno et Mendelzon dans le cadre de la révision en logique propositionnelle.

Conclusion et perspectives

Conclusion

Durant ce stage, on s'est intéressé à la question de la révision de systèmes d'argumentation. Si le changement dans les systèmes d'argumentation a déjà été étudié, ces travaux ne correspondent pas à de la révision. Après avoir défini un langage qui permet d'exprimer des informations inférées par un système d'argumentation, on a défini formellement ce qu'est la révision dans le cadre des systèmes d'argumentation, en s'appuyant sur ce langage. Un des buts principaux était de présenter des postulats de rationnalité, puisque les systèmes d'argumentation ne rentrent pas dans le cadre classique AGM. Cette tâche a été effectuée, puisqu'on a donné trois familles, qui correspondent aux trois sémantiques possibles pour le langage sus-cité.

D'autres contributions ont été apportées, avec l'étude de propriétés liées à la révision de systèmes d'argumentation. On a notamment identifié certains postulats dont on peut garantir la satisfaisabilité par un opérateur de révision qui autorise les ajouts et les retraits d'attaques, tandis qu'on a montré qu'il n'est pas possible de satisfaire tous les postulats en n'autorisant que l'ajout ou que le retrait. On a aussi proposé deux sens à la notion de changement minimal dans le cadre de l'argumentation.

Perspectives

Un certain nombre de perspectives de recherche existent. Premièrement, il serait très intéressant de montrer qu'il est possible de satisfaire les postulats manquants, cela confirmerait que nos postulats ont du sens, et que la méthode (changer le graphe, mais ne pas ajouter ou retirer d'arguments) est bonne. Il reste un des postulats de base, ainsi que le postulat supplémentaire qui porte sur l'indépendance. Dans le cas contraire, il serait intéressant de remettre en question notre méthode par un théorème d'impossibilité.

D'autres questions sont ouvertes, notamment quant au choix de la sémantique à choisir parmi les *labellings*, les extensions et les conséquences. Ce choix décidera de la famille de postulats à retenir, ou au contraire découlera du choix d'une famille de postulats. En effet, il est possible qu'une des familles soit « meilleure » que les autres, par exemple si elle peut être satisfaite par un opérateur, mais pas les autres. Une question essentielle sera alors celle d'un théorème de représentation pour associer une classe d'opérateurs aux postulats.
Une autre question intéressante est celle de la minimalité. On a déjà montré que les deux notions proposées ne sont pas équivalentes, il est intéressant de comparer les résultats d'un opérateur concret utilisant une des notions avec les résultats d'un opérateur basé sur l'autre notion. De plus on peut définir d'autres distances pour comparer les systèmes d'argumentation, notamment en utilisant la notion de *labelling*.

On peut de plus envisager plusieurs méthodes pour effectuer une révision de systèmes d'argumen-

tation via d'autres cadres. Par exemple, il serait intéressant de disposer d'un codage logique pour représenter la structure du graphe. Ainsi, on pourrait vérifier si une révision de système d'argumentation peut se faire via une révision en logique (par exemple avec la méthode des assignements fidèles de Katsuno et Mendelzon) suivi d'une « reconstruction » de graphe à partir de la formule révisée.
Un autre moyen de considérer la révision est la plannification. Si on définit l'ensemble des états possibles comme étant l'ensemble des systèmes d'argumentation construits sur l'ensemble d'arguments voulu, et les actions possibles étant « ajouter une attaque » et « retirer une attaque », alors la révision est par exemple le fait de passer en un minimum d'actions de l'état de départ (le système d'argumentation actuel) à un état but (un système qui satisfait la formule par laquelle on révise). Dans ce cas, la minimalité du changement porte sur le graphe, on pourrait définir un problème de plannification équivalent pour le cas où l'on souhaite imposer le changement minimal sur les conséquences.

Enfin, bien entendu, une étude intéressante est celle des autres opérations de dynamique des croyances. Il est intéressant de donner une contrepartie au cadre AGM en présentant des postulats pour l'expansion et la contraction, ainsi que des théorèmes de représentation correspondants. L'étude de ces opérations permettra de vérifier si les identités de Harper et de Levi peuvent être appliquées dans le cadre de l'argumentation abstraite.

Bibliographie

[AGM85] Carlos E. ALCHOURRÓN, Peter GÄRDENFORS, et David MAKINSON. « On The Logic Of Theory Change : Partial Meet Contraction And Revision Functions ». *Journal of Symbolic Logic*, 50 :510–530, 1985. *3 citations pages 1, 20, et 26*

[AM82] Carlos E. ALCHOURRÓN et David MAKINSON. « On the Logic of Theory Change : Contraction Functions and their associated revision functions ». *Theoria*, 48 :14–37, 1982. *Cité page 25*

[BCdSCLS11] Pierre BISQUERT, Claudette CAYROL, Florence Dupin de SAINT-CYR, et Marie-Christine LAGASQUIE-SCHIEX. « Change in Argumentation Systems : Exploring the Interest of Removing an Argument ». Dans Salem BENFERHAT et John GRANT, éditeurs, *SUM*, volume 6929 de *Lecture Notes in Computer Science*, pages 275–288. Springer, 2011. *2 citations pages 31 et 39*

[BG07] Pietro BARONI et Massimiliano GIACOMIN. « On Principle-Based Evaluation of Extension-Based Argumentation Semantics ». *Artificial Intelligence*, 171(10–15) :675–700, 2007. *Cité page 41*

[BKvdT09a] Guido BOELLA, Souhila KACI, et Leendert van der TORRE. « Dynamics in Argumentation with Single Extensions : Abstraction Principles and the Grounded Extension ». Dans Claudio SOSSAI et Gaetano CHEMELLO, éditeurs, *ECSQARU*, volume 5590 de *Lecture Notes in Computer Science*, pages 107–118. Springer, 2009. *2 citations pages 31 et 40*

[BKvdT09b] Guido BOELLA, Souhila KACI, et Leendert van der TORRE. « Dynamics in argumentation with single extensions : attack refinement and the grounded extension ». Dans Carles SIERRA, Cristiano CASTELFRANCHI, Keith S. DECKER, et Jaime Simão SICHMAN, éditeurs, *AAMAS (2)*, pages 1213–1214. IFAAMAS, 2009. *2 citations pages 31 et 45*

[Cam06] Martin CAMINADA. « On the Issue of Reinstatement in Argumentation ». Dans *Journées Européennes sur la Logique en Intelligence Artificielle*. Springer, 2006. *2 citations pages 10 et 11*

[CdSCLS10] Claudette CAYROL, Florence Dupin de SAINT-CYR, et Marie-Christine LAGASQUIE-SCHIEX. « Change in Abstract Argumentation Frameworks : Adding an Argument ». *Journal of Artificial Intelligence Research*, 38 :49–84, 2010. *Cité page 31*

[CMDK+07] Sylvie COSTE-MARQUIS, Caroline DEVRED, Sébastien KONIECZNY, Marie-Christine LAGASQUIE-SCHIEX, et Pierre MARQUIS. « On the merging of Dung's argumentation systems ». *Artificial Intelligence*, 171(10-15) :730–753, octobre 2007. *pas de citations*

[DMT07] Phan Minh DUNG, Paolo MANCARELLA, et Francesca TONI. « Computing Ideal Sceptical Argumentation ». *Artificial Intelligence*, 171(10–15) :642–674, 2007. *Cité page 17*

[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. *5 citations pages 1, 6, 7, 9, et 10*

[Gro88] Adam GROVE. « Two Modellings for Theory Change ». *Journal of Philosophical Logic*, 17 :157–170, 1988. *Cité page 27*

[Gä88] Peter GÄRDENFORS. *Knowledge In Flux*. Cambridge University Press, Cambridge, UK, 1988. *3 citations pages 22, 24, et 26*

[KM91] Hirofumi KATSUNO et Alberto O. MENDELZON. « Propositional Knowledge Base Revision and Minimal Change ». *Artificial Intelligence*, 52 :263–294, 1991. *3 citations pages 1, 28, et 29*

[Moo85] Robert C. MOORE. Semantical Considerations on Nonmonotonic Logic. Dans Matthew L. GINSBERG, éditeur, *Readings in Nonmonotonic Reasoning*, pages 127–136. Morgan Kaufmann, 1985. *Cité page 8*

[OW11] Emilia OIKARINEN et Stefan WOLTRAN. « Characterizing strong equivalence for argumentation frameworks ». *Artificial Intelligence*, 175(14-15) :1985–2009, 2011. *2 citations pages 16 et 17*

[Par99] Rohit PARIKH. « Beliefs, Belief Revision, and Splitting Languages ». Dans Lawrence MOSS, Jonathan GINZBURG, et Maarten de RIJKE, éditeurs, *Logic, Language, and Computation, Vol. 2*, pages 266–278, Stanford, California, 1999. CSLI Publications. *Cité page 58*

[QLB06] Guilin QI, Weiru LIU, et David A. BELL. « Knowledge Base Revision in Description Logics ». Dans Michael FISHER, Wiebe van der HOEK, Boris KONEV, et Alexei LISITSA, éditeurs, *Journées Européennes sur la Logique en Intelligence Artificielle*, volume 4160 de *Lecture Notes in Computer Science*, pages 386–398. Springer, 2006. *Cité page 55*

[Rei80] Raymond REITER. « A Logic for Default Reasoning ». *Artificial Intelligence*, 13 :81–132, 1980. *Cité page 8*

[Tar30] Alfred TARSKI. « Fundamentale Begriffe der Methodologie der Deduktiven Wissenschaften ». *Monatshefte Fur Mathematik*, 37 :361–404, 1930. *Cité page 20*

[Ves11] Srdjan VESIC. « *Preference-based argumentation frameworks : application in decision making and negotiation* ». Thèse de doctorat, Université Paul Sabatier, Toulouse, France, juillet 2011. *2 citations pages 15 et 16*

[vM80] John VON NEUMANN et Oskar MORGENSTERN. *Theory of games and economic behavior*. Princeton University Press, 1980. *Cité page 8*