Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *06/12/2013* **par :**

Pierre Bisquert

Étude du changement en argumentation

De la théorie à la pratique

Directrices de Thèse :

Marie-Christine Lagasquie-Schiex	Florence Dupin de Saint-Cyr
Professeur d'Université	Maître de Conférences
Université Paul Sabatier	Université Paul Sabatier

Rapporteurs :

Nicolas Maudet	Leon van der Torre
Professeur d'Université	Professeur d'Université
Université Pierre et Marie Curie	Université du Luxembourg

Examinatrices :

Claudette Cayrol	Odile Papini
Professeur d'Université	Professeur d'Université
Université Paul Sabatier	Université d'Aix-Marseille

École doctorale et spécialité :

MITT ; Domaine STIC : Intelligence Artificielle

Unité de Recherche :

Institut de Recherche en Informatique de Toulouse (IRIT - UMR 5505)

Résumé

L'argumentation, au sens de l'intelligence artificielle, est un formalisme permettant de raisonner à partir d'informations incomplètes et/ou contradictoires ainsi que de modéliser un échange d'arguments entre plusieurs agents. Un système d'argumentation consiste généralement en un ensemble d'arguments interagissant les uns avec les autres, et duquel il est possible d'extraire un ou plusieurs points de vue cohérents.

Dans cette thèse, nous nous plaçons dans le cadre de l'*argumentation abstraite* dans lequel les arguments sont manipulés en tant qu'entités abstraites dont le sens nous est inconnu et dans lequel les interactions représentent des conflits. Ceci nous permet de nous concentrer sur le point particulier de la dynamique dans les systèmes d'argumentation abstraits, c'est-à-dire les *changements* pouvant impacter ces systèmes, notamment dans le cadre d'un dialogue.

Nous commençons par justifier l'intérêt d'un tel cadre formel puis nous nous intéressons au *comment* et au *pourquoi* du changement en argumentation abstraite. Le *comment* est approché en établissant une liste des modifications que peut subir un système d'argumentation et en étudiant sous quelles conditions elles peuvent survenir. Le *pourquoi* est abordé par l'introduction de la notion de but motivant un changement et le choix du meilleur changement à faire pour satisfaire un but en prenant en considération des contraintes portant sur l'agent à convaincre. Enfin, nous concrétisons notre étude en proposant un outil logiciel implémentant les notions introduites et nous étudions ses performances.

Mots clefs : Argumentation abstraite, dynamique d'un système d'argumentation, changement.

Abstract

Argumentation, in the field of artificial intelligence, is a formalism allowing to reason with incomplete and/or contradictory information as well as to model an exchange of arguments between several agents. An argumentation system usually consists of a set of arguments interacting with each other, and from which it is possible to extract one or several consistent points of view.

In this thesis, we are mainly concerned with the *abstract argumentation* in which arguments are handled as abstract entities whose meaning is unknown and in which the interactions represent conflicts. This allows us to focus on the particular point of the dynamics in abstract argumentation systems, that is to say the *changes* that could impact these systems, particularly in the context of a dialogue.

We start with justifying the interest of such a formal framework, then we study the *how* and the *why* of change in abstract argumentation. The *how* is tackled by establishing a list of changes that an argumentation system can undergo and by studying the conditions under which they may occur. The *why* is addressed by introducing the notion of goal motivating a change and by choosing the best change to make in order to satisfy a goal, taking into account constraints on the agent to convince. Finally, we make our study concrete by proposing a tool that implements the concepts introduced and we study its performance.

Keywords: Abstract argumentation, dynamics of an argumentation system, change.

À ceux que j'aime.

Remerciements

En 900 ans de temps et d'espace, je n'ai jamais rencontré quelqu'un qui ne soit important.

Le Docteur,
Doctor Who.

Un doctorat est bien plus qu'un simple diplôme, bien plus qu'un grade. C'est un apprentissage allant au-delà d'une formation à un métier, et même au-delà d'un choix de carrière. Un doctorat tient plus, pour une personne telle que moi, qui a toujours douté de son avenir, de la plus belle des orientations. Il est difficile de discerner quels sont les éléments en assurant un déroulement fructueux. Néanmoins, un cadre de travail propice et des contacts humains agréables en font, à coup sûr, partie. Ces quelques lignes seront pour moi l'occasion parfaite de remercier les personnes qui, par leurs paroles, leurs actes ou juste leur présence, ont fait de chacun des jours de ces trois dernières années des moments agréables.

Aussi, il est d'une naturelle évidence de destiner mes premiers remerciements à mes directrices de thèses, Marie-Christine Lagasquie-Schiex et Florence Dupin de Saint-Cyr – Bannay, auxquelles j'associe Claudette Cayrol qui, si cela n'était pas officiel, a également veillé à mon avancement. Mes trois encadrantes donc, par leur soutien constant, leurs conseils avisés, leur aide en toute circonstance et leur encadrement délicat mais assuré, m'ont permis d'affronter ces années de travail sereinement. Avoir pu travailler avec elles est sans conteste la plus belle des occasions que m'ait offert cette thèse car il ne saurait être espéré de meilleurs exemples de chercheurs, d'enseignants et, plus que cela, de personnes. J'aurais à jamais un respect insondable à leur égard. La dédicace leur est également adressée.

Mes remerciements vont également aux rapporteurs, Nicolas Maudet et Leon van der Torre, auxquels j'associe Odile Papini, présidente du jury de ma soutenance, pour m'avoir fait l'honneur de prendre le temps de relire en détail mon travail et dont la gentillesse, l'intérêt et les conseils pour continuer et améliorer ma recherche sont inestimables. Que de telles personnes, exemples parfaits de chercheurs accomplis, se soient assis à la table de mon jury est un accomplissement en soi et m'emplit de fierté.

La réussite d'une telle entreprise est également étroitement liée au plaisir que l'on éprouve à s'y engager. Et si le sujet et l'encadrement sont des plus importants, l'ambiance et les liens humains le sont tout autant. À ce titre, je souhaiterais en premier lieu exprimer ma gratitude empreinte d'affection à toute ma famille, dont une partie m'a fait l'honneur d'assister à ma soutenance, Mamie, Christian, Christine, Élise, Jean-Louis, Mila et Serge, ainsi qu'à mes proches, notamment mon parrain, Bernard, et Geneviève. Et tout particulièrement, à mes parents, Rose et Patrick, toujours là lorsque j'en ai eu besoin, jamais avares en mots gentils et à qui je dois tout. La dédicace leur est adressée avant tout autre.

Par ailleurs, si j'ai pris plaisir à venir tous les jours travailler, c'est en très grande partie grâce aux personnes que j'y croisais. Ils sont nombreux et j'en oublierai sans doute ; qu'ils m'en excusent.

Tout d'abord, c'est avec une tendresse certaine que je remercie Armelle. Plus qu'une collègue au contact agréable, plus qu'une amie, elle a été en de nombreux points le mentor que quiconque rêve d'avoir. J'ai pris plaisir à enseigner à ses côtés, et j'en retire une expérience et des souvenirs qui jamais ne me quitteront.

Merci également aux thésards, chercheurs et personnels administratifs, ensuite, que j'ai apprécié côtoyer. En particulier :

- ceux qui m'ont accueilli et servi d'exemples, Srdjan, Manu, Nadine, Marwa et François.
- ceux que j'ai vu arriver et qui m'ont permis de me détendre lors des nombreuses fêtes, soirées et autres “oisivetés”, Audrey, Chloé, Fatma, Faustine, Juliette, Khadra, Raja, Sarah, Alex, Antoine (B.), Antoine (V.), Damien, Fayçal, François, Jean-Philippe, Julien, Matthieu, Newton, Nico, Raphaël, Sif, Simon, Teddy et tant d'autres ; ils ont chacun amené une touche de couleur agréable à mon quotidien.

  Parmi eux, il en est que je remercie tout particulièrement, ceux avec qui j'ai commencé, aux côtés de qui j'ai affronté chaque écueil de ces trois années de doctorat, et pour lesquels j'aurai à jamais une immense affection, Anaïs, Anke et Camille. Ils sont tous trois devenus depuis des amis très chers.

Merci, enfin, à l'ensemble des membres des équipes ADRIA, LILaC et MELODI, et plus généralement de l'IRIT, que j'ai eu l'honneur de connaître.

En guise de conclusion, ayant gardé le meilleur pour la fin, je souhaite remercier ceux que je suis fier de compter parmi mes amis et qui, loins ou proches, sont toujours quelque part dans mes pensées, Fab, Guillaume, Julien, Louis, Louka, Mamat, Nadège, Natha, Paullen, Peter, Samuel, Selma, Titi et Thomas.

À vous tous, la dédicace est adressée.

Table des matières

Chapitre 1

Introduction

Le soulagement est immense de savoir qu'on n'attend pas de vous monts et merveilles.

Agatha Christie,
Une autobiographie.

Sommaire

1.1 Contexte

Intelligence artificielle. Ce terme à lui seul suffit à aviver l'imagination. L'intelligence artificielle est généralement connue à travers l'art par le truchement de la littérature ou du cinéma. Dans ces œuvres, l'intelligence artificielle, à la fois fantasme et cauchemar, est souvent décrite comme au mieux une révolution changeant à jamais la vie des êtres humains et au pire une menace dangereuse pour l'humanité. En définitive, à travers ces fictions, il est difficile de se représenter ce qu'est réellement l'intelligence artificielle. Pour s'en donner une idée, il convient de l'introduire à la lumière de sa discipline mère, l'informatique.

L'informatique, mot-valise formé d'"information" et "automatique", est là encore mal comprise. Bien que son utilisation se soit démocratisée (ou, peut-être, à cause de ce fait), cette science est plus souvent qu'à son tour assimilée à l'étude et à l'utilisation d'un ordinateur. [1] Une célèbre phrase précise à ce sujet que *l'informatique n'est pas plus la science des ordinateurs que l'astronomie n'est celle des télescopes.* [2] Et pour cause, l'informatique est ce qui concerne le *traitement automatique et rationnel de l'information.*

1. Il n'y a ici aucune comparaison de valeur. L'auteur de ces lignes, bien en difficulté face au fonctionnement du cœur d'un ordinateur, pointe seulement du doigt une confusion courante.

2. Cette phrase est communément attribuée à Edsger Dijkstra, célèbre informaticien, mais serait en fait due à Fellows et Parberry (1993).

Comme son nom et sa définition l'indiquent, l'informatique traite des informations de manière automatique. S'il est possible de s'entendre aisément sur le sens d'automatique (c'est-à-dire, sans intervention d'un humain), la notion d'information est plus difficile à se représenter. Dans ce contexte, cette notion est à prendre au sens large et constitue toute donnée (élément "brut" décrivant une certaine réalité) à laquelle on associe du sens, c'est-à-dire que l'on interprète au regard d'un contexte particulier. Une température, un poids, une phrase, une image, un son sont autant d'exemples d'informations. L'exemple d'un sondage, dans lequel les données que l'on récupère sur le terrain sont interprétées de façon à tirer des informations, montre cette différence entre donnée et information.

Pour traiter ces informations, l'informatique use d'algorithmes, des séries d'instructions amenant au résultat souhaité. Notons que si ce terme est fortement associé à cette science, ce qu'il représente existe depuis des millénaires. Pour s'en convaincre, il suffit de penser aux recettes de cuisine. Une recette de cuisine, tout comme un algorithme, a besoin d'une entrée (ingrédient, ustensiles, etc.) et donne une procédure à suivre (mélanger des ingrédients, faire cuire au four, etc.) permettant d'arriver au résultat voulu (un bon gâteau).

Nous n'irons pas plus loin dans la présentation de l'informatique ; il serait en effet bien trop long d'explorer ses sous-domaines. Pour ne pas les nommer, on compte entre autres l'algorithmique (découvrir de nouveaux algorithmes pour résoudre un problème), la calculabilité (savoir s'il existe un algorithme permettant de résoudre un problème), la complexité (calculer les ressources nécessaires, en temps et en espace, d'un algorithme pour résoudre un problème), etc. [3]

Intelligence artificielle

Il y a longtemps que l'Homme est attiré par la vie artificielle et ses bénéfices. Déjà, dans l'Antiquité, Homère relatait dans son *Iliade* (Homère (1995)) les créations d'Héphaïstos capables de l'aider dans ses travaux. Si elle a depuis alimenté une part des cauchemars de l'humanité, l'intelligence artificielle se situe dans la filiation de cette aspiration. Du fait de sa jeunesse, il en existe plusieurs définitions, plus ou moins antagonistes suivant que l'on prend comme modèle l'intelligence humaine ou une intelligence idéale (Russell et Norvig (2002)). Pour les résumer, nous nous contenterons de citer cette phrase attribuée à Minsky, décrivant l'intelligence artificielle comme :

> "*La construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut*

3. Notons que ces sous-domaines appartiennent à la branche théorique de l'informatique ; la branche appliquée contient, entre autres, le génie logiciel (concernant les pratiques du développement de programmes), l'étude des réseaux (Internet, notamment) ou encore tout ce qui a trait à la visualisation et au graphisme.

> *niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique.*"

Le but de l'intelligence artificielle est donc de créer des programmes intelligents, et pour le réaliser, plusieurs axes de recherche existent. Allant de l'apprentissage (permettant à un programme de déduire de nouvelles informations à partir de ses expériences) au multi-agents (étudiant les capacités de communication et de coordination entre programmes) en passant par le traitement automatique de la langue (donnant la possibilité à un programme d'extraire des informations du langage humain), chacun de ces axes étudie et reproduit le fonctionnement d'un pan de l'intelligence humaine.

Deux sujets critiques de l'intelligence artificielle sont le raisonnement et l'interaction. Le premier de ces points concerne le fait de donner à un programme la capacité de raisonner à propos d'un problème pour tenter de trouver de lui-même la meilleure solution. Le second désigne tout ce qui a trait aux échanges entre différents agents (humains ou programmes). L'argumentation, approche que nous adoptons dans ce travail, se trouve au carrefour de ces deux sujets.

Argumentation

L'argumentation est courante dans la vie de tous les jours. Quotidiennement nous énonçons, nous partageons et nous confrontons nos avis. Nous utilisons des arguments pour défendre une idée, des contre-arguments pour en attaquer une autre, etc. Un simple, mais néanmoins typique, exemple pourrait être le suivant :

Exemple 1.1.

Georges	*(x_1) "Ce restaurant est cher, la nourriture doit donc y être de bonne qualité."*
Rosa	*(x_2) "Jean, qui est un fin gourmet, m'a dit le contraire."*
Georges	*(x_3) "Jean ne connaît rien à ce type de cuisine."*

Ces phrases, qui peuvent apparaître comme correctes ou fallacieuses, apportent des arguments destinés à agir sur l'acceptation de l'affirmation de base, c'est-à-dire que "la nourriture doit y être exquise".

Ainsi, l'argumentation, en tant que processus, a pour effet d'augmenter ou de diminuer l'acceptation d'un point de vue, généralement dans le but de convaincre quelqu'un n'étant pas du même avis. On attribue à Perelman et Olbrechts-Tyteca, dans leur *Traité de l'argumentation* (Perelman et Olbrechts-Tyteca (1958)), la définition de l'argumentation comme étant *la manière de présenter et de disposer des arguments (raisonnements ou raisons avancées n'ayant pas valeur de preuve mais qui s'imposent à tout être raisonnable) à l'appui d'une thèse ou contre celle-ci, en vue d'obtenir l'adhésion par consentement d'un auditoire.* Nous laissons au lecteur le plaisir d'approfondir une telle lecture. Notons du reste que l'argumentation ne requiert pas nécessairement la présence d'une autre personne pour exister. Elle peut également être menée avec soi-même ; elle est dans ce cas un parfait moyen pour raisonner, pour mettre en perspective plusieurs informations et prendre une décision en pondérant les divers choix s'offrant à nous.

En étant à tel point au cœur du raisonnement et des interactions humaines, l'argumentation ne saurait échapper au domaine de l'informatique et plus particulièrement à celui de l'intelligence artificielle, tant et si bien que plusieurs approches ont vu le jour pour tenter de la définir formellement. Mais dans quel but ? Pourquoi souhaiter donner la capacité d'argumenter à un programme ? Sans doute pour nous permettre d'interagir plus intuitivement avec lui, pour lui donner la possibilité de nous convaincre de ses choix et de ses conseils avec des arguments ayant un sens pour nous. Plus que cela, en lui donnant la possibilité de confronter et d'analyser nos argumentations, il pourrait nous aider à mieux argumenter entre nous.

Quoi qu'il en soit, examiner le processus d'argumentation dans le cadre de l'intelligence artificielle met en jeu d'autres notions éminentes de ce domaine, en particulier celles de changement et de but. Ces deux notions constituent les sources principales de notre problématique.

1.2 Problématique

Changement

Le changement est, explicitement ou implicitement, un objet d'étude primordial de l'intelligence artificielle. Que ce soit au niveau de ce que sait un programme sur le monde, au niveau de ce qu'il compte y faire ou encore au niveau de ce qui peut changer dans ce monde au cours du temps, la notion de changement est particulièrement importante si l'on veut rendre ce programme capable de s'adapter à ce qui l'entoure et de décider de la marche à suivre.

Cette notion a été étudiée sous différents angles dans le cadre de l'intelligence artificielle. Deux disciplines classiques traitant le changement sont la planification et les théories du changement de croyances.

La planification vise à établir une stratégie consistant en une série d'actes à effectuer par le programme pour résoudre une tâche. Ce dernier doit donc organiser sa course d'actions, c'est-à-dire les changements qu'il doit effectuer sur le monde ; choisir et parcourir un chemin pour se retrouver à un endroit précis est un exemple de changement provoqué par une suite de déplacements.

Les théories du changement de croyances sont les processus permettant à un programme de prendre en compte une nouvelle information en changeant ses croyances sur le monde qui l'entoure ; un programme ayant par exemple pour croyance qu'il fait beau et recevant une information lui indiquant qu'il pleut doit modifier ses croyances pour s'adapter au changement du monde.

L'argumentation, lorsqu'elle concerne des échanges d'arguments entre plusieurs personnes, est elle aussi empreinte de changement. Ces échanges ne sont pas anodins car ils impliquent le changement d'un monde ; de notre monde interne. À chaque argument prononcé par un autre, que l'on soit convaincu ou non, notre vision du monde change. La force d'un argument est telle qu'elle nous change, au moins dans ce que l'on sait. Il semble donc important d'analyser le lien qui existe entre argumentation et changement. Les questions sous-jacentes sont alors "*Dans quelle mesure un argument modifie-t-il les connaissances ? Quel est son impact ?*" et *Pourquoi souhaitons-nous ces changements ?* ; cette dernière constitue la seconde base de notre problématique.

But

De par sa nature, l'argumentation appelle une autre question plus générale. *Pourquoi ?* Plus précisément, que cherchons-nous lorsque nous nous lançons dans une argumentation ?

A priori, la raison la plus probable est que nous espérons convaincre celui ou celle à qui l'on s'adresse et faire en sorte que sa représentation du monde corresponde à ce que l'on souhaite. Ainsi, si nous argumentons avec quelqu'un, c'est parce que nous avons un objectif ; lui faire accepter un argument par exemple. Ceci montre que l'argumentation, en tant qu'interaction entre plusieurs personnes, est intrinsèquement liée à la notion de *but*.

Tout comme le changement, la notion de but est un thème classique de l'intelligence artificielle. C'est elle qui "lance le mouvement", qui pousse au changement. Dans le cas de la planification, le but se trouve être la tâche à résoudre. Dans le cas de l'argumentation, le but d'une personne est donc ce qui le pousse à argumenter. C'est également ce qui lui fait choisir un ou plusieurs arguments à avancer. Pour convaincre quelqu'un, il faut être capable de faire un choix quant à l'argument adéquat, celui qui, on l'espère, permettra de faire accepter notre point de vue par l'autre. Ce choix dépend du but, puisqu'on ne dira pas la même chose suivant ce que l'on souhaite.

Il est donc naturel de lier argumentation, changement et but. Ces notions participent d'ailleurs à un seul et même processus argumentatif. Il est possible de le résumer ainsi :

- Souhaiter d'une personne qu'elle ait une représentation du monde particulière (qu'elle accepte un argument, par exemple), ce qui est représenté par un but.
- Décider d'un argumentaire pour la convaincre, ce qui correspond au choix d'un ou plusieurs arguments satisfaisant ce but.
- Avancer les arguments pour amener cette personne à accepter le point de vue, ce qui engendre des changements dans les connaissances de cette dernière.

Ce processus constitue notre problématique principale et le moteur de nos recherches. Les différentes étapes sont traitées, pas nécessairement dans cet ordre, au fil des pages de ce manuscrit. De façon à présenter la structure de notre travail, nous décrivons en quelques mots les chapitres qui la composent et qui étudient ce processus.

1.3 Guide de lecture

Notre travail se découpe en trois parties principales, elles-mêmes constituées de huit chapitres.

La **première partie** rappelle les bases sur lesquelles nous construisons notre travail et tente de leur donner un caractère concret. Elle est constituée des chapitres 2 et 3.

En particulier, le **chapitre 2** présente le cadre de Dung, formalisme fondateur de l'argumentation abstraite, auquel nous empruntons les notions de *système d'argumentation*, contenant les arguments et leurs interactions, de *sémantique*, permettant de calculer des ensembles d'arguments acceptables (les *extensions*), et de *statut*, indiquant si un argument est accepté ou non. Nous présentons également le travail servant de fondation à notre étude du changement en argumentation. Celui-ci propose une définition des *changements* possibles d'un système d'argumentation, à savoir l'addition ou la suppression d'un argument et l'addition ou la suppression d'une interaction, ainsi qu'une *typologie du changement* listant les impacts de ce changement sur un système d'argumentation du point de vue de ses extensions (ces impacts sont appelés *propriétés du changement*).

Le **chapitre 3** constitue notre premier apport. Dans celui-ci, nous donnons une justification de l'intérêt du cadre de Dung dans des situations concrètes. Pour ce faire, nous détaillons trois exemples sur lesquels nous exposons et explicitons les notions théoriques que nous utilisons, que ce soit au niveau d'un système d'argumentation ou des modifications que ses extensions peuvent subir. Nous y justifions notamment une opération de changement négligée jusque-là, la suppression d'un argument.

La **deuxième partie** propose une extension et un enrichissement des notions sur lesquelles nous nous basons. Elle est constituée des chapitres 4 et 5.

Le **chapitre 4** affine la typologie du changement sur laquelle nous nous basons en la décomposant en trois niveaux. Cela nous permet de représenter tout type de modification des extensions d'un système d'argumentation et de prendre en compte à la fois l'opération d'addition d'un argument et l'opération de suppression d'un argument.

Le **chapitre 5** se place dans la droite lignée du chapitre 4. Il propose une caractérisation des propriétés du changement, c'est-à-dire des conditions nécessaires et/ou suffisantes assurant ces propriétés. Plus précisément, pour une propriété du changement donnée, si une opération de changement remplit un certain nombre de conditions la concernant et concernant le système sur lequel elle est appliquée, alors nous sommes sûr que le résultat vérifiera cette propriété. Pour nous aider dans notre tâche de caractérisation, nous proposons une méthodologie permettant de transformer un ensemble de conditions concernant une opération (addition ou suppression d'un argument) en un autre ensemble de conditions concernant l'opération duale.

La **troisième partie** est constituée des chapitres 6 et 7. Elle propose de contraindre le changement pour le rendre moins trivial, ainsi que de l'axiomatiser.

Le **chapitre 6** introduit un nouveau cadre formel mettant en exergue l'importance de la notion de but pour le choix de l'opération de changement à faire. Dans ce cadre, deux systèmes d'argumentation sont présents. Le premier représente les connaissances d'un agent voulant en convaincre un autre, alors que le deuxième représente les connaissances de l'agent à convaincre (le système cible). Le premier agent doit choisir une opération permettant de satisfaire son but. Cette opération doit de plus lui être possible ; il doit en effet connaître les éléments du changement qu'il propose et ces éléments doivent être cohérents avec le système cible. En outre, nous étudions comment cet agent peut choisir un changement *optimal*, c'est-à-dire le "meilleur" changement à opérer suivant certains critères.

Le **chapitre 7** propose un parallèle entre l'argumentation et une approche classique permettant de gérer le changement dans le cadre des bases de croyances, la mise à jour de croyances. L'axiomatisation issue de ce parallèle nous permet d'étendre notre cadre au cas où le système cible n'est pas entièrement connu.

La **quatrième partie** s'intéresse à l'implémentation de notre travail théorique. Elle est constituée de l'unique **chapitre 8**. L'outil proposé permet à un utilisateur de trouver automatiquement une opération de changement à partir d'un système représentant ses connaissances et d'un système cible. Après avoir présenté la structure de l'outil, nous exposons quelques expérimentations et analysons les résultats.

Le **chapitre 9** conclut notre travail. Nous y rappelons l'ensemble de nos apports ainsi que les perspectives que nous voyons à notre travail.

Le schéma 1.1 page suivante présente visuellement la structure principale de la thèse et les dépendances entre chapitres.

Notre travail comporte par ailleurs plusieurs annexes complétant les apports des chapitres principaux.

L'**annexe A** récapitule l'ensemble des notations que nous utilisons tout au long de ce travail. L'**annexe B** rappelle quelques notions utiles à notre travail. L'**annexe C** donne l'ensemble des démonstrations des propositions et lemmes parsemés dans les chapitres principaux. Enfin, l'**annexe D** prolonge les justifications de notre travail en donnant de nouveaux exemples de buts commentés contextuellement ainsi que deux exemples montrant l'intérêt des opérations de suppression d'un argument et de suppression d'une attaque dans les situations d'évolution du contexte et des priorités.

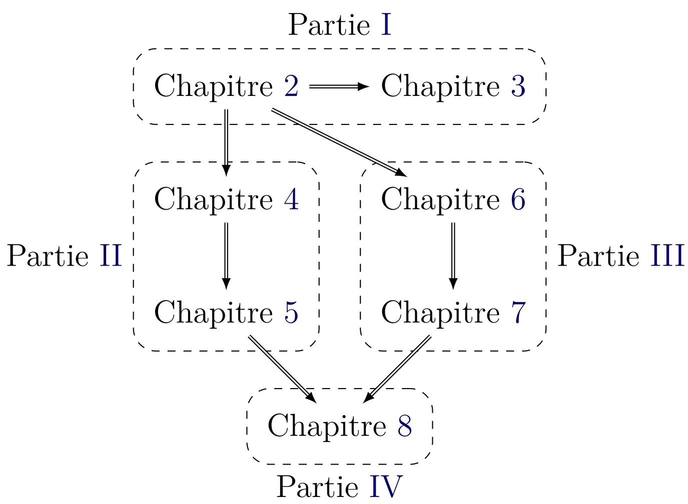


Figure 1.1 – *Guide de lecture des chapitres principaux. Les flèches indiquent les relations de dépendance entre ces chapitres.*

Notons, pour terminer cette introduction, que l'ensemble de nos travaux, hormis le chapitre 3, a été publié (éventuellement en version réduite). En voici la liste :

- la typologie du chapitre 4 est présentée en partie dans Bisquert (2011); Bisquert *et al.* (2011, 2012b) ;
- les diverses caractérisations du chapitre 5 apparaissent dans Bisquert (2011); Bisquert *et al.* (2012a,b, 2013b) ;
- le cadre formel du chapitre 6 sert d'introduction aux travaux de Bisquert *et al.* (2013c,d) ;
- l'axiomatisation du changement du chapitre 7 est publiée dans Bisquert *et al.* (2013c) ;
- l'implémentation et les expérimentations du chapitre 8 ont été exposées dans Bisquert *et al.* (2013a,d).

Première partie

Bases et intuition

Chapitre 2

Bases théoriques

Changer le monde commence par se changer soi-même.

Roger Mondoloni,
L'Aube du temps qui vient.

Sommaire

Un travail, quel qu'il soit, ne saurait être fait *ex nihilo*. Toute idée est inspirée d'une situation rencontrée, toute réflexion est basée sur un travail précédent. Aussi, nous présentons dans ce chapitre les deux bases de notre travail, le modèle de Dung (1995) et l'approche de Cayrol *et al.* (2010c). La première constitue l'un des cadres formels pour l'argumentation les plus abstraits qui soit, tandis que la deuxième permet d'aborder l'aspect de la dynamique en donnant des moyens de gérer les modifications qu'une argumentation peut subir.

Un cadre tel que celui de Dung (1995), en se détachant du sens interne des arguments et en ne gardant que les principes généraux de l'argumentation, en particulier les interactions entre arguments, permet de se concentrer sur d'autres facettes. En particulier, il permet d'étudier la dynamique d'une argumentation, par exemple l'ajout ou le retrait d'arguments au cours d'un dialogue, en ne s'intéressant qu'aux relations entre les arguments, c'est-à-dire sans s'intéresser aux arguments en eux-mêmes. En étant à ce point général, il présente des caractéristiques intéressantes quant à l'étude globale d'un processus aussi compliqué que l'argumentation. En particulier, ce cadre permet le traitement et le raisonnement sur des informations parcellaires ou contradictoires, sans avoir besoin de connaître à proprement parler ces informations.

Par ailleurs, le caractère abstrait et général du cadre de Dung (1995) lui permet de s'appliquer à de nombreux contextes, moyennant une justification de ses notions théoriques. Il en découle des applications variées, allant de l'inévitable représentation d'un tribunal à l'aide à la création d'une équipe de sport. Nous aurons l'occasion d'étudier en profondeur trois de ces applications dans le chapitre 3 page 29.

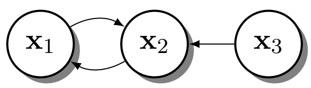
Nous commençons donc par présenter le cadre de Dung dans la section 2.1. La section 2.2 page 16 se propose quant à elle de présenter quelques éléments sur la dynamique des systèmes d'argumentation. Enfin, la section 2.3 page 21 évoque certaines des autres approches permettant le traitement de l'argumentation.

2.1 Système d'argumentation à la Dung

Le cadre de Dung (1995) est dit abstrait car il ne nécessite aucune information sur les objets manipulés. Plus précisément, ni la structure ni l'origine des arguments ne sont requises. Ainsi, un système d'argumentation consiste simplement en un ensemble d'arguments et une relation binaire symbolisant les interactions entre eux. Les interactions peuvent être interprétées librement, et dépeignent généralement les attaques entre arguments. [1]

Définition 2.1 (**Système d'argumentation** (Dung (1995))). *Un* système d'argumentation *est une paire $(\mathcal{A}, \mathcal{R})$, où $\mathcal{A}$ est un ensemble non vide fini d'arguments et $\mathcal{R}$ est une relation binaire sur $\mathcal{A}$, appelée* relation d'attaque*. Soit $x, y \in \mathcal{A}$, $x\mathcal{R}y$ signifie que x attaque y. $(\mathcal{A}, \mathcal{R})$ est représenté par un* graphe d'argumentation $\mathcal{G}$ *dont les sommets et les arcs correspondent respectivement aux arguments et aux attaques*[2].

Exemple 1.1 (cont.). *Si nous utilisons le graphe d'argumentation, nous pouvons représenter le rapide échange d'arguments de l'introduction de la façon suivante :*



Dans la suite de ce document, nous aurons besoin d'une notion étendue de l'attaque, à savoir l'attaque d'un argument vers un ensemble et vice versa.

Définition 2.2 (**Attaque par et vers un ensemble**). *Soit $\mathcal{G}$ un système d'argumentation, $x \in \mathcal{A}$ et $\mathcal{S} \subseteq \mathcal{A}$,*

1. Des interactions de type "support" peuvent également être considérées : lorsque un argument supporte un autre argument, le premier constitue une justification augmentant l'acceptation du dernier. Ainsi, certains travaux concernent des systèmes d'argumentation contenant à la fois attaques et supports (voir notamment Amgoud *et al.* (2008), Cayrol et Lagasquie-Schiex (2009) ou encore Cayrol et Lagasquie-Schiex (2013)).

2. Dans ce document, nous utiliserons librement $(\mathcal{A}, \mathcal{R})$ ou $\mathcal{G}$ pour faire allusion à un système d'argumentation. De façon similaire, lorsqu'il n'y a pas d'ambiguïté, nous utiliserons sans distinction $\mathcal{A}$ et $\mathcal{G}$ pour dénoter l'ensemble des arguments d'un système d'argumentation.

- $\mathcal{S}$ attaque x *si et seulement si* $\exists y \in \mathcal{S}$ *tel que* $y\mathcal{R}x$. *Plus généralement,* $\mathcal{S}$ attaque indirectement x *si et seulement si* $\exists y \in \mathcal{S}$ *tel qu'il existe un chemin de longueur impaire de* y *vers* x *dans* $\mathcal{G}$.
- x attaque $\mathcal{S}$ *si et seulement si* $\exists y \in \mathcal{S}$ *tel que* $x\mathcal{R}y$.

Étant donné un système d'argumentation $(\mathcal{A}, \mathcal{R})$, l'un des enjeux est de déterminer les ensembles acceptables d'arguments ("*extensions*"), ce qui est réalisé à l'aide de sémantiques. Une sémantique est définie par un ensemble de conditions qu'un ensemble d'arguments doit satisfaire pour être considéré comme acceptable au regard de cette dernière.

Les sémantiques, bien que différentes, ont certaines conditions en commun. Notamment, le fait que les extensions *représentent des points de vue cohérents* et doivent donc être *sans conflit*.

Par ailleurs, un point de vue ne doit contenir que des arguments "justifiés", c'est-à-dire défendus s'ils sont attaqués. Un argument est *défendu* par un ensemble d'arguments si (et seulement si) tous les attaquants de cet argument sont eux-mêmes attaqués par des arguments appartenant à cet ensemble. Notons qu'un argument non attaqué est également défendu. Ainsi, tout point de vue doit *défendre ses propres arguments*.

En définitive, un ensemble d'arguments ne peut être considéré comme un point de vue (formellement, une extension) qu'à la condition qu'il soit à la fois sans conflit et qu'il défende tous ses éléments ; nous dirons qu'un tel ensemble est *admissible*.

Formellement, ces notions sont traduites de la façon suivante :

Définition 2.3 (**Sans conflit, défense et admissibilité**). *Soit* $x \in \mathcal{A}$ *et* $\mathcal{S} \subseteq \mathcal{A}$

- $\mathcal{S}$ *est* sans conflit *si et seulement s'il n'existe pas* $x, y \in \mathcal{S}$ *tels que* $x\mathcal{R}y$.
- $\mathcal{S}$ défend *un argument* x *si et seulement si* $\mathcal{S}$ *attaque tout argument attaquant* x. *L'ensemble des arguments défendus par* $\mathcal{S}$ *est noté* $\mathcal{F}(\mathcal{S})$ *;* $\mathcal{F}$ *est appelée la* fonction caractéristique *de* $(\mathcal{A}, \mathcal{R})$. *Plus généralement,* $\mathcal{S}$ défend indirectement x *si et seulement si* $x \in \bigcup_{i \geq 1} \mathcal{F}^i(\mathcal{S})$.
- $\mathcal{S}$ *est un ensemble* admissible *si et seulement s'il est à la fois sans conflit et qu'il défend tous ses éléments.*

Exemple 1.1 (cont.). *Dans le système d'argumentation représentant la discussion entre Rosa et Georges, nous pouvons dire que :*

- *l'ensemble* $\{x_1, x_3\}$ *est sans conflit, contrairement à l'ensemble* $\{x_1, x_2\}$,
- *l'ensemble* $\{x_3\}$ *défend* x_1 *mais l'ensemble* $\{x_1\}$ *ne défend pas* x_2,
- *l'ensemble* $\{x_1, x_3\}$ *est admissible, contrairement à l'ensemble* $\{x_1, x_2\}$.

Notation 2.1. *L'ensemble des extensions de* $(\mathcal{A}, \mathcal{R})$ *est noté* $\mathbf{E}$ *(avec* $\mathcal{E}_1, \mathcal{E}_2, \cdots$ *dénotant les extensions).*

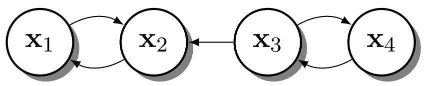
Ainsi, en nous limitant aux sémantiques les plus utilisées proposées par Dung (1995), nous avons :

Définition 2.4 (Sémantiques d'acceptabilité). *Soit $\mathcal{E} \subseteq \mathcal{A}$.*

- $\mathcal{E}$ *est* une extension préférée *si et seulement si* $\mathcal{E}$ *est un ensemble admissible maximal (par rapport à l'inclusion ensembliste* $\subseteq$*).*
- $\mathcal{E}$ *est* l'unique extension basique *si et seulement si* $\mathcal{E}$ *est le plus petit point fixe (par rapport à* $\subseteq$*) de la fonction caractéristique* $\mathcal{F}$*.*
- $\mathcal{E}$ *est* une extension stable *si et seulement si* $\mathcal{E}$ *est sans conflit et attaque tout argument n'appartenant pas à* $\mathcal{E}$*.*

Exemple 1.1 (cont.). *Si l'on se réfère au système d'argumentation représentant la discussion entre Rosa et Georges, un seul point de vue apparaît. Celui-ci est représenté, dans le cadre des sémantiques basique, préférée et stable, par l'extension* $\{x_1, x_3\}$ *ce qui signifie que Jean, bien qu'il soit un gastronome, ne connaît pas suffisamment ce type de cuisine pour être capable de se prononcer sur sa qualité.*

Exemple 2.1. *Supposons que Rosa reprenne la parole et ajoute le nouvel argument suivant :* (x_4) "Si, Jean a eu l'occasion de suivre des cours avec un spécialiste de ce type de cuisine." *Nous obtenons alors le système d'argumentation suivant :*



Nous pouvons voir dans le système que :

- *l'extension basique est vide* ($\{\}$)*, signifiant qu'aucun point de vue valable ne peut être tiré selon la sémantique basique ;*
- *les extensions préférées et stables sont égales à* $\{x_1, x_4\}$, $\{x_1, x_3\}$ *et* $\{x_2, x_4\}$*, traduisant les trois points de vue possibles dans ces sémantiques.*

Notons que Dung (1995) propose quelques propositions (ainsi que leurs preuves) établissant des liens entre ces différentes sémantiques ; nous les résumons ci-dessous :

Proposition 2.1 (Dung (1995)).

1. *Il y a au moins une extension préférée, toujours une unique extension basique, et aucune, une ou plusieurs extensions stables.*
2. *Tout ensemble admissible est contenu dans une extension préférée.*
3. *Toute extension stable est également préférée, l'inverse étant faux.*
4. *Toute extension préférée (resp. stable) contient l'extension basique.*
5. *Tout argument n'étant pas attaqué appartient à l'extension basique (et donc à chaque extension préférée et chaque extension stable).*
6. *La fonction* $\mathcal{F}$ *est monotone.*
7. *Si* $\mathcal{R}$ *est fini, l'extension basique peut être calculée en appliquant itérativement la fonction* $\mathcal{F}$ *à partir de l'ensemble vide.*
8. *Si* $\mathcal{A}$ *n'est pas vide, alors une extension stable est toujours non vide.*

Nous aurons également besoin, plus loin dans ce travail, de certaines propositions de Dunne et Bench-Capon (2001, 2002) établissant des liens entre d'éventuels cycles présents dans un système d'argumentation et les extensions de ce dernier.

Proposition 2.2 (Dunne et Bench-Capon (2001, 2002))**.**

1. *Si $\mathcal{G}$ ne contient aucun cycle, alors $(\mathcal{A}, \mathcal{R})$ a une unique extension préférée, qui est également l'extension basique et l'unique extension stable.*
2. *Si $\{\}$ est l'unique extension préférée de $(\mathcal{A}, \mathcal{R})$, alors $\mathcal{G}$ contient un cycle de longueur impaire.*
3. *Si $(\mathcal{A}, \mathcal{R})$ n'a pas d'extension stable, alors $\mathcal{G}$ contient un cycle de longueur impaire.*
4. *Si $\mathcal{G}$ ne contient aucun cycle de longueur impaire, alors les extensions préférée et stable coïncident.*
5. *Si $\mathcal{G}$ ne contient aucun cycle de longueur paire, alors $(\mathcal{A}, \mathcal{R})$ a une unique extension préférée.*

Nous l'avons vu, dans ce cadre, l'acceptabilité est collective. Néanmoins, à partir des ensembles d'arguments acceptables, nous pouvons déterminer l'acceptabilité des arguments. En effet, le statut d'un argument est fonction de sa présence dans les extensions de la sémantique choisie. Intuitivement, un argument présent dans tous les points de vue n'est aucunement mis en doute ; il est en quelque sorte “fortement” accepté. Un argument présent dans au moins un mais pas tous les points de vue est quant à lui en partie admis ; il est plus “faiblement” accepté. Un argument n'étant présent dans aucun point de vue est unanimement écarté.

Définition 2.5 (Statut d'un argument). *Soit $(\mathcal{A}, \mathcal{R})$ un système d'argumentation et $x \in \mathcal{A}$ un argument. Étant donné une sémantique définissant un ensemble* $\mathbf{E}$ *d'extensions à partir de $\mathcal{G}$:*

- *x est* accepté sceptiquement *si et seulement s'il appartient à toutes les extensions de $(\mathcal{A}, \mathcal{R})$ ($\forall \mathcal{E} \in \mathbf{E}, x \in \mathcal{E}$) ;*
- *x est* accepté crédulement *si et seulement s'il appartient à au moins une extension de $(\mathcal{A}, \mathcal{R})$ ($\exists \mathcal{E} \in \mathbf{E}, x \in \mathcal{E}$) ;*
- *x est* rejeté *si et seulement s'il n'appartient à aucune des extensions de $(\mathcal{A}, \mathcal{R})$ ($\forall \mathcal{E} \in \mathbf{E}, x \notin \mathcal{E}$).*

Notons que dans le cas où il n'y a qu'une seule extension, tout argument lui appartenant sera alors accepté sceptiquement et crédulement. Plus généralement, tout argument accepté sceptiquement est accepté crédulement, l'inverse n'étant pas vrai.

Exemple 2.1 (cont.). *Dans le système d'argumentation représentant la discussion entre Rosa et Georges,*

- *avant que l'argument x_4 ne soit avancé, dans le cadre des sémantiques basique et préférée,*

  — *x_1 et x_3 sont acceptés sceptiquement et crédulement,*
  — *x_2 est rejeté.*
- *après que l'argument x_4 soit avancé :*
  — *dans le cadre de la sémantique basique, les quatre arguments sont rejetés ;*
  — *dans le cadre des sémantiques préférée et stable, les quatre arguments sont acceptés crédulement.*

2.2 Changement

Notre travail se fonde également sur les notions introduites dans Cayrol *et al.* (2010c). Plus précisément, nous reprenons la notion d'"opération de changement". Une opération de changement, telle que l'addition ou la suppression d'un argument, provoque une modification d'un système d'argumentation et permet d'obtenir un nouveau système d'argumentation. Nous modifierons et approfondirons cette notion dans la suite de ce travail.

Selon Cayrol *et al.* (2010c), il existe quatre façons "élémentaires" de modifier un système d'argumentation :

- *Ajouter une interaction*, lorsque, par exemple, on veut mentionner une nouvelle attaque.
- *Supprimer une interaction*, notamment lorsqu'une attaque est illégale ou que l'on ne veut pas en tenir compte.
- *Ajouter un argument* ainsi que, éventuellement, des interactions le concernant ; ceci constitue l'acte naturel d'énonciation d'un nouvel argument qui, le plus souvent, attaque d'autres arguments.
- *Supprimer un argument* ainsi que toutes les interactions le concernant, lorsque, par exemple, un argument est reconnu illégal.

La définition 2.6 rappelle formellement ces quatre opérations telles qu'elles sont définies dans Cayrol *et al.* (2010c).

Définition 2.6 (**Opérations élémentaires de changement**, Cayrol *et al.* (2010c))**.** *Soit $(\mathcal{A}, \mathcal{R})$ un système d'argumentation, $(x, y) \in \mathcal{A} \times \mathcal{A}$ un couple d'arguments et z.*

- Ajouter une interaction $(x, y) \notin \mathcal{R}$ *est une opération de changement, notée $\oplus$, fournissant un nouveau système d'argumentation tel que :*

$$(\mathcal{A}', \mathcal{R}') = (\mathcal{A}, \mathcal{R}) \bigoplus (x, y) = (\mathcal{A}, \mathcal{R} \cup \{(x, y)\})$$

- Supprimer une interaction $(x, y) \in \mathcal{R}$ *est une opération de changement, notée $\ominus$, fournissant un nouveau système d'argumentation tel que :*

$$(\mathcal{A}', \mathcal{R}') = (\mathcal{A}, \mathcal{R}) \bigominus (x, y) = (\mathcal{A}, \mathcal{R} \setminus \{(x, y)\})$$

- Ajouter un argument $z \notin \mathcal{A}$ *avec un ensemble* $\mathcal{R}_z$ *d'interactions concernant* z [3] *est une opération de changement, notée* $\bigoplus$, *fournissant un nouveau système d'argumentation tel que :*

$$(\mathcal{A}', \mathcal{R}') = (\mathcal{A}, \mathcal{R}) \bigoplus (z, \mathcal{R}_z) = (\mathcal{A} \cup \{z\}, \mathcal{R} \cup \mathcal{R}_z)$$

- Supprimer un argument $z \in \mathcal{A}$ *avec l'ensemble* $\mathcal{R}_z = \{(x, y) \in \mathcal{R} \mid x = z \text{ ou } y = z\}$ *de toutes les interactions concernant* z *est une opération de changement, notée* $\bigominus$, *fournissant un nouveau système d'argumentation tel que*

$$(\mathcal{A}', \mathcal{R}') = (\mathcal{A}, \mathcal{R}) \bigominus (z, \mathcal{R}_z) = (\mathcal{A} \setminus \{z\}, \mathcal{R} \setminus \mathcal{R}_z)$$

Une opération élémentaire de changement au sens de Cayrol *et al.* (2010c) est donc une fonction d'arité deux dont le premier paramètre est toujours un système d'argumentation et dont le second paramètre est, suivant les cas, une interaction (pour $\oplus$ et $\ominus$) ou un couple (argument, ensemble d'interactions concernant l'argument) (pour $\bigoplus$ et $\bigominus$).

Notation 2.2. *L'ensemble des extensions de* $(\mathcal{A}', \mathcal{R}')$ *est noté* $\mathbf{E}'$ *(avec* $\mathcal{E}'_1, \mathcal{E}'_2, \ldots$ *dénotant les extensions).*

Exemple 2.2. *Le système d'argumentation présenté dans l'exemple 1.1 page 3 a subi un changement lors de l'exemple 2.1 page 14. Plus précisément, Rosa a ajouté l'argument* x_4 *ainsi que l'ensemble d'attaques* $\{(x_3, x_4), (x_4, x_3)\}$. *En conséquence, le système d'argumentation a été modifié et nous obtenons :*

$$\begin{aligned}(\mathcal{A}', \mathcal{R}') &= (\mathcal{A}, \mathcal{R}) \bigoplus (x_4, \mathcal{R}_{x_4}) \\ &= (\{x_1, x_2, x_3\}, \{(x_1, x_2), (x_2, x_1), (x_3, x_2)\}) \bigoplus \\ &\quad\ (x_4, \{(x_3, x_4), (x_4, x_3)\}) \\ &= (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_2), (x_2, x_1), (x_3, x_2), (x_3, x_4), (x_4, x_3)\}).\end{aligned}$$

Remarquons qu'un même système d'argumentation peut être obtenu via différentes opérations de changement, comme l'illustre l'exemple 2.3. Néanmoins, il est important de noter que la définition 2.6 page précédente nous assure que $\forall \mathcal{G}$, $\mathcal{G}' = \mathcal{G} \bigominus (z, \mathcal{R}_z)$ est unique même si les attaques à supprimer (l'ensemble $\mathcal{R}_z$) associées à $\bigominus$ ne sont pas précisées.

Exemple 2.3. *Chacun des systèmes* $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{G}_3$ *est changé en* $\mathcal{G}'$ *par la suppression de l'argument* z *et des interactions le concernant ; néanmoins, chaque changement correspond à une opération distincte puisque* $\mathcal{R}_z$ *est différent. Le système* $\mathcal{G}_4$ *est changé en* $\mathcal{G}'$ *par l'addition de l'argument* u *et de l'ensemble d'interactions* $\{(x, u), (u, x)\}$. *La table 2.1 page suivante présente ces graphes, ainsi que l'extension basique pour chacun d'entre eux.*

3. Notons que Cayrol *et al.* (2010c) imposent que z ne s'attaque pas lui-même et que $\forall (x, y) \in \mathcal{R}_z$, on a soit ($x = z$ et $y \neq z$, $y \in \mathcal{A}$) soit ($y = z$ et $x \neq z$, $x \in \mathcal{A}$).

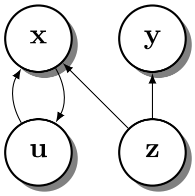
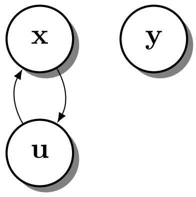
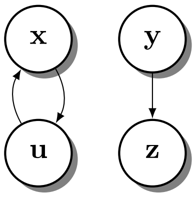
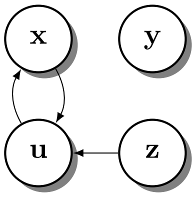
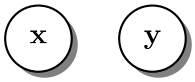
Système avant le changement	Opération de changement	Système après le changement
$\mathcal{G}_1$:  $\mathbf{E}_{\mathcal{G}_1} = \{\{u, z\}\}$	$o_1 = \ominus(z, \mathcal{R}_z)$ avec $\mathcal{R}_z = (\{(z, x), (z, y)\})$	$\mathcal{G}'$:  $\mathbf{E}'_{\mathcal{G}'} = \{\{y\}\}$
$\mathcal{G}_2$:  $\mathbf{E}_{\mathcal{G}_2} = \{\{y\}\}$	$o_2 = \ominus(z, \mathcal{R}_z)$ avec $\mathcal{R}_z = (\{(y, z)\})$	
$\mathcal{G}_3$:  $\mathbf{E}_{\mathcal{G}_3} = \{\{x, y, z\}\}$	$o_3 = \ominus(z, \mathcal{R}_z)$ avec $\mathcal{R}_z = (\{(z, u)\})$	
$\mathcal{G}_4$:  $\mathbf{E}_{\mathcal{G}_4} = \{\{x, y\}\}$	$o_4 = \oplus(u, \mathcal{R}_u)$ avec $\mathcal{R}_u = (\{(x, u), (u, x)\})$	

Table 2.1 – *Illustration de la suppression d'un argument (graphes $\mathcal{G}_1$, $\mathcal{G}_2$ et $\mathcal{G}_3$) et de l'addition d'un argument (graphe $\mathcal{G}_4$) ; $\mathbf{E}_\mathcal{G}$ représente l'ensemble des extensions basiques de $\mathcal{G}$.*

Ceci illustre la dynamique des systèmes d'argumentation : un système d'argumentation peut subir un changement et être modifié. Par voie de conséquence, les extensions elles aussi peuvent être modifiées. On peut alors se demander s'il est possible de distinguer différents types de modification des extensions.

Pour répondre à cette interrogation, une classification a été établie dans Cayrol *et al.* (2010c) sous la forme d'une liste de plusieurs types de modifications possibles des extensions, appelées *propriétés du changement*. Ces propriétés sont classées selon deux grandes familles :

- les propriétés *structurelles* concernant les modifications de la cardinalité des ensembles d'arguments ou des relations d'inclusion,
- les propriétés *basées sur le statut* des arguments, notamment sur leur acceptabilité.

La table 2.2 présente les propriétés du changement structurelles définies par Cayrol *et al.* (2010c), ainsi que leurs définitions.

Propriété du changement	Définition
Décisif ("decisive")	$\mathbf{E} = \varnothing$ ou $\mathbf{E} = \{\{\}\}$ ou $\lvert\mathbf{E}\rvert > 2$ et $\lvert\mathbf{E}'\rvert = 1$ et $\mathbf{E}' \neq \{\{\}\}$
Restrictif ("restrictive")	$\lvert\mathbf{E}\rvert > \lvert\mathbf{E}'\rvert > 2$
Ouvrant ("questioning")	$\lvert\mathbf{E}\rvert < \lvert\mathbf{E}'\rvert$
Destructif ("destructive")	$\mathbf{E} \neq \varnothing$ ou $\mathbf{E} \neq \{\{\}\}$ et $\mathbf{E}' = \varnothing$ ou $\mathbf{E}' = \{\{\}\}$
Expansif ("expansive")	$\lvert\mathbf{E}\rvert = \lvert\mathbf{E}'\rvert$ et $\forall \mathcal{E}'_j \in \mathbf{E}', \exists \mathcal{E}_i \in \mathbf{E}, \mathcal{E}_i \subset \mathcal{E}'_j$
Conservatif ("conservative")	$\mathbf{E} = \mathbf{E}'$
Modifiant ("altering")	$\lvert\mathbf{E}\rvert = \lvert\mathbf{E}'\rvert$ et $\exists \mathcal{E}_i \in \mathbf{E}$ *t.q.* $\forall \mathcal{E}'_j \in \mathbf{E}', \mathcal{E}_i \not\subseteq \mathcal{E}'_j$

Table 2.2 – *Liste des propriétés du changement de Cayrol* et al. *(2010c).*

Intuitivement, un changement est : [4]

- *décisif* ("decisive") lorsque le système n'a pas d'extension, a une extension vide ou plusieurs extensions avant le changement, et qu'il a exactement une extension non vide après le changement ;
- *restrictif* ("restrictive") lorsque le système perd des extensions et qu'il a donc plus d'extensions avant le changement qu'après (tout en gardant un minimum de deux extensions) ;
- *ouvrant* ("questioning") lorsque le système gagne des extensions et qu'il a donc moins d'extensions avant le changement qu'après ;
- *destructif* ("destructive") lorsque le système a une extension non vide ou plusieurs avant le changement, et qu'il n'a aucune extension ou qu'il a une extension vide après le changement ;
- *expansif* ("expansive") lorsque le système a le même nombre d'extensions, mais que ces dernières s'accroissent (chacune des extensions avant le changement est incluse dans une extension après le changement) ;
- *conservatif* ("conservative") lorsque le système a exactement les mêmes extensions avant et après le changement ;
- *modifiant* ("altering") lorsque le système a le même nombre d'extensions, mais qu'elles ne correspondent pas (il existe une extension avant le changement qui n'est incluse dans aucune des extensions après le changement).

4. Les termes entre parenthèses correspondent aux noms originaux des propriétés du changement définies dans Cayrol *et al.* (2010c).

Les propriétés du changement basées sur le statut concernent deux points particuliers, la *monotonie* et la *priorité à la nouveauté* ("priority to recency").

La monotonie s'intéresse aux arguments acceptés avant le changement et qui le restent après le changement. En termes d'argumentation abstraite, cela signifie qu'elle concerne les inclusions entre les extensions avant le changement et celles après le changement. En plus de cette monotonie, Cayrol *et al.* (2010c) distinguent deux autres types de monotonie de façon à tenir compte des différents cas d'acceptation d'un argument :

- la *monotonie* qui est satisfaite par un changement de $\mathcal{G}$ à $\mathcal{G}'$ si et seulement si toute extension de $\mathcal{G}$ est incluse dans au moins une extension de $\mathcal{G}'$,
- la *monotonie crédule* qui est satisfaite par un changement de $\mathcal{G}$ à $\mathcal{G}'$ si et seulement si l'union des extensions de $\mathcal{G}$ est incluse dans l'union des extensions de $\mathcal{G}'$ et
- la *monotonie sceptique* qui est satisfaite par un changement de $\mathcal{G}$ à $\mathcal{G}'$ si et seulement si l'intersection des extensions de $\mathcal{G}$ est incluse dans l'intersection des extensions de $\mathcal{G}'$.

Il est possible d'étendre cette notion de monotonie à un argument particulier. Pour cela, Cayrol *et al.* (2010c) définissent la notion de *monotonie partielle* qui est satisfaite par un changement de $\mathcal{G}$ à $\mathcal{G}'$ pour un argument x si et seulement si lorsque x appartient à une extension de $\mathcal{G}$, x appartient également à au moins une extension de $\mathcal{G}'$.

La *priorité à la nouveauté* concerne uniquement le statut de l'argument ajouté lors d'une opération d'addition d'un argument. Elle est satisfaite par un changement de $\mathcal{G}$ à $\mathcal{G}'$ causé par l'addition d'un argument z si et seulement si $\mathcal{G}'$ a au moins une extension et z appartient à toutes les extensions de $\mathcal{G}'$.

Ces différentes propriétés du changement (ainsi que leurs liens) ont été étudiées par Cayrol *et al.* (2010c) et caractérisées dans le cadre des sémantiques préférée et basique, et uniquement lors de l'addition d'un argument. Une caractérisation définit un ensemble de conditions qui, si elles sont satisfaites par un système d'argumentation donné et une opération sur ce système, garantissent que le système subira une modification particulière de ses extensions (cette modification étant indiquée par la caractérisation).

Nous verrons dans le chapitre 5 page 81 les caractérisations principales de Cayrol *et al.* (2010c) que nous utilisons.

Les travaux présentés dans les sections précédentes, s'ils ont servi de base à nos recherches, ne sont pas les seuls dans le domaine de l'argumentation. La section suivante présente certaines des autres approches majeures en argumentation.

2.3 Discussion et travaux liés

Domaine fertile, l'argumentation formelle a donné lieu à de nombreux travaux. Nous avons présenté dans ce chapitre un des cadres formels les plus utilisés (Dung (1995)) ainsi qu'une approche facilitant l'étude du changement dans ce cadre (Cayrol *et al.* (2010c)).

Du fait de sa généralité, le cadre de Dung a l'avantage de pouvoir s'appliquer à beaucoup de contextes différents, y compris ceux n'ayant qu'un rapport lointain avec le fait de convaincre. L'autre avantage majeur de ce cadre est qu'il permet une représentation simple et facile à manipuler. En faisant abstraction de la manière dont sont construits les éléments de base et en reposant principalement sur des graphes, il correspond à ce dont nous avons besoin pour étudier des concepts de hauts niveaux tels que le changement.

Néanmoins, de nombreux autres travaux existent. Ainsi, nous nous intéressons, dans cette section, aux autres approches formelles. Pour cela, nous présentons tout d'abord des cadres formels alternatifs dans la section 2.3.1, puis nous étudions d'autres approches permettant le traitement de la dynamique dans la section 2.3.2 page 27.

2.3.1 Cadres formels alternatifs

Ainsi, le cadre de Dung n'est pas la seule alternative pour aborder l'argumentation formelle. Mais qu'est-il possible d'utiliser en lieu et place de ce dernier ? Pour répondre à cette question, nous présentons tout d'abord l'argumentation logique, puis des extensions du cadre de Dung.

2.3.1.1 Argumentation logique

Le cadre de Dung n'est pas exempt de défauts. En particulier, son utilisation, de par son parti pris très abstrait, a pour conséquence une perte des informations contenues “dans” les arguments, ces derniers n'étant plus que des sommets dans un graphe. À ce titre, il est par exemple complexe de représenter tout ce qui a trait à l'illocutoire.[5] En effet, bien qu'il soit possible de supposer que certaines attaques sont fournies dans le but de considérer la dimension illocutoire, ne pas savoir ce qu'un argument représente et dans quel contexte il est énoncé rend difficile l'étude de l'impact qu'il peut avoir sur un auditoire.

Contrairement au cadre de Dung, qui suppose que les arguments sont fournis bien que leur structure ne soit pas connue, l'approche logique de l'argumentation tente de construire des arguments à partir d'une base de connaissances. Elle débute

5. La fonction *illocutoire* d'un énoncé correspond au message que l'on veut faire passer au-delà de ce que l'on dit (qui correspond au côté *locutoire* de l'énoncé). Par exemple, demander à quelqu'un *Avez-vous l'heure ?* n'a pas pour but de s'assurer que cette personne a bien sa montre (locutoire), mais traduit le souhait d'être renseigné sur l'heure qu'il est (illocutoire). Le fait que cette personne donne l'heure (en principe), c'est-à-dire la modification de son comportement suite à l'énoncé, est appelé fonction *perlocutoire* (Austin (1962)).

donc son traitement de l'argumentation plus “en amont”. Celle-ci repose, comme son nom l'indique, sur une logique (logique propositionnelle, logique du premier ordre, etc.).

Notons qu'en argumentation logique, notamment dans Besnard et Hunter (2008), un argument est généralement représenté par une paire (P, c), où P, correspondant aux prémisses de l'argument, est un ensemble de formules issues d'une base de connaissances, et c est une formule représentant la conclusion de l'argument, telle que :

- P est cohérent ($P \nvdash \perp$),
- P permet de dériver c ($P \vdash c$),
- P est minimal ($\nexists P' \subset P$ tel que $P' \vdash c$).

Le système *ASPIC* (Amgoud *et al.* (2006); Caminada et Amgoud (2007)) est un autre exemple de cadre logique pour l'argumentation. Notons néanmoins que dans le cas d'*ASPIC*, les arguments ne sont pas des paires de prémisses et conclusions, mais des arbres d'inférence construits en appliquant des règles d'inférence (strictes ou révisables) à des prémisses.

L'argumentation logique a largement été étudiée. Elle a notamment été utilisée dans trois domaines particuliers :

- le raisonnement avec des informations contradictoires,
- la décision et le choix d'intentions d'un agent,
- l'interaction multi-agents.

Nous donnons maintenant quelques travaux typiques dans ces domaines.

Raisonnement

L'argumentation logique est une instanciation du cadre de Dung et à ce titre, elle permet de traiter naturellement la question du raisonnement. Du fait de son caractère logique, de nombreux travaux l'ont utilisé pour manipuler les bases de connaissances et leur point faible inhérent, à savoir l'incohérence des connaissances (Simari et Loui (1992); Besnard et Hunter (2001, 2008); Amgoud et Cayrol (2002a); García et Simari (2004); Governatori *et al.* (2004)). En particulier, l'argumentation logique permet d'extraire de ces bases des informations cohérentes et pertinentes. Pour arriver à ses fins, l'argumentation logique procède en plusieurs étapes. Tout d'abord, il est nécessaire de construire les arguments à partir des connaissances de la base. Ainsi, les prémisses de chaque argument sont un ensemble minimal de formules cohérentes issues de la base et qui permettent de prouver par déduction la conclusion de cet argument. Une fois que les arguments sont construits, il est possible de calculer leurs interactions (les attaques) grâce aux incohérences entre les connaissances desquelles ils sont issus. Les arguments et attaques obtenus peuvent alors être étudiés grâce au cadre de Dung, notamment pour déterminer les extensions et le statut des arguments.

Décision

L'argumentation, telle que nous l'avons vu jusque-là consiste à fournir un ensemble de conclusions cohérentes à partir de l'évaluation des interactions entre des arguments. Dans le cadre de la *décision* (Bonet et Geffner (1996); Fox et Parsons (1997); Gordon et Karacapilidis (1997); Fox et McBurney (2002); Amgoud *et al.* (2006); Amgoud et Prade (2009)), elle consiste plutôt à déterminer le ou les meilleurs choix (qui sont souvent des actions à entreprendre) possibles parmi différentes options pour satisfaire un ensemble de buts. L'argumentation permet ainsi de départager ces options suivant les informations, éventuellement incomplètes ou incertaines, que l'on a sur leurs conséquences et sur le monde dans lequel elles sont considérées. Les arguments, dits *pratiques* ("practical") lorsqu'ils concernent un but, ont donc ici un double rôle, à savoir d'une part permettre la sélection d'une ou plusieurs alternatives et d'autre part de justifier les raisons qui ont porté à faire ce choix.

Interaction

De par sa nature, l'argumentation se révèle naturellement adaptée à l'échange d'arguments entre plusieurs protagonistes. Suite à la catégorisation de différents types de dialogues (Walton et Krabbe (1995)), de nombreux travaux ont visé à donner la possibilité à des programmes d'interagir entre eux dans différents contextes. En particulier, quatre types de dialogue ont été étudiés à travers l'argumentation logique, à savoir les dialogues de persuasion, de négociation, d'enquête et de délibération, en général grâce à des protocoles.

- Le dialogue de persuasion (Amgoud *et al.* (2000a); Prakken (2006); Hadjinikolis *et al.* (2013)) permet à deux (ou plus) agents de résoudre une différence d'opinion en tentant de persuader l'autre (ou les autres) agents.
- Le dialogue de négociation (Parsons et Jennings (1996); Kraus *et al.* (1998); Amgoud *et al.* (2000b); Amgoud et Prade (2004); Kakas et Moraitis (2006); Amgoud et Vesic (2012)) vise à résoudre un conflit d'intérêt entre plusieurs agents en atteignant un accord satisfaisant tous les partis.
- Le dialogue d'enquête (Parsons *et al.* (2003); Black et Hunter (2007)) a pour but de prouver qu'une information particulière est effectivement vraie ou fausse grâce à un échange d'arguments entre plusieurs agents.
- Le dialogue de délibération (Hitchcock *et al.* (2001); Black et Atkinson (2011); Kok *et al.* (2011), enfin, a pour tâche d'évaluer et de choisir la meilleure solution possible, en termes d'actions à entreprendre, pour solutionner un problème.

2.3.1.2 Extensions du cadre de Dung

Le cadre de Dung a donné lieu à de nombreux travaux. Certains ont tenté d'étendre les notions qu'il permet de modéliser. Dans cette section, nous nous penchons sur les extensions classiques de Dung (1995).

Enrichissement des interactions

Support et bipolarité Dans les travaux que nous avons présentés jusque-là, en particulier le cadre originel de Dung, un seul type de relation binaire est considéré : l'attaque. Si cette interaction est primordiale pour l'argumentation, car elle permet d'exprimer le rapport *négatif* entre deux arguments, il en existe une autre : le *support*. Cette relation, à l'inverse de l'attaque, exprime le rapport *positif* entre deux arguments, c'est-à-dire l'"aide" qu'apporte un argument à un autre argument, ou encore, une raison de croire en sa conclusion. La relation d'attaque permet d'exprimer indirectement cette aide avec le mécanisme de défense ; on parle de défense lorsqu'un argument attaque un attaquant d'un autre argument. Considérer une relation de support explicite donne directement la possibilité de représenter ce lien positif indépendamment de tout autre argument.

Les cadres formels tenant compte à la fois des relations de support et d'attaque sont dits *bipolaires* (Karacapilidis et Papadias (2001); Verheij (2003); Amgoud *et al.* (2008); Cayrol et Lagasquie-Schiex (2009, 2010); Boella *et al.* (2010); Cayrol et Lagasquie-Schiex (2011, 2013)). Ils étendent l'expressivité du cadre de Dung en permettant la définition de nouvelles façons de calculer le statut des arguments, notamment en évaluant le nombre d'attaques par rapport au nombre de supports que reçoit un argument.

Préférence et pondération Intuitivement, il paraît sensé de penser que certains arguments ont plus d'impact que d'autres sur une personne. Par exemple, si un argument (avancé par une personne pour laquelle on n'a aucune confiance) attaque un autre argument (avancé, lui, par une personne en qui on a toute confiance), il peut nous arriver de ne pas tenir compte de cette attaque et d'accepter l'argument attaqué malgré tout. En quelque sorte, les arguments sont de qualité différente, ce qui peut dépendre de la personne avançant l'argument, de la personne le recevant, de la certitude des informations associées à l'argument, etc. Deux types de travaux étudient cette question, les systèmes d'argumentation préférentiels et la pondération des attaques.

Les systèmes d'argumentation préférentiels (Amgoud *et al.* (1996); Amgoud et Cayrol (2002a,b); Modgil (2009)) utilisent une relation de préférence[6] portant sur les arguments. De manière générale, dans ces travaux, les préférences permettent de moduler les conclusions d'un système d'argumentation, en particulier pour "protéger" les arguments préférés des autres arguments (c'est-à-dire, les accepter même s'ils sont attaqués par des arguments ayant moins d'impact).

La pondération des attaques (Martínez *et al.* (2007); Dunne *et al.* (2009); Cayrol *et al.* (2010a,b); Coste-Marquis *et al.* (2012)) consiste à donner à toutes les attaques d'un système un poids représentant sa force ou l'importance de l'argument duquel

6. Le lecteur pourra se reporter à la section B page 195 pour un rappel sur les relations de préférence.

elle est issue. Il est alors possible, de façon similaire aux systèmes d'argumentation préférentiels, de ne pas tenir compte des attaques jugées trop faibles. Ceci permet de retourner dans un système classique de Dung pour raisonner et donne une piste intéressante pour étudier l'opération de suppression d'attaque.

Enrichissement de la notion d'acceptabilité La notion d'acceptabilité telle qu'elle est définie dans le cadre de Dung prend obligatoirement en compte des "collectifs" d'arguments (des ensembles qui doivent par exemple être cohérents, qui se défendent collectivement face aux autres arguments, etc.). Cela ne permet pas d'établir des conditions d'acceptation précises pour chaque argument. Ceci est problématique lorsque l'on traite la question du *standard de preuve*[7], c'est-à-dire le "degré de certitude" qu'un argument doit atteindre pour être accepté.

Ce problème est étudié par les travaux sur les *cadres abstraits dialectiques* ("abstract dialectical frameworks", Brewka et Woltran (2010); Brewka *et al.* (2011, 2013)). Dans ces travaux, le statut de chaque argument est défini par une *condition d'acceptation* lui étant associée qui prend en compte le statut des "arguments parents" (les arguments ayant une interaction vers cet argument). Avec un tel cadre, il est donc possible de spécifier précisément et individuellement dans quels cas un argument est accepté.

Notons que ces conditions d'acceptation, du fait de leur généralité, permettent de traiter d'autres questions que le standard de preuve. Elles peuvent en particulier capturer les supports entre argument. De plus, il est possible d'introduire aisément une pondération sur les interactions et donc de capturer la notion de préférence.

Contrainte logique sur les extensions

Nous avons vu plus haut que la logique peut être utilisée pour instancier le cadre de Dung, notamment pour la construction des arguments. Certains travaux (Coste-Marquis *et al.* (2006); Amgoud *et al.* (2008); Devred *et al.* (2010)) étudient comment elle peut également être utilisée pour contraindre les extensions. Plus précisément, ces contraintes indiquent, au moyen d'un formule propositionnelle sur l'ensemble des arguments, les conditions rendant une extension valable si elles sont satisfaites. Ceci permet d'exprimer des besoins qui ne sont pas accessibles au cadre de Dung originel. Il est par exemple possible d'indiquer que toute extension contenant un argument particulier doit obligatoirement en contenir un autre ou que si une extension contient un argument particulier, alors elle ne doit pas en contenir un autre, sauf si un troisième est présent, etc. De fait, une telle approche est une généralisation du cadre de Dung.

7. Cette notion, peu usitée dans le droit français, est importante dans la justice anglo-saxonne ("common law" ; voir Fluet (2011)). On trouve notamment des standards tels que la *prépondérance de preuve* ("preponderance of the evidence", où aucune preuve ayant une probabilité de moins de 50% ne peut être acceptée), ou encore l'*au-delà du doute raisonnable* ("beyond reasonable doubt", où les preuves ne doivent laisser aucun doute à une personne "raisonnable").

Nouvelles sémantiques

Si elles ne sont pas à proprement parler des extensions du cadre de Dung, plusieurs nouvelles sémantiques ont été introduites. Celles-ci traduisent de nouveaux besoins en termes de calculs d'ensembles d'arguments acceptables. Deux grandes familles de sémantiques existent, les sémantiques basées sur les extensions et les sémantiques basées sur les labels. Les premières, à l'instar de celles définies par Dung (1995) (préférée, stable et basique), définissent et calculent des ensembles d'arguments acceptables qui sont généralement sans conflits et capables de se défendre collectivement face aux autres arguments. Les secondes abordent les extensions d'un autre angle : tout d'abord, chaque argument reçoit un label suivant une fonction de label particulière, puis les extensions sont calculées en fonction de ces labels. En général, trois labels sont considérés, à savoir *Acc*, statuant que l'argument est accepté, *Rej*, signifiant que l'argument est rejeté et *Ind* dans le cas où son statut est inconnu (car à la fois accepté et rejeté).

Notons que le lien entre ces deux familles est fort ; Caminada (2006) a montré que les sémantiques de Dung (1995) peuvent être redéfinies en usant des fonctions de label.

Sémantiques basées sur les extensions

- La sémantique semi-stable, initialement proposée dans Verheij (1996) puis reprise dans Caminada (2006), est une version plus flexible de la sémantique stable. En effet, cette sémantique permet de calculer un ensemble admissible d'arguments attaquant *le plus possible d'arguments* ne faisant pas partie de l'ensemble.

- La sémantique idéale de Dung *et al.* (2007) consiste à sélectionner un seul et unique ensemble admissible d'arguments contenu dans toutes les extensions préférées. Elle est en ce sens proche de la sémantique basique tout en étant moins contraignante.

- La sémantique récursive de Baroni *et al.* (2005) n'est pas à proprement parler une sémantique, mais plutôt un schéma général de calcul d'extensions se basant sur une procédure de décomposition du graphe d'argumentation en composantes fortement connexes. Ce schéma capture les sémantiques définies dans Dung (1995) tout en permettant de définir de nouvelles sémantiques ; il est à ce titre plus général que ces dernières.

- La sémantique prudente de Coste-Marquis *et al.* (2005) a pour principal objectif de permettre une meilleure gestion des arguments controversés, c'est-à-dire les arguments qui à la fois attaquent et défendent (indirectement) un autre argument. Pour éviter qu'un argument controversé se retrouve dans la même extension que l'argument qu'il attaque et défend à la fois, cette sémantique interdit à toute extension de contenir deux arguments tels que l'un attaque indirectement l'autre.

Sémantiques basées sur les labels

- La sémantique robuste de Jakobovits et Vermeir (1999) tente d'assurer la stabilité des arguments *décidés* (les arguments ayant le label *Acc* ou *Rej*). Plus précisément, parmi les différents ensembles de labels possibles, ne sont gardés que ceux dont les arguments décidés ne changent pas de label si l'on modifie un argument étiqueté *Ind*.
- La sémantique stage de Verheij (1996) se base sur la notion de *stage* qui est une paire constituée d'un ensemble d'arguments acceptés et d'un ensemble d'arguments rejetés ; un stage impose le fait qu'un argument est rejeté si et seulement s'il est attaqué par un argument accepté. Une extension stage est un stage maximal.
- La sémantique à base de labels conditionnels Booth *et al.* (2012) généralise la notion de fonction d'acceptation pour y inclure des conditions permettant de contraindre l'évaluation du statut des arguments. La fonction d'acceptation renvoie donc des ensembles de labels parmi ceux fournis par la condition donnée en entrée.

Notons que d'autres approches sortent du cadre habituel des sémantiques. En particulier, Villata *et al.* (2011) considère le statut des attaques plutôt que celui des arguments. Plus précisément, ce travail propose plusieurs sémantiques se basant sur la distinction entre les attaques réussies et les attaques échouées ; un argument est alors accepté lorsqu'il n'y a aucune attaque réussie le prenant pour cible.

Par ailleurs, certaines sémantiques ont été introduites pour tenir compte des divers enrichissements du cadre de Dung. Notamment, Brewka et Woltran (2010) introduit la notion de *modèle* et adapte grâce à elle plusieurs des sémantiques définies dans Dung (1995) aux cadres abstraits dialectiques. De la même façon, Cayrol *et al.* (2010a) et Coste-Marquis *et al.* (2012) adaptent, chacun à leurs façons, certaines sémantiques de façon à prendre en compte la pondération des attaques, notamment en revisitant la notion de défense.

2.3.2 Traitement de la dynamique

La dynamique en argumentation, bien que peu traitée pour le moment, commence à prendre de l'ampleur. Nous pouvons citer trois catégories de travaux.

- L'étude des modifications que va subir le système d'argumentation, en particulier en ce qui concerne ses extensions. Les travaux de Boella *et al.* (2009a,b,c) rentrent dans cette catégorie ; ils sont commentés dans la section 5.6 page 103.
- L'étude des modifications à faire dans un système d'argumentation pour s'assurer qu'un ou plusieurs arguments soient acceptés après le changement. Moguillansky *et al.* (2010), Baumann et Brewka (2010); Baumann (2012a,b), Coste-Marquis *et al.* (2013) et Booth *et al.* (2013) ont cet objectif ; ils sont présentés dans les sections 6.3 page 131 et 7.5 page 149.

- L'étude, d'un point de vue pratique, de la possibilité de calculer efficacement les modifications impliquées par un changement. Cette question est notamment traitée par Liao *et al.* (2011) et Kontarinis *et al.* (2013), que nous étudions dans la section 8.3 page 165.

Pour analyser la dynamique en argumentation, notre choix s'est porté sur l'utilisation du travail de Cayrol *et al.* (2010c) et donc du cadre de Dung. Le caractère abstrait de ce cadre, qui en fait sa force, pourrait laisser penser qu'il est cantonné à un rôle purement théorique ; en d'autres termes, il pourrait n'avoir d'utilité que dans une démarche d'étude purement artificielle, notamment du fait de ses notions de base qui ne semblent pas adaptées à la réalité. À ce titre, peu de travaux se sont penchés sur son intérêt dans des cadres concrets et, par conséquent, il existe à notre connaissance très peu de justifications de son utilisation effective pour analyser des situations pratiques.

Nous pensons au contraire que la généralité du cadre de Dung n'est en rien un frein à son utilisation dans des cas réels. Dans le chapitre suivant, nous montrons cela en proposant trois exemples que nous analysons grâce au cadre de Dung et au travail de Cayrol *et al.* (2010c) et nous apportons un éclairage nouveau sur les notions théoriques de l'argumentation abstraite.

Chapitre 3

Justifications de l'étude

Rien, en effet, n'exige plus d'effort de pensée que l'argumentation destinée à justifier la non-pensée.

Milan Kundera,
L'Immortalité.

Sommaire

Le sujet principal de ce travail est la dynamique en argumentation. Intuitivement, nous savons à quoi correspond cette vaste notion ; le déroulement d'un dialogue en est un exemple. Mais nous savons moins à quoi elle peut correspondre dans le cadre de l'argumentation abstraite. Le lecteur, après avoir parcouru les définitions de la section 2.2 page 16, est en droit de se demander ce qu'elles signifient, ce qu'elles racontent à propos de nos échanges d'arguments ; plus prosaïquement, il pourrait se demander *à quoi servent-elles (dans la vraie vie) ?* Il semble utile de proposer quelques éléments de réponse en faisant le lien entre la dynamique argumentative théorique et la dynamique argumentative concrète. Il semble précieux, même, de montrer que ces notions, que l'on imaginerait aux antipodes de l'interaction humaine, peuvent représenter des concepts qu'il est commun de rencontrer dans notre quotidien.

Aussi, il nous paraît crucial, avant de rentrer dans le vif du sujet, c'est-à-dire l'étude formelle de la dynamique en argumentation abstraite, de nous attarder quelques minutes sur l'idée, l'intuition, de ce que nous tenterons de construire au fil de ce travail.

Mais pour expliquer ce qu'est la dynamique en argumentation abstraite, notre sujet, nous ne pouvons nous passer d'expliquer ce qu'est l'argumentation abstraite sur laquelle elle se base. Après tout, les définitions de la section 2.1 page 12 sont elles aussi loin de l'idée que l'on peut se faire de l'argumentation telle que nous la connaissons.

Ainsi, dans cette section, nous nous efforçons à travers plusieurs exemples,

- de justifier l'utilité du cadre abstrait que nous utilisons en donnant des intuitions permettant d'expliquer les notions théoriques,
- et de montrer des cas d'utilisation motivant l'introduction d'un cadre particulier d'échanges d'arguments entre plusieurs protagonistes.

Notamment, le premier exemple est une application simple et qui se prête bien à la notion d'argumentation : une audience de tribunal. Le deuxième, proche du premier, propose la formalisation d'une discussion entre plusieurs personnes tentant de prendre une décision collective. Bien entendu, si ces exemples permettent d'illustrer aisément les notions mises en jeu, car étant par essence argumentatifs, ils ne résument pas à eux seuls le cadre applicatif de l'argumentation abstraite. Ainsi, nous proposons un troisième exemple et contexte où l'argumentation abstraite pourrait se révéler intéressante : la modélisation de l'allocation de ressources.

Nous présentons ces exemples respectivement dans les sections 3.1, 3.2 page 39 et 3.3 page 47. De façon à bien distinguer l'apport de l'argumentation abstraite et celui de sa composante dynamique, nous divisons chacune de ces sections en deux parties. La première partie procède à l'analyse “statique” de l'exemple, où l'on introduit la situation et s'intéresse à l'interprétation des notions classiques de l'argumentation abstraite. La deuxième partie propose, elle, de présenter comment la dynamique peut surgir dans un tel contexte et quel est son intérêt. Enfin, nous donnons dans la section 3.4 une liste d'applications potentielles triées par opération de changement.

Notons que l'annexe D.2 page 225 propose quelques intuitions supplémentaires.

3.1 Audience de tribunal

Notre première application, inspirée librement de Brewka (2001), est celle qui se rapproche le plus de l'idée que l'on peut se faire d'une argumentation formelle. L'intuition concrète qui en découle est donc (en général) aisément saisissable. À ce titre, elle constituera le cadre de la majorité de nos exemples dans la suite de ce travail.

3.1.1 Argumentation abstraite dans le cadre de l'audience

Cette section est présentée en deux parties, la première se concentrant sur les protagonistes et leurs motivations, la seconde traduisant les notions théoriques présentées dans la section 2.1 page 12 avec des intuitions adaptées au contexte d'une audience de tribunal.

3.1.1.1 Protagonistes

Cet exemple s'inspire du fonctionnement d'une audience de tribunal mettant en scène quatre entités aux rôles bien distincts qui interagissent pour déterminer si un argument x_0 est acceptable. Puisque nous nous plaçons dans un contexte respectant la présomption d'innocence, l'argument x_0 représente le fait que l'inculpé n'est pas coupable du crime dont il est accusé.

- Le **procureur** veut faire rejeter l'argument x_0. Il a à sa disposition un ensemble d'arguments, éventuellement différents de ceux de son adversaire (l'avocat).
- L'**avocat de la défense**, disposant également d'un ensemble d'arguments, tente de faire accepter l'argument x_0.
- Le **juge** s'assure que le processus d'argumentation s'opère dans de bonnes conditions. Il intervient lorsqu'une objection est faite par un des participants ; il peut alors accepter cette objection (donc faire retirer l'argument correspondant), ou la rejeter.
- Les **jurés** [1] ont le mot de la fin. Leur rôle est d'écouter les arguments du procureur et de l'avocat et d'en tirer une conclusion concernant l'acceptabilité de l'argument x_0. Ils n'interviennent pas durant les échanges entre le procureur, l'avocat et le juge. Lorsque l'audience est terminée (*i.e.* lorsque ni le procureur, ni l'avocat ne peuvent, ou ne veulent, donner de nouveaux arguments), les jurés peuvent délibérer et déterminer si l'argument x_0 est acceptable ou non.

L'audience de tribunal voyant s'affronter le procureur et l'avocat concerne le prévenu M. X. De multiples arguments peuvent être mis en jeu pour déterminer la culpabilité de ce dernier. La table 3.1 page suivante présente les divers arguments qui peuvent être avancés durant cette audience

Dans cet exemple, le procureur et l'avocat ne sont pas intéressés par le fait de se convaincre mutuellement. Par contre, ils souhaitent convaincre les jurés qui vont prendre la décision finale. Les deux protagonistes vont donc agir sur un système d'argumentation cible représentant l'état des connaissances des jurés, ce dernier étant vide au départ. Ce système représente le lieu central d'échanges entre le procureur et l'avocat, où chacun de ces derniers va placer à tour de rôle des arguments, de sorte qu'à la fin les jurés penchent en sa faveur.

1. Bien qu'au pluriel, les jurés sont une seule et même entité décisionnaire.

	Argument
x_0	*M. X n'est pas coupable d'homicide volontaire avec préméditation sur la personne de M^{me} X, sa femme.*
x_1	*M. X est coupable d'homicide volontaire avec préméditation sur la personne de M^{me} X, sa femme.*
x_2	*L'accusé a un alibi, sa secrétaire ayant juré sur l'honneur qu'elle l'avait vu à l'heure de crime.*
x_3	*La personnalité effacée et influençable de la secrétaire ne peut que mettre en doute la véracité de ses propos.*
x_4	*M. X aime si fort sa femme qu'il l'a demandée en mariage une deuxième fois. Or, un homme qui aime sa femme ne saurait en être le meurtrier.*
x_5	*M. X a été retrouvé avec beaucoup de sang appartenant à sa femme sur ses vêtements ; une personne dans cette situation est soit un meurtrier soit un idiot, or, M. X est loin d'être idiot puisqu'il est docteur ès sciences politiques.*
x_6	*L'accusé n'aurait eu aucun intérêt à tuer sa femme, puisqu'il n'était pas le bénéficiaire de l'énorme assurance vie contractée par celle-ci.*
x_7	*L'accusé est un homme connu pour être vénal et son "amour" pour une femme très riche ne pourrait être qu'appât du gain.*

Table 3.1 – *Arguments mis en jeu lors de l'audience au tribunal.*

3.1.1.2 Interprétation des notions d'argumentation abstraite

Comme nous l'avons vu plus haut, pris à un niveau basique, les buts des participants sont assez évidents : le procureur, voulant que l'accusé soit condamné, doit faire en sorte que l'argument x_0 soit rejeté lors de la délibération ; l'avocat, souhaitant l'inverse, doit faire en sorte que l'argument x_0 soit accepté. Ceci étant dit, il est possible d'enrichir ces notions de victoire et défaite grâce aux sémantiques d'acceptabilité et au statut de x_0 (*i.e.* son appartenance à une ou plusieurs extensions).

Interprétation des notions de bases de l'argumentation abstraite

Étudions tout d'abord ce que représentent les notions théoriques d'arguments abstraits, d'attaques et de défense dans le contexte de l'audience de tribunal.

Argument Dans le contexte d'une audience de tribunal, les arguments abstraits sont proches de ce qu'il est commun de se représenter, c'est-à-dire des raisonnements destinés à appuyer ou contredire une opinion. Néanmoins, l'argumentation abstraite permet de ne pas se préoccuper de leur nature exacte. En effet, un argument peut être, entre autres choses, une phrase ou plusieurs, un syllogisme, une démonstration, voire un monologue. L'argumentation abstraite simplifie donc la manipulation de la notion d'argument en le considérant comme une entité reflétant une certaine idée à communiquer.

Attaque De manière générale, une attaque représente le fait que deux arguments sont en conflit. Mais, de la même façon que pour les arguments, il existe plusieurs types d'attaques. Il se peut, par exemple, que la conclusion d'une longue phrase vienne entacher la légitimité d'une des hypothèses de la démonstration adverse. Il se peut tout aussi bien que les conclusions de deux syllogismes soient inconciliables.

Dans le cadre de l'argumentation abstraite, ces différents types sont ignorés et regroupés sous la forme d'une seule et unique forme d'attaque. Elle représente le sentiment que ce qui vient d'être dit est incompatible avec ce qui a été dit précédemment, que les énoncés ne peuvent coexister dans une vision cohérente du monde et qu'en accepter une partie implique en rejeter une autre.

Défense La défense représente le fait qu'un argument peut être réinstauré, c'est-à-dire qu'il peut être accepté quand bien même il serait attaqué. En effet, l'attaquant d'un argument a peut lui-même être attaqué par un autre argument c. Les points amenés par l'attaquant ne sont alors plus valables (car contestés par l'argument c) et, par voie de conséquence, n'ont plus l'opportunité de contredire l'argument a. Ce dernier se voit, de fait, justifié grâce à la défense de l'argument c.

Interprétation des sémantiques d'acceptabilité

Les sémantiques d'acceptabilité permettent de déterminer les ensembles d'arguments acceptables. Comme nous l'avons vu dans le chapitre 2 page 11, une sémantique se définit par un ensemble de critères qu'un ensemble d'arguments doit satisfaire pour être accepté. Ici, nous nous intéresserons à quatre sémantiques différentes, et nous discuterons de leur intérêt dans le cadre de notre exemple.

Sémantique basique La sémantique basique correspond à un calcul de défense sans faille. En effet, pour qu'un argument soit accepté dans le cadre de cette sémantique, *il faut qu'il soit défendu (éventuellement indirectement) par un argument n'étant pas attaqué*. La sémantique basique fournit donc un seul ensemble "solide" d'arguments acceptables, reposant sur des arguments inattaqués et donc, en quelque sorte, acceptés de manière consensuelle par tous les participants. Ces derniers pourraient être vus comme des évidences ou des faits.

Le "revers de la médaille" d'une telle solidité est que la sémantique basique ne conduit pas toujours à un ensemble non vide d'arguments acceptables. En effet, tout argument qui n'est pas solidement appuyé (c'est-à-dire qui est attaqué et non défendu), donc sur lequel le doute est permis, ne peut pas être pris en considération pour la délibération. On pourrait y voir une lourde *charge de preuve*[2], seyant bien à la question de la culpabilité d'une personne, chose particulièrement délicate. Cette sémantique peut être souhaitable pour l'avocat puisqu'elle obligerait son adversaire, lequel ayant la charge de la preuve concernant la culpabilité de l'accusé, à fournir une argumentation inébranlable, c'est-à-dire qui ne laisse aucun doute aux jurés. Néanmoins, cette sémantique est à double tranchant, puisque si le procureur parvient à prouver la culpabilité de l'accusé, il sera plus difficile pour l'avocat de contrer son argumentation.

Sémantique préférée La sémantique préférée, quant à elle, correspond à un calcul d'*ensembles d'arguments capables de se défendre collectivement*. Elle permet de ce fait de fournir les différents "points de vue" pouvant coexister à l'issue de l'échange d'arguments. Cette sémantique peut être vue comme plus "flexible", ou plus laxiste, que la précédente en cela qu'un argument peut être considéré valable dès l'appartenance à au moins un point de vue, et que ce dernier n'a donc pas forcément besoin d'être aussi solidement soutenu.

La conséquence immédiate pour notre exemple est de permettre la considération d'une charge de preuve plus faible et d'introduire une certaine notion de *bénéfice du doute*. Ce dernier point est particulièrement pertinent pour une audience de tribunal. En effet, en vertu du principe de présomption d'innocence, il est important qu'un accusé ne soit pas reconnu coupable s'il n'existe pas de preuves patentes. Dans le cadre de l'exemple de l'audience de tribunal, le choix d'une telle sémantique par les jurés s'avèrerait à l'avantage de l'avocat de la défense qui, s'il lui est impossible de prouver purement et simplement l'innocence de son client, peut tenter de montrer que les preuves ne permettent pas d'être sûr de sa culpabilité, c'est-à-dire qu'il existe au moins un point de vue où l'accusé n'est pas coupable.

Sémantique stable Toute extension stable étant préférée (nous l'avons vu dans la proposition 2.1 page 14), on peut dire que la sémantique stable respecte ce qui a été dit plus haut concernant la sémantique préférée. Cependant, la sémantique stable exige un autre critère, et non des moindres : un ensemble d'arguments dit stable doit *défaire tout argument ne faisant pas partie de cet ensemble*.

À première vue, ce critère peut sembler inutile, voire inopportun, par rapport à la sémantique préférée. Après tout, qu'un ensemble d'arguments attaque ou n'attaque pas tous les "autres" arguments n'impacte pas la cohérence de ce même ensemble. Mais pris dans le contexte d'un argumentaire, c'est une indication intéressante sur ce qui a été dit. Plus précisément, un ensemble d'arguments stable attaque tout argument n'en faisant pas partie et répond donc à ces derniers. Ainsi, si l'argumentation d'un protagoniste forme un ensemble d'arguments stable, elle répond alors à

2. "*Il incombe à chaque partie de prouver conformément à la loi les faits nécessaires au succès de sa prétention*" (Article 9 du *Code de procédure civile française*). La charge de la preuve est le principe selon lequel il incombe à une partie de prouver un fait. En général, c'est à la partie demandeuse que revient la charge de la preuve.

tous les arguments qu'il n'a pas avancés lui-même. Ce critère, relevant autant de la manière que du fond, pourrait à terme permettre d'évaluer la "performance" d'un orateur au-delà de la simple victoire.

Malheureusement, remarquons qu'il peut n'y avoir aucun ensemble stable d'arguments, signifiant alors qu'aucun argumentaire n'a été capable de répondre à tous les arguments extérieurs à lui. Pour pallier ce problème, il est possible d'utiliser la sémantique semi-stable.

Sémantique semi-stable

Bien que notre propos ne soit pas de les traiter toutes, notons l'existence d'autres sémantiques intéressantes dans le cadre de cet exemple. Citons notamment la sémantique semi-stable qui, comme son nom pourrait l'indiquer, constitue un palliatif de la sémantique stable. Cette sémantique privilégie les *ensembles d'arguments capables de se défendre collectivement pour lesquels l'union de l'ensemble et des arguments qu'il attaque est maximale.*

Cette approche intermédiaire entre sémantique préférée et sémantique stable permet d'avoir à coup sûr au moins un ensemble d'arguments qui attaque le plus possible d'arguments extérieurs. Ainsi, si l'argumentation d'un protagoniste se trouve être un ensemble d'arguments semi-stable, cette dernière est la plus "performante" possible, en cela qu'elle repose sur des arguments se défendant face à ceux de son adversaire, et répond au plus grand nombre des arguments de celui-ci. Une stratégie basée sur la sémantique semi-stable est intéressante, car un protagoniste dont la plupart des arguments se font contrer par ceux de son adversaire peut perdre la considération des jurés et la confiance qu'ils ont en ce qu'il dit.

Interprétation du statut d'un argument

Nous avons vu plus haut différentes sémantiques, et ce à quoi elles peuvent renvoyer dans le contexte de notre exemple. Ces sémantiques permettent de calculer le statut de chacun des arguments. Nous nous intéressons donc ici à interpréter cette notion de statut dans le cadre de notre exemple. Pour ce faire, décrivons chacun des trois statuts possibles, à savoir *accepté sceptiquement*, *accepté crédulement* et *rejeté*.

Accepté sceptiquement Sous une sémantique donnée, un argument dit accepté sceptiquement est un argument appartenant à toutes les extensions de la sémantique. Pour reprendre ce que nous avons dit plus tôt, c'est un argument qui fait partie de tous les points de vue s'affrontant. Il n'y a donc aucune hésitation sur la validité de l'argument pour les acteurs du dialogue. En définitive, c'est une acceptation qui ne laisse place à aucun doute, qui permet une approbation "universelle" de l'argument.

Par exemple, étant donné la gravité d'un tel sujet, il est important qu'un argument représentant la culpabilité d'un accusé soit accepté de manière sceptique.

Accepté crédulement Un argument dit accepté crédulement est un argument appartenant à au moins une extension de la sémantique donnée. Un tel statut permet de décréter valable un argument même si celui-ci est sous le coup du doute, ou ne présente pas une preuve suffisante.

Ainsi, un argument représentant l'innocence d'un accusé ne doit être rejeté qu'à la condition qu'on puisse prouver *sans aucun doute possible* qu'il est coupable (sa culpabilité doit être acceptée de manière sceptique). Par opposition, il suffira de quelques doutes sur sa culpabilité pour que son innocence soit acceptée (son innocence peut être acceptée de manière crédule). Ceci peut être résumé dans notre exemple par le fait que l'avocat, qui est intéressé par une acceptation sceptique de l'argument représentant l'innocence de l'accusé peut, s'il n'y arrive pas, se rabattre sur son acceptation crédule.

Notons que dans le cas particulier des sémantiques n'ayant qu'une seule extension, telles que la sémantique basique, les statuts *accepté sceptiquement* et *accepté crédulement* se confondent pour donner naissance au simple statut *accepté*, traduisant le fait que l'argument appartient à la seule et unique extension.

Rejeté Un argument dit rejeté n'appartient à aucune extension de la sémantique donnée. Ainsi, quels que soient les points de vue, cet argument n'en fait pas partie. Ce dernier est donc unanimement considéré comme non valable par les participants.

Dans notre exemple, le procureur tentera de faire rejeter l'argument représentant l'innocence de l'accusé.

La section 3.1.2 présente un protocole gérant le dialogue entre le procureur et l'avocat.

3.1.2 Protocole de l'audience

Nous présentons ici un protocole gérant l'échange d'arguments dans l'exemple de l'audience de tribunal.

1. Tout d'abord, l'argument x_0 de l'audience est mis en place. Par convention, c'est l'avocat de la défense qui prend la parole en premier et énonce l'innocence de son client. Cette action prend pour cible le système d'argumentation des jurés, qui est modifié.
2. S'ensuit une séquence d'actions proposées par les différents agents (soit le procureur, soit l'avocat) en fonction de leur propre système d'argumentation. Deux types d'actions sont possibles :
   - Les *actions additives*, c'est-à-dire l'addition d'un ou plusieurs arguments et/ou attaques directement dans le système d'argumentation des jurés.
   - Les *actions suppressives*, c'est-à-dire le retrait d'un ou plusieurs arguments et/ou attaques du système d'argumentation des jurés. Cela peut par exemple se produire par le truchement d'une objection qui consiste à

demander au juge de statuer sur la légalité d'arguments et/ou d'attaques de l'adversaire. Si l'objection est rejetée, aucune modification du système des jurés n'est effectuée. Si l'objection est acceptée, les arguments ou attaques objectés sont retirés du système des jurés.

Il est aussi possible pour un agent de ne *rien faire*. Ceci peut notamment arriver lorsqu'un agent n'a plus d'argument à avancer, ou lorsque le système des jurés est dans un état lui convenant.

3. Un effet de bord des actions est le fait que si un des deux protagonistes n'a pas connaissance de l'argument que son adversaire énonce, il va l'ajouter à son propre système. De même, si un argument est jugé illégal suite à une objection acceptée par le juge, les deux agents doivent l'occulter afin de ne plus l'utiliser.
4. Lorsque les deux agents décident de ne plus rien faire, l'échange s'arrête et les jurés délibèrent grâce à leur système d'argumentation.

La figure 3.1 page suivante présente un court exemple illustrant ce protocole.

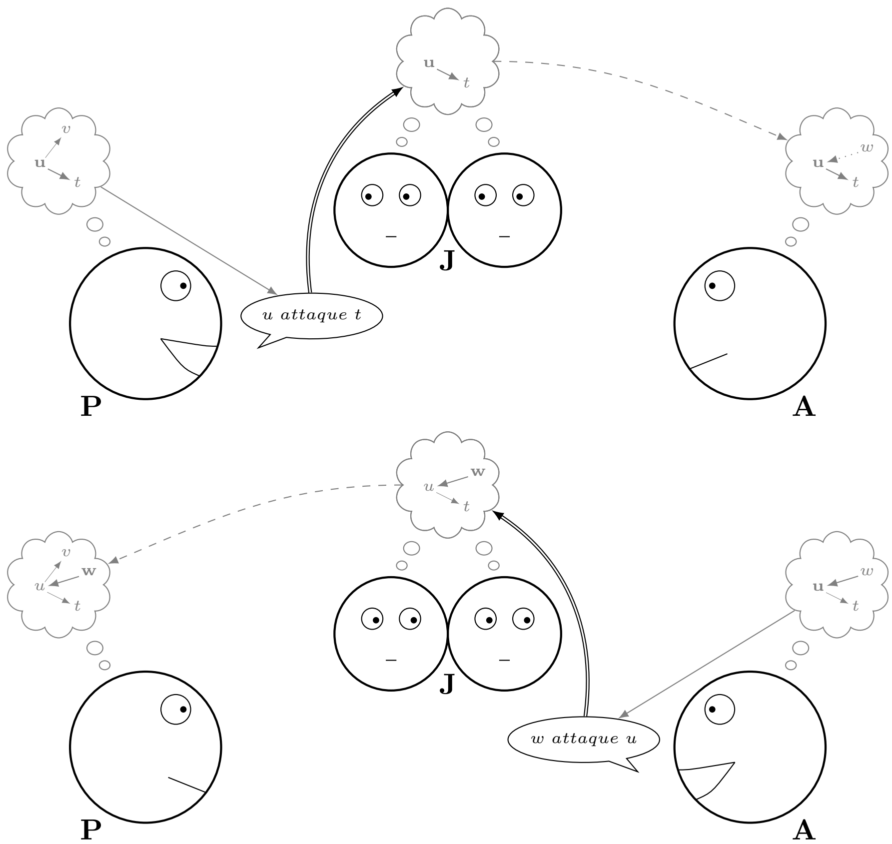


Figure 3.1 – *Illustration présentant un échange typique entre le procureur (**P**) et l'avocat (**A**) sous les yeux des jurés (**J**). On y voit tout d'abord la prise de parole du procureur, énonçant l'argument u et l'attaque de u vers t, ce qui modifie le système d'argumentation des jurés ; celui de l'avocat est ensuite mis à jour avec cette nouvelle information, ce qui lui permet de calculer l'attaque de w vers u. Cette nouvelle attaque est bien utile à l'avocat, qui s'empresse de l'énoncer (ainsi que w), modifiant à son tour le système des jurés ; le système du procureur est finalement mis à jour pour y inclure ces nouvelles informations.*

3.2 Dialogue pour l'organisation d'une soirée

Nous présentons maintenant une autre application concernant le choix de l'activité d'un groupe de personnes n'ayant pas nécessairement les mêmes envies. Cette deuxième application est une variante de la première. La différence principale est que l'auditoire n'est plus une entité déterminée comme les jurés, mais est constituée des orateurs eux-mêmes. À ce titre, les interprétations que l'on peut faire des notions théoriques de l'argumentation abstraite sont similaires à celles présentées précédemment ; nous ne les répèterons donc pas. Par ailleurs, l'application de l'audience mettait en scène un dialogue de *persuasion* (de l'auditoire), celle-ci tient plus de la *délibération* entre les protagonistes.[3]

3.2.1 Distribution des arguments et buts

Imaginons qu'un groupe de cinq amis (que nous nommons Rosa, Georges, Jean, Léo et Louise) se retrouvent et se mettent en tête de choisir une activité pour la soirée qui arrive. Trois choix s'offrent à eux : verser une larme devant un film d'auteur (nommé f_a), en prendre plein les yeux avec un film hollywoodien (nommé f_h), ou se sustenter au-delà des besoins recommandés par les diététiciens au restaurant. Ces choix sont représentés respectivement par les arguments x_1, x_2 et x_3.

Pour déterminer l'issue de la discussion, et donc l'activité pour laquelle les cinq amis opteront, ces derniers disposent d'arguments qu'ils vont échanger. Comme dans l'application de l'audience, ils ne connaissent pas tous les arguments, et ne les partagent pas nécessairement. La table 3.2 page suivante présente les différents arguments mis en jeu dans ce dialogue et la figure 3.2 le système d'argumentation en résultant.

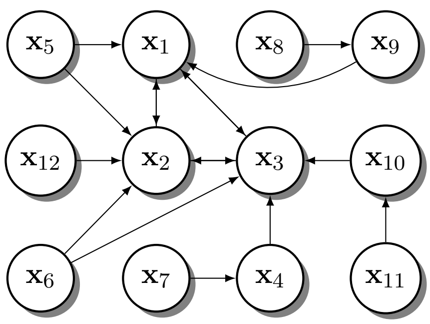


Figure 3.2 – *Système d'argumentation global représentant tous les arguments mis en jeu.*

Ces arguments sont distribués entre les différents agents, certains ayant accès à des informations et d'autres non. La figure 3.3 page 41 présente les systèmes d'argumentation correspondant à nos cinq agents. Notons sur cette figure la présence

3. Selon Walton (2003), si la persuasion implique un conflit d'opinion qu'il convient de résoudre, la délibération consiste à trouver la meilleure solution possible permettant de résorber un dilemme.

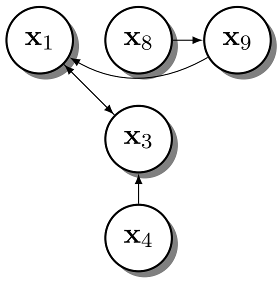
	Arguments
x_1	*f_a est le film que nous irons voir ce soir !*
x_2	*f_h est le film que nous irons voir ce soir !*
x_3	*Ce soir, c'est au restaurant que nous irons manger !*
x_4	*Le restaurant est trop cher.*
x_5	*Nous n'aurons pas le temps de manger si nous allons au cinéma (et les gourmandises disponibles là-bas ne sont que peu appropriées pour la ligne).*
x_6	*Aller voir f_a me permettrait de recroiser une amie.*
x_7	*Connaissant le patron du restaurant, nous pourrons avoir un tarif réduit.*
x_8	*Les acteurs de f_a ont déjà été primés dans leur carrière.*
x_9	*Les acteurs de f_a sont mauvais.*
x_{10}	*Le restaurant n'est pas végétarien, alors que nous le sommes tous.*
x_{11}	*Le restaurant sert des plats préparés spécifiquement pour les végétariens.*
x_{12}	*Le film f_h est en version française, et nous savons tous que c'est mal.*

Table 3.2 – *Arguments mis en jeu pour déterminer l'objet de la soirée.*

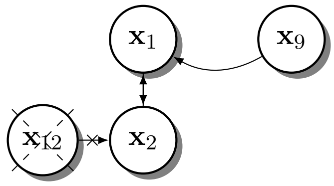
d'un argument barré dans le système de l'agent Georges ; ceci indique que Georges, bien que connaissant cet argument, ne l'utilisera pas. Ce procédé, que nous nommons *occultation*, traduit le fait qu'un argument peut avoir un impact inintéressant, voire négatif, pour l'agent le connaissant. Par exemple, un argument révélant une information très secrète (et n'étant pas destinée aux agents participant au dialogue) ou ayant une influence indésirable et ne sera pas avancé. Ici, on peut imaginer que Georges sait que le film hollywoodien a été doublé en français et que ses amis sont des cinéphiles puristes (et un peu snobs) n'aimant pas ça ; il préfère donc ne pas être totalement honnête et ne pas faire remarquer ce fait pour avoir une chance d'aller voir le film.

Par ailleurs, puisque cet exemple se place dans le cadre d'un dialogue de *délibération*, chaque agent possède un but pour la collectivité (ce qu'il souhaite que le groupe fasse de sa soirée) qu'il cherche à atteindre en argumentant. Étudions ces différents *desiderata* :

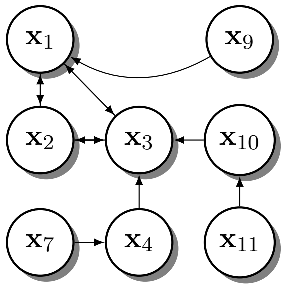
- Rosa, amateur d'Art et Essai s'il en est, *veut voir le film f_a*. Elle doit donc faire en sorte qu'il n'y ait qu'une seule option possible pour la soirée, c'est-à-dire une seule extension, et que celle-ci ne contienne que l'argument x_1.



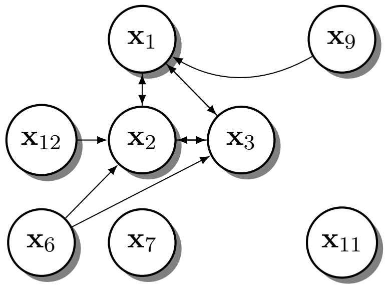
(a) *Rosa.*



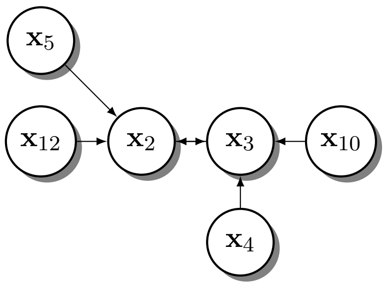
(b) *Georges.*



(c) *Jean.*



(d) *Léo.*



(e) *Louise.*

Figure 3.3 – *Systèmes d'argumentation des différents agents.*

- Georges, n'étant pas d'humeur à se confronter à la vacuité de l'existence par l'intermédiaire d'un film en noir et blanc, *préfère voir le film f_h*. Il a donc la même démarche que Rosa à la différence qu'il souhaite voir l'argument x_2 accepté.

- Jean, ne voulant pas mettre en péril son régime à base de poireaux, *veut aller au cinéma (peu lui importe le film)* pour éviter toute grasse tentation. Il souhaite donc que x_1 ou x_2 soit accepté, mais surtout pas x_3.

- Léo est de mauvaise humeur et fatigué. Même s'il n'ose pas l'avouer à ses amis, qui le traiteraient d'empêcheur de tourner en rond, il veut rentrer chez lui et *ne rien faire*. Il préfère donc agir de façon détournée et empêcher toute prise de décision (qu'il n'y ait aucune option possible, qu'il y en ait plusieurs, ou tout simplement qu'aucun des arguments x_1, x_2 ou x_3 ne soit accepté) jusqu'à ce que la soirée soit annulée.

- Louise s'en moque, tout ce qui lui importe c'est d'être avec ses amis. Par contre, elle a un peu froid et *voudrait qu'une décision soit prise* assez rapidement pour leur permettre de se mettre au chaud. Pour cela, il faut juste qu'une option possible se dessine.

Nous allons voir dans la section 3.2.2 comment ces buts vont aiguiller le choix des opérations à faire pour chacun des cinq amis.

3.2.2 Déroulement du dialogue

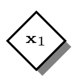
Nous avons vu dans la section 3.2.1 page 39 les différents systèmes d'argumentation des cinq amis ainsi que ce que chacun d'entre eux souhaiteraient faire de leur soirée. Nous allons maintenant étudier comment chacun d'entre eux va choisir une action en fonction de son système d'argumentation et de son but.

Nous supposons que les cinq amis, poliment organisés, parlent à tour de rôle. Ils utilisent donc, à quelques détails près, le même protocole que celui utilisé pour l'audience, c'est-à-dire, en quelques mots :

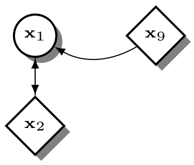
1. Mise en place des arguments représentant les choix possibles (x_1, x_2 et x_3).
2. Séquence d'arguments échangés par les cinq personnes avec des actions *additives* ou *suppressives* (il est également possible de ne rien faire). Notons qu'ici, nous l'avons dit, il n'y a pas de jurés, les participants ne modifient donc pas le système d'une personne en particulier mais, plutôt, un système d'argumentation partagé représentant le stade d'évolution du dialogue.
3. Mise à jour (lors de la séquence précédente) des systèmes d'argumentation des participants lorsqu'ils ne connaissent pas un argument énoncé ou lorsqu'un argument est jugé (collégialement) illégal.
4. Arrêt du dialogue lorsque les cinq participants décident de ne plus rien dire. Le choix de l'activité se fait alors grâce au système d'argumentation partagé.

Étudions donc au tour par tour un déroulement possible du dialogue entre les agents en utilisant le cadre de Dung. Pour chaque tour (hormis ceux où aucune opération n'est faite), une aide visuelle est proposée par une table correspondante. La table 3.3 page 47 présente le système d'argumentation final du dialogue. Notons par ailleurs que, dans cet exemple, nous n'utilisons que la sémantique préférée.

Tour 1 Rosa prend la parole en premier. Elle fait part à ses amis de ce qu'elle souhaiterait faire avec eux, c'est-à-dire aller voir f_a. Elle ajoute donc l'argument x_1 au système qui est de fait le seul accepté.

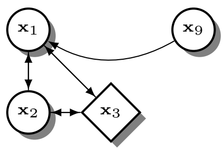
Tour	Opérations	État du dialogue
1 (Rosa)	$(\bigoplus, \{x_1\}, \varnothing)$	
	$\mathbf{E} = \{\{x_1\}\}$	

Tour 2 Georges, inquiet de passer une soirée à s'ennuyer, s'empresse de proposer et justifier un autre film, à savoir f_h, sous le prétexte que les acteurs de f_a ne lui plaisent pas. Il ajoute ainsi x_2 au système et les attaques entre ce dernier et x_1 (puisqu'ils sont incompatibles) ainsi que l'argument x_9 et une attaque de x_9 vers x_1. L'argument x_1 se retrouve alors rejeté, alors que x_2 devient accepté.

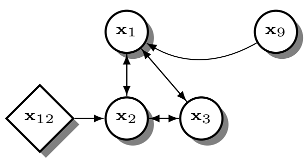
Tour	Opérations	État du dialogue
2 (Georges)	$(\bigoplus, \{x_2\}, \{(x_1, x_2), (x_2, x_1)\}),$ $(\bigoplus, \{x_9\}, \{(x_9, x_1)\})$	
	$\mathbf{E} = \{\{x_2, x_9\}\}$	

Tour 3 Jean est satisfait, son envie d'aller au cinéma quel que soit le film est comblée. Il ne dit donc rien et x_2 est toujours accepté.

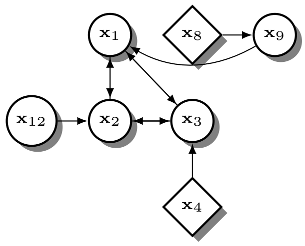
Tour 4 Léo a lui aussi son idée sur la soirée : ne rien faire. Pour être sûr de mettre tout le monde dans l'embarras, il propose un restaurant. Il ajoute donc au système partagé l'argument x_3 et les interactions qu'il a avec x_1 et x_2. En conséquence, seul x_9 est accepté, et aucune décision ne peut être prise concernant x_1, x_2 et x_3.

Tour	Opérations	État du dialogue
4 (Léo)	$(\bigoplus, \{x_3\}, \{(x_3, x_1), (x_3, x_2),$ $(x_1, x_3), (x_2, x_3)\})$	
	$\mathbf{E} = \{\{x_2, x_9\}, \{x_3, x_9\}\}$	

Tour 5 Louise voudrait bien qu'une décision soit prise, et pour cela elle peut s'inscrire en faux contre x_2 ou x_3. Parmi les différents arguments qui lui sont accessibles, elle décide de jouer la carte de l'amateurisme cinéphile éclairé en s'insurgeant du fait que f_h ait été localisé en version française (x_{12}). Cet argument, qui attaque x_2, permet de réinstaurer x_3.

Tour	Opérations	État du dialogue
5 (Louise)	$(\bigoplus, \{x_{12}\}, \{(x_{12}, x_2)\})$	
	$\mathbf{E} = \{\{x_3, x_9, x_{12}\}\}$	

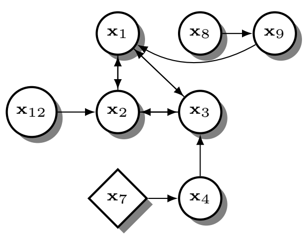
Tour 6 Rosa n'en démord pas, elle veut voir f_a. Pour cela elle conteste la médiocrité des acteurs en disant qu'ils ont déjà été primés (x_8) et joue sur la corde sensible de l'inquiétude pécuniaire de l'étudiant sans le sou (x_4). L'argument x_1 redevient accepté tandis que x_2 et x_3 sont rejetés.

Tour	Opérations	État du dialogue
6 (Rosa)	$(\bigoplus, \{x_4\}, \{(x_4, x_3)\})$, $(\bigoplus, \{x_8\}, \{(x_8, x_9)\})$	
	$\mathbf{E} = \{\{x_1, x_4, x_8, x_{12}\}\}$	

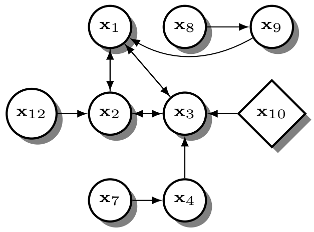
Tour 7 Georges perd espoir, il n'a plus d'argument pour défendre son idée. Il ne dit donc rien et passe son tour.

Tour 8 Jean ne dit rien non plus, son but est satisfait et il ne veut pas prendre le risque de faire changer la donne.

Tour 9 Léo n'a pas changé son fusil d'épaule et retente d'embrouiller son monde. Il prétexte connaître un moyen pour avoir une facture moins salée au restaurant (x_7). Ceci défait l'argument x_4 et remet l'argument x_3 dans la course, ce qui donne deux solutions possibles pour la soirée : aller voir f_a ou manger un morceau.

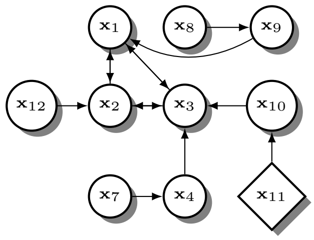
Tour	Opérations	État du dialogue
9 (Léo)	$(\bigoplus, \{x_7\}, \{(x_7, x_4)\})$	
	$\mathbf{E} = \{\{x_1, x_7, x_8, x_{12}\}, \{x_3, x_7, x_8, x_{12}\}\}$	

Tour 10 Louise est ulcérée par la mauvaise foi de son ami. Elle réplique en signalant que ce restaurant n'est pas végétarien alors que, depuis peu, ils avaient tous promis de s'y mettre (x_{10}). L'argument x_3 est à nouveau rejeté.

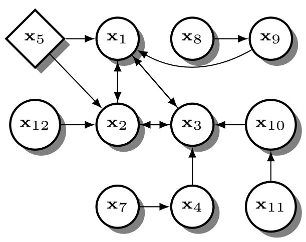
Tour	Opérations	État du dialogue
10 (Louise)	$(\bigoplus, \{x_{10}\}, \{(x_{10}, x_3)\})$	
	$\mathbf{E} = \{\{x_1, x_7, x_8, x_{10}, x_{12}\}\}$	

Tour 11, 12 et 13 Rosa, Georges et Jean passent leur tour et se contentent d'admirer la bataille entre Léo et Louise. En effet, Rosa et Jean ont leurs buts satisfaits et Georges n'a plus d'argument à avancer.

Tour 14 Léo continue son entreprise démoniaque en annonçant qu'il connait très bien ce restaurant et qu'il sait que des plats végétariens sont disponibles (x_{11}). L'argument x_3 redevient une option valable face à x_1.

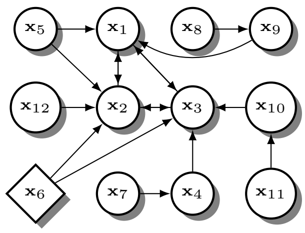
Tour	Opérations	État du dialogue
14 (Léo)	$(\bigoplus, \{x_{11}\}, \{(x_{11}, x_{10})\})$	
	$\mathbf{E} = \{\{x_1, x_7, x_8, x_{11}, x_{12}\},\ \{x_3, x_7, x_8, x_{11}, x_{12}\}\}$	

Tour 15 Louise n'a plus qu'un seul argument à sa disposition pour être sûre que la soirée ne termine pas en un échec fracassant. Elle décide donc de l'utiliser et annonce que s'ils vont au cinéma, ils n'auront pas le temps de manger (x_5). Aller au restaurant (x_3) devient donc la seule option possible.

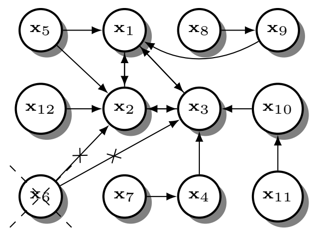
Tour	Opérations	État du dialogue
15 (Louise)	$(\bigoplus, \{x_5\}, \{(x_5, x_1), (x_5, x_2)\})$	
	$\mathbf{E} = \{\{x_3, x_5, x_7, x_8, x_{11}, x_{12}\}\}$	

Tour 16, 17 et 18 Rosa, Georges et Jean, lassés par cette discussion qui n'en finit pas, se sont écartés de quelques pas pour faire une bataille de boules de neige.

Tour 19 Léo est déstabilisé par le dernier argument de Louise, il ne pensait pas qu'elle en viendrait à accepter l'idée du restaurant. Pour tenter d'avoir le dernier mot, il bafouille en disant qu'il vient de se souvenir qu'une amie à lui allait voir f_a (x_6). Dans cette situation, aucun de x_1, x_2 et x_3 n'est accepté.

Tour	Opérations	État du dialogue
19 (Léo)	$(\bigoplus, \{x_6\}, \{(x_6, x_2), (x_6, x_3)\})$	
$\mathbf{E} = \{\{x_5, x_6, x_7, x_8, x_{11}, x_{12}\}\}$		

Tour 20 Louise proteste contre le dernier argument de Léo : premièrement, le restaurant était son idée, et deuxièmement, c'est hors de propos car ils avaient prévu depuis longtemps de passer la soirée entre eux ; Rosa, Georges et Jean annoncent de loin qu'ils sont unanimes sur l'invalidité de l'argument de Léo qui est donc rejeté.

Tour	Opérations	État du dialogue
20 (Louise)	$(\ominus, \{x_6\}, \{(x_6, x_2), (x_6, x_3)\})$	
$\mathbf{E} = \{\{x_3, x_5, x_7, x_8, x_{11}, x_{12}\}\}$		

Fin N'ayant plus rien à ajouter, les cinq amis arrêtent là leur discussion. Ils iront manger au restaurant (au grand désarroi de Léo).

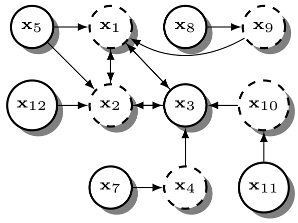
Tour	Opérations	État du dialogue
Fin	–	
	$\mathbf{E} = \{\{x_3, x_5, x_7, x_8, x_{11}, x_{12}\}\}$	

Table 3.3 – *Système d'argumentation représentant l'état final du dialogue entre les cinq agents à propos de l'activité de la soirée (les arguments n'étant pas en pointillés sont acceptés dans le cadre de la sémantique préférée).*

3.3 Allocation de ressources

Cette nouvelle application sera pour nous l'occasion d'utiliser le caractère très abstrait du cadre de Dung (1995). En effet, puisque nous ne savons rien sur les arguments, nous pouvons supposer qu'ils n'en sont pas et considérer que les sommets et arcs d'un système d'argumentation sont totalement autres. Plus précisément, le cadre de Dung (1995) peut être utilisé pour modéliser toute situation où des entités, quelles qu'elles soient, interagissent selon une relation de "dominance" (en fait, de préférence) et d'incompatibilité. Ainsi, le fait qu'une entité A attaque (c'est-à-dire, *a une interaction vers*) une entité B signifie que si A est sélectionnée, B, étant incompatible et dominée par A, est automatiquement rejetée ; notons qu'il est alors très facile de représenter une interaction sans préférence entre deux entités, il suffit en effet que ces dernières s'entre-attaquent. De la même façon, si A est elle-même incompatible avec une autre entité C lui étant préférée, alors A sera rejetée ; plus rien n'empêche alors B d'être sélectionnée, puisqu'elle n'est plus dominée par une entité sélectionnée.

Ainsi, nous le voyons bien dans ce petit exemple, ces deux mécanismes typiques de l'argumentation, en particulier abstraite, s'appliquent bien au-delà de la simple persuasion. Nous allons vérifier cela en nous plaçant dans le contexte d'une *allocation de ressources*. Derrière ce terme fleuri se cache (uniquement pour nous, car nous ne rentrerons pas dans les détails théoriques du concept économique) le simple fait d'affecter une ressource ou une personne à une tâche ou un poste particulier. Cette ressource ou personne affectée n'est alors plus disponible pour un autre poste ; la tâche ou le poste utilisé n'est, quant à lui, plus vacant.

Nous examinons tout d'abord comment utiliser l'argumentation abstraite pour traiter l'allocation de ressources dans la section 3.3.1 page suivante, puis nous nous penchons sur son caractère dynamique dans la section 3.3.2 page 53.

3.3.1 Argumentation abstraite

Un processus tel que l'allocation de ressources est commun dans la vie de tous les jours. On fait par exemple appel à lui lorsque l'on détermine l'emploi du temps de la semaine à venir ou que l'on souhaite constituer une équipe dans le cadre du travail ou du sport. Dans le premier cas, les ressources sont les jours de la semaine et les tâches sont les activités que l'on doit remplir, dans le deuxième cas les personnes sont les salariés de l'équipe et les postes sont le travail qu'ils vont avoir à faire.

Approfondissons et analysons cette application en nous plongeant dans le cadre du sport. Plus précisément, nous nous plaçons dans le cadre de la constitution d'une équipe d'un sport. [4]

3.3.1.1 Contexte

Dans une équipe, chaque joueur est apte à assumer un ou plusieurs postes, pour lesquels il a plus ou moins de succès. Ainsi, supposons qu'un entraîneur cherche à déterminer une manière de placer trois joueurs j_1, j_2 et j_3 sur quatre postes po_1, po_2, po_3 et po_4. L'entraîneur a les contraintes suivantes :

- j_1 peut jouer aux postes po_1, po_2 et po_3,
- j_2 peut jouer aux postes po_2 et po_3,
- j_3 peut jouer aux postes po_3 et po_4.

Les joueurs n'étant pas dotés du don d'ubiquité, la prise d'un poste est exclusive et empêche la prise d'un autre poste. De la même façon, un poste ne peut être occupé par plusieurs joueurs. Ainsi, par exemple, si le joueur j_1 prend le poste po_2, d'une part il ne pourra prétendre aux autres postes, et d'autre part le joueur j_2 ne pourra pas aspirer à prendre ce poste.

Un joueur peut être meilleur à un poste que ses camarades. Ainsi, l'entraîneur de l'équipe se rend compte que j_1 est plus performant au poste po_3 qu'aux postes po_1 et po_2, mais néanmoins meilleur au poste po_2 que j_2. Ce dernier est par contre le meilleur au poste po_3, et l'entraîneur préfère donc que j_2 occupe celui-ci. Enfin, j_3 est assez équilibré dans ses qualités et égale j_1 au poste po_3. L'entraîneur, plongé dans un profond désarroi, se demande s'il y aura des dispositions possibles pour son équipe, et si oui, lesquelles.

Étudions maintenant cet exemple à travers les notions d'argumentation abstraite, et comment l'entraîneur peut s'en servir pour prendre sa décision.

3.3.1.2 Interprétation des notions d'argumentation abstraite

Suite à la présentation de son contexte, nous allons voir comment utiliser les notions théoriques de l'argumentation abstraite dans notre application de l'allocation de ressources.

4. La détermination du sport considéré est laissée au bon vouloir du lecteur ; l'auteur de ces lignes lui suggèrera simplement d'en choisir un véhiculant des valeurs d'entraide et de partage.

Interprétation des notions de bases de l'argumentation abstraite

Dans un tel contexte, les arguments abstraits n'ont plus rien à voir avec des arguments tels qu'il est possible de les imaginer. Il est intéressant de se demander alors à quoi ces derniers, ainsi que leurs interactions, peuvent renvoyer.

Argument L'application de l'argumentation abstraite dans un tel contexte se fait par la création d'arguments représentant l'association d'une ressource à une tâche ou d'une personne à un poste. Nous pouvons donc créer sept arguments, chacun représentant la prise de fonction d'un joueur sur un poste qui lui est accessible ; nous obtenons donc l'ensemble d'arguments $\{j_1po_1, j_1po_2, j_1po_3, j_2po_2, j_2po_3, j_3po_3, j_3po_4\}$.

Attaque Une attaque provient de l'incompatibilité entre deux associations. Une incompatibilité représente tout ce qui empêche deux associations de coexister, par exemple, une même personne ne pouvant faire deux choses à la fois, des personnalités en conflit ne pouvant travailler en même temps, etc.

Une "simple" incompatibilité est représentée par des associations s'entre-attaquant, c'est-à-dire lorsque les deux arguments s'attaquent l'un l'autre. Ainsi, si l'on ne tient pas compte des préférences de l'entraîneur vues plus haut, et que l'on s'en tient aux simples incompatibilités entre associations, nous obtenons le système d'argumentation présenté dans la figure 3.4.

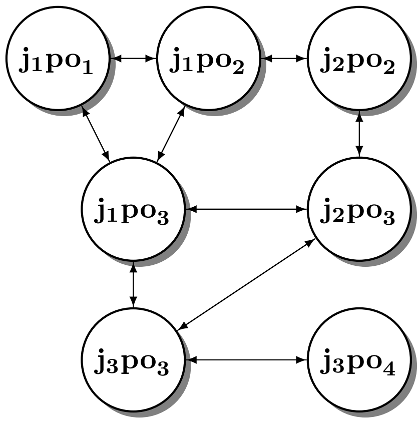


Figure 3.4 – *Illustration des interactions entre les différentes associations joueur/poste.*

Une attaque simple, c'est-à-dire lorsque une association en attaque une autre mais que l'inverse n'est pas vrai, est obtenue par une incompatibilité entre deux associations *et* une préférence pour l'une d'entre elles. Par exemple, si l'entraîneur préfère que j_1 prenne le poste po_3 plutôt que po_1, alors l'argument j_1po_3 attaque l'argument j_1po_1 mais pas l'inverse. La figure 3.5 page suivante présente les nouvelles interactions entre les associations joueur/poste prenant en compte les préférences de l'entraîneur.

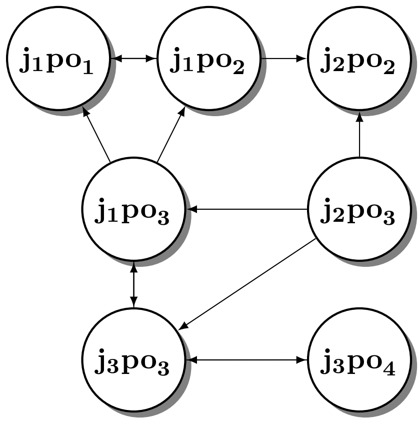


Figure 3.5 – *Illustration des interactions entre les différentes associations joueur/poste prenant en compte les préférences de l'entraîneur.*

Défense La défense représente l'affectation d'un joueur a un poste alors qu'il n'est pas forcément celui qui aurait été préféré à l'origine. Ceci peut notamment arriver dans le cas où un joueur est mis sur la touche par un autre joueur meilleur que lui à un poste particulier, mais que ce dernier est préféré sur un autre poste, libérant la place au passage.

Interprétation des sémantiques d'acceptabilité

De manière similaire à l'application de l'audience, nous nous intéressons ici aux sémantiques et à leur interprétation dans le cadre de l'allocation de ressources.

Sémantique basique La sémantique basique correspond à la constitution d'une équipe basée sur des joueurs étant incontestés à leur poste. Par voie de conséquence, les joueurs qui, à l'origine ou par la force des choses, ne peuvent prétendre qu'à une seule place sont sélectionnés. Plus précisément, tout joueur pouvant jouer à plusieurs postes, et hésitant entre ceux-là, est éloigné. Cette sémantique favorise donc une équipe où le meilleur joueur disponible pour un poste l'occupe, et empêche la diversification des compétences des joueurs à plusieurs postes. Elle assure une équipe "forte", déterminée par les joueurs préférés à leur poste, et empêche tout doute sur les associations joueur/poste, c'est-à-dire sur la qualité des joueurs mis sur le terrain. Incidemment, une telle rigidité dans le choix des joueurs peut mener à un échec de la constitution de l'équipe, c'est-à-dire à une extension vide.

Une telle sémantique peut être désirable lorsque l'on veut s'assurer que les "stars" de l'équipe occupent bien leur poste fétiche (par exemple pour contenter le public), au risque de ne tout simplement pas pouvoir constituer l'équipe.

Si nous utilisons la sémantique basique sur le système d'argumentation de la figure 3.5, nous obtenons l'ensemble d'extensions $\{\{j_2po_3, j_3po_4\}\}$.

Sémantique préférée La sémantique préférée permet de déterminer différentes combinaisons viables de l'équipe. En effet, les différentes extensions, se défendant les unes contre les autres, traduisent tout ce qu'il est possible de faire grâce aux compétences des joueurs. À la manière de l'audience, le doute est ici permis sur l'affectation d'un joueur à un poste ou un autre, les deux étant envisagés. Cette sémantique, contrairement à la précédente, autorise la diversification des joueurs au niveau de leurs compétences, et même, à un certain degré, l'encourage ; un joueur jouant à plusieurs postes aura en effet plus de chance de se retrouver sur le terrain. Néanmoins, elle ne permet donc pas forcément d'arrêter un choix quant à l'équipe, puisque elle peut fournir plusieurs solutions.

Cette sémantique est donc désirable lorsque l'on veut s'assurer de considérer toutes les possibilités d'organisation de l'équipe, le risque étant qu'elle ne puisse pas nous permettre de déterminer laquelle choisir.

Si nous utilisons la sémantique préférée sur le système d'argumentation de la figure 3.5 page précédente, nous obtenons l'ensemble d'extensions $\{\{j_1po_1, j_2po_3, j_3po_4\}, \{j_1po_2, j_2po_3, j_3po_4\}\}$.

Sémantique stable Comme vu précédemment, cette sémantique rajoute une contrainte forte à la précédente. En effet, en plus de respecter ce qui a été dit plus haut pour la sémantique préférée, une équipe stable doit faire en sorte que toute autre association joueur/poste soit attaquée par l'équipe. Ceci permet une justification optimale du rejet d'un joueur sur un poste car si ce dernier ne peut jouer à un poste, c'est forcément parce qu'un autre joueur lui est préféré à ce poste, ou que lui-même doit jouer à un autre poste. Cette sémantique permet donc, en quelque sorte, de trouver les organisations de l'équipe les plus justifiables (vis-à-vis des critiques, par exemple), puisque tout choix d'abandon d'une association joueur/poste sera justifié par une "meilleure" association. Il est possible de ne pas trouver d'équipe stable, signifiant alors qu'aucune disposition de l'équipe n'a permis de justifier l'impossibilité pour un joueur d'officier à un poste particulier.

Cette sémantique peut donc se révéler désirable lorsque l'on souhaite ne négliger aucune des organisations possibles de l'équipe (à la manière de la sémantique préférée) tout en favorisant les organisations justifiant l'éviction de joueurs sur certains postes.

Si nous utilisons la sémantique stable sur le système d'argumentation de la figure 3.5 page ci-contre, nous obtenons l'ensemble d'extensions $\{\{j_1po_1, j_2po_3, j_3po_4\}, \{j_1po_2, j_2po_3, j_3po_4\}\}$.

Sémantique semi-stable Cette sémantique à mi-chemin entre sémantique préférée et sémantique stable permet de sélectionner les ensembles associations joueur/poste attaquant le plus grand nombre d'autres associations. Son premier avantage est de s'assurer, à l'inverse de la sémantique stable, qu'il y aura au moins une équipe possible. Elle permet par ailleurs de trouver les équipes justifiant le plus possible ses choix par l'éviction du plus grand nombre d'associations joueurs/postes.

À ce titre, elle est donc désirable lorsque la sémantique stable n'offre aucune extension car elle permet de se rabattre sur la ou les "meilleures" organisations d'équipe possibles, c'est-à-dire celles qui laissent le moins de doute sur le choix des associations joueur/poste,

Si nous utilisons la sémantique semi-stable sur le système d'argumentation de la figure 3.5 page 50, nous obtenons l'ensemble d'extensions $\{\{j_1po_1, j_2po_3, j_3po_4\}, \{j_1po_2, j_2po_3, j_3po_4\}\}$.

Interprétation du statut d'un argument

Examinons maintenant les statuts des arguments dans ces différentes sémantiques et leurs interprétations dans le contexte de l'allocation de ressources.

Accepté sceptiquement Un argument accepté sceptiquement appartient à toutes les extensions de la sémantique considérée. Un tel argument représente une association joueur/poste valable et reconnue dans toutes les configurations possibles de l'équipe. Il n'y a donc aucun doute sur le fait que ce joueur doit jouer à ce poste, étant incontesté à celui-ci.

Accepté crédulement Un argument accepté crédulement appartient à une (ou plusieurs mais pas toutes) des multiples extensions de la sémantique considérée. Un tel argument représente ici une association joueur/poste considérée comme valable bien qu'elle soit contestable dans une ou plusieurs des organisations possibles de l'équipe.

Une association joueur/poste étant exclusive, il est impossible que deux joueurs soient au même poste à la fois (ou que deux postes soient occupés par un même joueur). Néanmoins, ceci peut être intéressant pour tenir compte des remplaçants, c'est-à-dire des joueurs jouant au même poste, mais pas en même temps. Cela permet également de représenter l'information qu'un joueur peut officier à un poste différent. Un argument accepté crédulement représente donc une possibilité stratégique pour l'entraîneur, un choix potentiel éprouvé par une des organisations, qu'il garde dans sa besace dans l'éventualité d'un changement nécessaire de la structure de son équipe.

Rejeté N'appartenant à aucune des extensions de la sémantique considérée, un argument rejeté représente une association joueur/poste n'étant valable dans aucune des organisations possibles de l'équipe. Quelle que soit la forme finale de cette dernière, le joueur ne jouera pas à ce poste, ni en titulaire, ni en remplaçant.

Maintenant que nous avons présenté ces trois applications du point de vue statique de l'argumentation abstraite classique, il convient de s'intéresser à sa facette dynamique, ce que nous faisons dans la section suivante.

3.3.2 Interprétation des opérations de changement pour l'allocation de ressources

Addition d'argument L'addition d'un argument correspond à la prise en compte d'une nouvelle association joueur/poste. Ceci peut correspondre à l'arrivée d'un nouvel inscrit pouvant jouer à certains postes, au retour de blessure d'un ancien joueur voulant reprendre ses activités, l'entraînement d'un joueur à un poste auquel il ne pouvait prétendre auparavant, etc. L'addition d'une nouvelle association occasionne de fait des conflits et autres incompatibilités avec les autres associations concernant le même joueur ou le même poste, ce qui engendre l'addition de nouvelles attaques. [5]

Suppression d'argument Opération inverse de la précédente, elle représente la défausse d'une association joueur/poste. Un joueur quittant l'équipe, une blessure ou encore une perte de niveau à un certain poste ont pour conséquence la suppression de l'association correspondante. Une incompatibilité n'ayant plus lieu d'être si sa rivale n'existe plus, les attaques concernant l'association sont supprimées par la même occasion. [6]

Addition d'attaque L'addition d'attaque correspond à l'apparition d'une nouvelle incompatibilité entre deux associations. Intuitivement, cela signifie le fait qu'une association pouvant auparavant être sélectionnée simultanément à une autre auparavant, ne le peut plus ; par ailleurs, une association étant auparavant rejetée peut désormais se “défendre” face à une association qui la court-circuitait.

Cela permet de représenter, entre autres, l'émergence de problèmes relationnels entre deux joueurs, un changement de préférence ou encore qu'un joueur a repris suffisamment de niveau peut être à nouveau sélectionnable sur un poste particulier.

Suppression d'attaque La suppression d'attaque correspond, à l'inverse de la précédente, à la disparition d'une incompatibilité entre deux associations. Cela indique soit que le choix d'une association n'implique plus le rejet d'une autre, soit, dans le cas d'une attaque mutuelle entre deux associations, l'apparition d'une préférence pour l'une d'entre elles.

Inversement à l'addition d'attaque, une telle opération permet de représenter l'adoucissement des mœurs entre deux joueurs, la perte de niveau d'un joueur à un poste lui faisant perdre sa place, etc.

Nous allons maintenant conclure en présentant une liste non exhaustive d'applications potentielles présentant les causes d'une addition ou d'une suppression d'argument ou d'attaque.

5. Dans une dimension moins sportive et plus relative à l'emploi, cela peut également correspondre à la création d'un nouveau poste que les actuels employés peuvent tenter de briguer ; une telle nouvelle association sera donc en conflit avec celle représentant l'ancien emploi de la personne.

6. Dans le domaine de l'emploi, la tristement célèbre “suppression de postes” se comprend sans besoin d'explication.

3.4 Discussion

Dans ce chapitre, nous avons donné l'intuition des concepts abstraits sur laquelle nous nous basons grâce à trois exemples détaillés, à savoir

- une audience de tribunal voyant s'affronter un procureur et un avocat pour déterminer la culpabilité d'un accusé,
- une discussion entre plusieurs personnes pour arrêter une décision quant à l'objet de la soirée et
- un problème d'allocation de ressources, c'est-à-dire d'affectation de personnes sur des postes mutuellement exclusifs.

Dans chacun de ces exemples, nous avons en premier lieu montré comment raccrocher la théorie de l'argumentation abstraite de Dung (1995) à la réalité. Pour cela, nous avons modélisé ces contextes de la vie de tous les jours grâce à l'argumentation abstraite. Pour chacun de ces contextes, une explication intuitive des notions importantes du cadre de Dung a été donnée, en particulier pour les notions de sémantique et de statut des arguments, justifiant ainsi leur existence théorique. En second lieu, nous avons étudié le caractère dynamique de l'argumentation dans ce contexte en montrant son intérêt pour capturer des mécanismes classiques de l'interaction humaine ainsi qu'en expliquant à quoi pourrait correspondre des opérations de changement dans un contexte tel que l'allocation de ressources.

Notons que ces trois exemples ne résument pas l'ensemble des cas d'utilisation des quatre opérations de changement. Sans tenter d'en donner une liste exhaustive, voici un ensemble de cas induisant l'addition d'un argument, la suppression d'un argument et enfin l'addition ou la suppression d'une interactions uniquement.

- L'**addition d'un argument** peut être induite par
  - l'arrivée d'une nouvelle information permettant la construction d'un nouvel argument ;
  - dans un contexte multi-agents, l'énonciation par un agent d'un argument non encore connu par les autres agents (on ne construit pas un nouvel argument, il est donné directement) : ceci est illustré par l'exemple de l'audience de tribunal présenté dans la section 3.1 page 30 ;
  - la réhabilitation d'un argument préalablement interdit : un argument qui a été écarté parce qu'il était illégal au regard du contexte du moment peut être à nouveau pris en compte si ce contexte vient à changer.
- La **suppression d'un argument** peut être induite par
  - une révision des connaissances sous-jacentes (si une connaissance n'a plus de sens, peut-être vaut-il mieux supprimer le ou les arguments y étant associés plutôt que d'ajouter un attaquant) ;
  - la mise en place d'une stratégie, c'est-à-dire ne pas vouloir/pouvoir révéler un argument (normes sociales, volonté de ne pas fournir d'informations à un adversaire, etc.) : nous en donnons une illustration concise

au travers de la notion d'*occultation* grâce à l'exemple de l'organisation d'une soirée dans la section 3.2 page 39 ;

— l'adaptation au contexte (arguments tronqués [7], agents ne partageant pas les mêmes bases de connaissances, les mêmes logiques de construction d'arguments) : l'*objection*, décrite également dans l'exemple de l'audience, exemplifie la suppression d'un argument considéré comme illégal au regard du contexte ;

— un processus de vérification *a posteriori* : si cet argument n'avait pas été donné, que se serait-il passé ?

- Les **addition et suppression d'une attaque** peuvent être induites par
  - — l'addition ou la suppression d'un argument ;
  - — des préférences entre arguments : suppression de l'attaque allant sur un argument que l'on préfère à son attaquant ; notons que ces préférences peuvent aussi évoluer suite à un changement de priorités : par exemple, si l'on suppose que les arguments sont créés à partir de règles ayant chacune une certaine priorité, un changement dans ces priorités peut modifier les préférences que l'on a sur nos arguments, et donc les attaques qu'il y a entre eux ;
  - — la prise en compte de la force (Cayrol *et al.* (2010a)) des attaques : des attaques pondérées jugées trop faibles ou trop incertaine (*i.e.* sous un certain seuil) peuvent être supprimées ;
  - — l'adaptation au contexte (arguments tronqués, agents ne partageant pas les mêmes bases de connaissances, les mêmes logiques de construction d'attaques).

Il est intéressant de noter que certaines causes sont présentes plusieurs fois : la "révision" des connaissances de l'agent (ajout ou remise en cause), la prise en compte du contexte ou bien celle de préférences/priorités. La première de ces causes est classique et nous ne l'aborderons donc pas ; par contre, l'annexe D.2 page 225 examine de plus près les deux autres situations au travers d'exemples.

Parmi ces quatre opérations, nous avons en particulier présenté des cas d'utilisation concrets de l'opération de suppression d'un argument (opération peu approfondie dans la littérature jusque-là) notamment lors de l'objection d'un argument. L'utilité de cette opération nous pousse à l'étudier plus avant, au côté de celle, plus classique, d'addition d'un argument.

Ainsi, dans la suite de ce travail, nous proposons une étude théorique du changement couvrant à la fois l'addition et la suppression d'un argument. Pour cela, dans la partie suivante, nous commençons par modifier la typologie de propriétés du changement proposée par Cayrol *et al.* (2010c) de façon en prendre en compte

7. Un argument tronqué, aussi appelé *enthymème*, est un argument dont *une des prémisses ou la conclusion a été omise* (Audi (1999)).

ces deux opérations (chapitre 4 page 59). Nous proposons ensuite une caractérisation des différentes propriétés du changement pour chacune de ces deux opérations (chapitre 5 page 81).

Deuxième partie

Enrichissement du cadre de base

Chapitre 4

Typologie du changement

Nous classifions trop et ne jouissons pas assez.

Okakura Kakuzo,
Le Livre du thé.

Sommaire

Nous l'avons vu précédemment, au fil d'une discussion ou lors de l'acquisition de nouvelles informations, un système d'argumentation peut subir des changements, tels que l'addition d'un nouvel argument ou la suppression d'un argument considéré comme inapproprié. Il est intéressant d'examiner ces changements et les modifications qu'ils entraînent. En particulier, nous souhaitons pouvoir caractériser ces modifications, appelées propriétés du changement (voir la section 2.2 page 16), c'est-à-dire trouver les conditions nécessaires et/ou suffisantes décrivant l'évolution du système d'argumentation lorsqu'on rajoute ou supprime un argument. Cette étude a déjà été menée en partie pour l'addition d'un argument. Or, bien que l'étude de la dynamique des systèmes d'argumentation commence à prendre de l'ampleur (Boella *et al.* (2009a,b); Baumann et Brewka (2010); Moguillansky *et al.* (2010); Liao *et al.* (2011); Liao (2014)), la suppression d'argument n'a été que peu considérée pour le moment.

Nous proposons donc, dans ce chapitre, de préciser la notion de propriété du changement, introduite dans Cayrol *et al.* (2010c), ainsi que la typologie les répertoriant, afin de prendre en compte l'opération de suppression d'un argument.

En effet, cette typologie, ayant été faite initialement dans l'optique d'une étude focalisée sur l'addition d'un argument, permet difficilement de couvrir à la fois les changements causés par l'addition et la suppression d'argument. Par exemple, en nous référant à la typologie présentée dans la table 2.2 page 19, si l'addition d'un argument cause un changement *expansif* ("expansive"), le fait de le supprimer pour obtenir à nouveau le système d'origine causerait un changement *modifiant* ("altering") ; or, ce dernier pourrait être qualifié de façon plus précise.

Ce chapitre se compose de deux sections principales, la première donnant quelques notions formelles portant sur les propriétés du changement (section 4.1), et la seconde proposant un catalogue approfondi des propriétés du changement reflétant les possibles modifications d'un système d'argumentation (section 4.2 page ci-contre). Enfin, la section 4.3 page 75 propose une synthèse des résultats.

En outre, notons que ce chapitre développe les résultats de Bisquert (2011); Bisquert *et al.* (2011, 2012b).

4.1 Nouvelles notions

L'impact d'une opération de changement sur un système d'argumentation est étudié au travers de la notion de *propriété du changement*, ce qui constitue la première de nos nouvelles notions. Ainsi, une propriété du changement π peut être vue comme un ensemble de paires $(\mathcal{G}, \mathcal{G}')$, où $\mathcal{G}$ et $\mathcal{G}'$ sont les graphes d'argumentation représentant le système avant et après le changement et telles que le changement de $\mathcal{G}$ à $\mathcal{G}'$ satisfait π.

Reprenons notre exemple 2.3 page 17 montrant qu'un même système d'argumentation peut être obtenu via différentes opérations de changement.

Exemple 2.3 (cont.). *Soit la propriété π suivante :*

> $\pi(\mathcal{G}, \mathcal{G}')$ *si et seulement si : "Toute extension du système d'argumentation d'arrivée ($\mathcal{G}'$) est incluse dans au moins une extension du système d'argumentation de départ ($\mathcal{G}$)."*

Ainsi, pour la sémantique basique, $\pi(\mathcal{G}_1, \mathcal{G}')$ n'est pas vérifiée, mais $\pi(\mathcal{G}_2, \mathcal{G}')$, $\pi(\mathcal{G}_3, \mathcal{G}')$ et $\pi(\mathcal{G}_4, \mathcal{G}')$ le sont.

Dans la suite de ce travail, nous parlerons souvent du fait qu'une opération de changement *satisfait* une propriété particulière :

Définition 4.1 (Opération satisfaisant une propriété). *Une opération de changement o* satisfait *une propriété π si et seulement si $\forall \mathcal{G},\ \pi(\mathcal{G}, o(\mathcal{G}))$.*

Exemple 2.3 (cont.). *Le système d'argumentation $\mathcal{G}_1$ est tel que $\pi(\mathcal{G}_1, \mathcal{G}')$ n'est pas vérifiée. Ainsi, l'opération de changement $o_1 = \circleddash(z, \mathcal{R}_z)$, avec $\mathcal{R}_z = (\{(z,x), (z,y)\})$, ne satisfait pas π.*

Nous allons maintenant établir une typologie de ces propriétés du changement.

4.2 Nouvel ensemble de propriétés du changement

Les propriétés du changement reflètent des modifications d'un système d'argumentation causées par une opération de changement. Au sein de cette section, nous nous intéressons à ces modifications et tentons de les définir de façon à obtenir une classification précise (sans recouvrement) et exhaustive. Pour ce faire, nous nous concentrons en premier lieu sur les propriétés concernant l'ensemble des extensions, puis en second lieu sur les propriétés concernant les ensembles d'arguments et enfin sur les propriétés concernant le statut d'un argument particulier.

Notons que, bien que les propriétés décrites dans cette section concernent n'importe quel type de changement ($\oplus$, $\ominus$, $\textcircled{\oplus}$, $\textcircled{\ominus}$), elles ne seront illustrées que grâce à des exemples portant sur l'opération de suppression d'un argument et de ses interactions ($\textcircled{\ominus}$).

4.2.1 Propriétés concernant l'ensemble des extensions

Tout d'abord, nous décrirons trois cas possibles d'évolution de l'ensemble des extensions :

- le cas *extensif*, dans lequel le nombre d'extensions augmente,
- le cas *restrictif*, dans lequel le nombre d'extensions diminue,
- le cas *constant*, dans lequel le nombre d'extensions reste le même.

Pour chacun de ces cas, nous étudierons des sous-cas particuliers ; ces derniers étant nombreux, voici une courte explication de la notation servant à les désigner : une propriété [1] est qualifiée grâce à une lettre (e pour *extensif*, r pour *restrictif* et c pour *constant*) ayant un indice de la forme $\gamma - \gamma'$, où γ se rapporte à l'ensemble des extensions avant le changement et γ' se rapporte à l'ensemble des extensions après le changement. γ et γ' peuvent ainsi être :

- 0, qui indique que le système d'argumentation n'a pas d'extension,
- $1v$, qui indique que le système d'argumentation a une seule extension vide,
- $1nv$, qui indique que le système d'argumentation a une seule extension non vide,
- k (resp. j), qui indique que le système d'argumentation a k (resp. j) extensions telles que $1 < k$ (resp. $1 < j < k$).

Par exemple, la notation e_{0-1nv} traduit le fait, qu'au cours de changement, le nombre d'extensions augmente (un cas *extensif* donc), n'ayant aucune extension avant le changement (0) et une extension non vide après le changement ($1nv$).

1. Notons néanmoins que certaines propriétés particulières du cas *constant* ne peuvent pas utiliser la nomenclature présentée ici car elles s'appuient sur d'autres notions que le nombre d'extensions ou leur possible vacuité ; nous utiliserons donc une notation où l'indice est remplacé par un qualificatif. Par exemple, nous aurons affaire à la propriété *c-conservatif* quand les extensions sont les mêmes avant et après le changement.

Notons également que pour des raisons de simplicité, nous dirons d'un changement satisfaisant une propriété π qu'il est un "changement π" ; par exemple, un changement satisfaisant la propriété e_{0-1nv} est dit changement e_{0-1nv}.

4.2.1.1 Cas *extensif*

Cas général

Nous nous proposons d'étudier tout d'abord le cas où un changement induit une augmentation du nombre d'extensions dans le système d'argumentation, ce que nous nommons changement *extensif*.

Définition 4.2 (Changement extensif). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est* extensif *si et seulement si* $|\mathbf{E}| < |\mathbf{E}'|$.

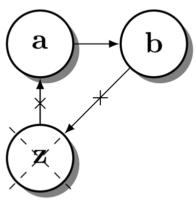
Cas particuliers

Le changement *extensif* étant très général, nous nous proposons donc de raffiner cette propriété suivant le nombre d'extensions de $\mathcal{G}$ et $\mathcal{G}'$.

En outre, puisque nous ne supposons pas l'utilisation d'une sémantique particulière, nous nous devons de distinguer tous les cas particuliers possibles : la sémantique stable autorisera par exemple de n'avoir aucune extension ($|\mathbf{E}| = 0$), chose qui est impossible en sémantique préférée, mais où il est possible d'avoir une extension vide ($|\mathbf{E}| = 1$, $\mathcal{E} = \varnothing$).

Définition 4.3 (Changement e_{0-1nv}). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est e_{0-1nv} si et seulement si $|\mathbf{E}| = 0$ et $|\mathbf{E}'| = 1$, avec $\mathcal{E}' \neq \varnothing$.*

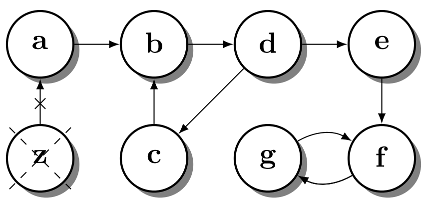
Exemple 4.1. *Sous la sémantique stable, le changement $\ominus$ avec z et $\mathcal{R}_z = \{(z, a), (b, z)\}$ est e_{0-1nv} car :*



$\mathbf{E} = \varnothing$,
$\mathbf{E}' = \{\{a\}\}$.

Définition 4.4 (Changement e_{0-k}). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est e_{0-k} si et seulement si $|\mathbf{E}| = 0$ et $|\mathbf{E}'| > 1$.*

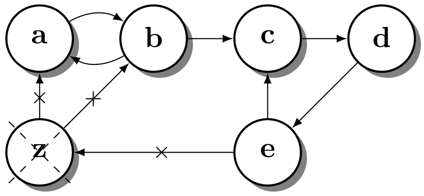
Exemple 4.2. *Sous la sémantique stable, le changement $\ominus$ avec z et $\mathcal{R}_z = \{(z, a)\}$ est e_{0-k} car :*



$\mathbf{E} = \varnothing$,
$\mathbf{E}' = \{\{a, d, f\}, \{a, d, g\}\}$.

Définition 4.5 (**Changement** e_{1v-k})**.** *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est e_{1v-k} si et seulement si $|\mathbf{E}| < |\mathbf{E}'|$ et $|\mathbf{E}| = 1$, avec $\mathcal{E} = \varnothing$.*

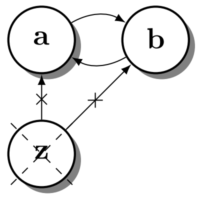
Exemple 4.3. *Sous la sémantique préférée, le changement $\ominus$ avec z et $\mathcal{R}_z = \{(z,a),(z,b),(e,z)\}$ est e_{1v-k} car :*



$$\mathbf{E} = \{\{\}\},$$
$$\mathbf{E}' = \{\{a\},\{b,d\}\}.$$

Définition 4.6 (**Changement** e_{1nv-k})**.** *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est e_{1nv-k} si et seulement si $|\mathbf{E}| < |\mathbf{E}'|$ et $|\mathbf{E}| = 1$, avec $\mathcal{E} \neq \varnothing$.*

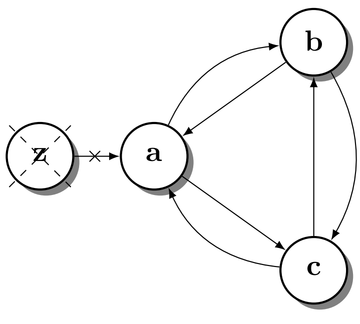
Exemple 4.4. *Sous les sémantiques préférée et stable, le changement $\ominus$ avec z et $\mathcal{R}_z = \{(z,a),(z,b)\}$ est e_{1nv-k} car :*



$$\mathbf{E} = \{\{z\}\},$$
$$\mathbf{E}' = \{\{a\},\{b\}\}.$$

Définition 4.7 (**Changement** e_{j-k})**.** *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est e_{j-k} si et seulement si $1 < |\mathbf{E}| < |\mathbf{E}'|$.*

Exemple 4.5. *Sous les sémantiques préférée et stable, le changement $\ominus$ avec z et $\mathcal{R}_z = \{(z,a)\}$ est e_{j-k} car :*



$$\mathbf{E} = \{\{z,b\},\{z,c\}\},$$
$$\mathbf{E}' = \{\{a\},\{b\},\{c\}\}.$$

Remarque 4.1. *Puisque nous ne considérons que les sémantiques préférée, basique et stable, il est important de noter que le cas e_{0-1v}, c'est-à-dire le fait de n'avoir aucune extension avant le changement et une extension vide après le changement, est impossible. En effet, un tel changement pourrait avoir lieu dans deux configurations :*

- *s'il est possible de changer de sémantique lors du changement,*
- *si l'on manipule un système d'argumentation n'ayant qu'un seul argument s'attaquant lui-même que l'on supprime (par exemple, dans le cas de la sémantique stable).*

Or, nous avons supposé précédemment que le changement de sémantique ainsi que les arguments s'attaquant eux-même ne sont pas autorisés. La propriété du changement e_{0-1v} *ne peut donc jamais avoir lieu.*

4.2.1.2 Cas *restrictif*

Cas général

De façon symétrique au changement *extensif*, nous nous proposons d'étudier à présent le cas où un changement induit une diminution du nombre d'extensions dans le système d'argumentation, ce que nous nommons changement *restrictif*.

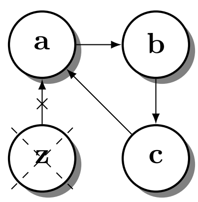
Définition 4.8 (Changement restrictif). *Le changement de* $\mathcal{G}$ *à* $\mathcal{G}'$ *est* restrictif *si et seulement si* $|\mathbf{E}| > |\mathbf{E}'|$.

Cas particuliers

Pour les cas particuliers de changement *restrictif*, nous procédons de la même manière que pour le changement *extensif*, c'est-à-dire que nous prenons en compte le nombre d'extensions ainsi que leur possible vacuité.

Définition 4.9 (Changement r_{1nv-0}). *Le changement de* $\mathcal{G}$ *à* $\mathcal{G}'$ *est* r_{1nv-0} *si et seulement si* $|\mathbf{E}| = 1$, *avec* $\mathcal{E} \neq \varnothing$, *et* $|\mathbf{E}'| = 0$.

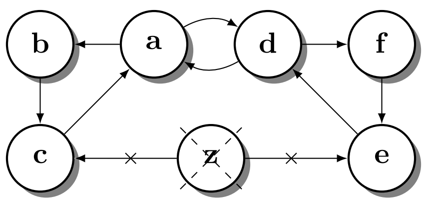
Exemple 4.6. *Sous la sémantique stable, le changement* $\ominus$ *avec* z *et* $\mathcal{R}_z = \{(z, a)\}$ *est* r_{1nv-0} *car :*



$$\mathbf{E} = \{\{z, b\}\},$$
$$\mathbf{E}' = \varnothing.$$

Définition 4.10 (Changement r_{k-0}). *Le changement de* $\mathcal{G}$ *à* $\mathcal{G}'$ *est* r_{k-0} *si et seulement si* $|\mathbf{E}| > 1$ *et* $|\mathbf{E}'| = 0$.

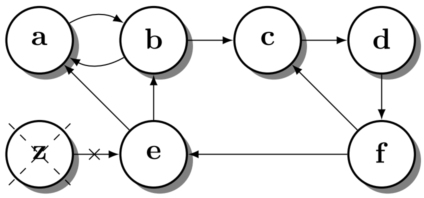
Exemple 4.7. *Sous la sémantique stable, le changement* $\ominus$ *avec* z *et* $\mathcal{R}_z = \{(z, c), (z, e)\}$ *est* r_{k-0} *car :*



$$\mathbf{E} = \{\{z, a, f\}, \{z, b, d\}\},$$
$$\mathbf{E}' = \varnothing.$$

Définition 4.11 (**Changement** r_{k-1v}). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est r_{k-1v} si et seulement si $|\mathbf{E}| > |\mathbf{E}'|$ et $|\mathbf{E}'| = 1$, avec $\mathcal{E}' = \varnothing$.*

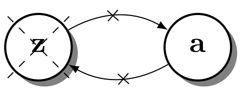
Exemple 4.8. *Sous la sémantique préférée, le changement $\ominus$ avec z et $\mathcal{R}_z = \{(z,e)\}$ est r_{k-1v} car :*



$$\mathbf{E} = \{\{z,a\},\{z,b,d\}\},$$
$$\mathbf{E}' = \{\{\}\}.$$

Définition 4.12 (**Changement** r_{k-1nv}). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est r_{k-1nv} si et seulement si $|\mathbf{E}| > |\mathbf{E}'|$ et $|\mathbf{E}'| = 1$, avec $\mathcal{E}' \neq \varnothing$.*

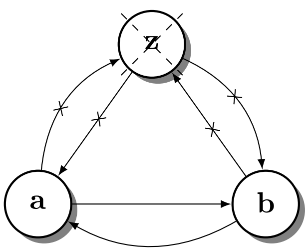
Exemple 4.9. *Sous les sémantiques préférée et stable, le changement $\ominus$ avec z et $\mathcal{R}_z = \{(z,a),(a,z)\}$ est r_{k-1nv} car :*



$$\mathbf{E} = \{\{z\},\{a\}\},$$
$$\mathbf{E}' = \{\{a\}\}.$$

Définition 4.13 (**Changement** r_{k-j}). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est r_{k-j} si et seulement si $1 < |\mathbf{E}'| < |\mathbf{E}|$.*

Exemple 4.10. *Sous les sémantiques préférée et stable, le changement $\ominus$ avec z et $\mathcal{R}_z = \{(z,a),(a,z),(z,b),(b,z)\}$ est r_{k-j} car :*



$$\mathbf{E} = \{\{z\},\{a\},\{b\}\},$$
$$\mathbf{E}' = \{\{a\},\{b\}\}.$$

Remarque 4.2. *Pour les mêmes raisons que celles évoquées dans la remarque 4.1 page 63, le cas r_{1v-0}, c'est-à-dire le fait d'avoir une extension vide avant le changement et de n'avoir aucune extension après le changement, est impossible.*

4.2.1.3 Cas *constant*

Cas général

Contrairement à ce que nous avons vu précédemment, nous nous intéressons maintenant au cas où un changement n'induit aucune modification du nombre d'extensions, ce que nous nommons changement *constant*.

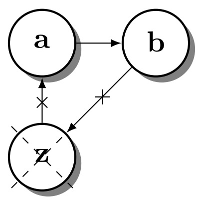
Définition 4.14 (**Changement constant**). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est* constant *si et seulement si* $|\mathbf{E}| = |\mathbf{E}'|$.

Cas particuliers

Parmi les cas particuliers, nous raffinons selon la vacuité des extensions (être vide ou ne pas l'être).

Définition 4.15 (**Changement** c_{1v-1nv}). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est* c_{1v-1nv} *si et seulement si* $|\mathbf{E}| = |\mathbf{E}'| = 1$, $\mathcal{E} = \varnothing$ *et* $\mathcal{E}' \neq \varnothing$.

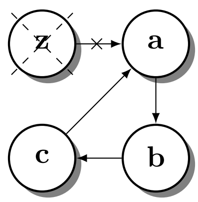
Exemple 4.11. *Sous la sémantique préférée, le changement* $\ominus$ *avec* z *et* $\mathcal{R}_z = \{(z, a), (b, z)\}$ *est* c_{1v-1nv} *car :*



$$\mathbf{E} = \{\{\}\},$$
$$\mathbf{E}' = \{\{a\}\}.$$

Définition 4.16 (**Changement** c_{1nv-1v}). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est* c_{1nv-1v} *si et seulement si* $|\mathbf{E}| = |\mathbf{E}'| = 1$, $\mathcal{E} \neq \varnothing$ *et* $\mathcal{E}' = \varnothing$.

Exemple 4.12. *Sous la sémantique préférée, le changement* $\ominus$ *avec* z *et* $\mathcal{R}_z = \{(z, a)\}$ *est* c_{1nv-1v} *car :*

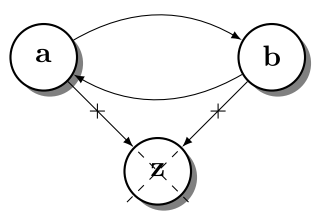


$$\mathbf{E} = \{\{z, b\}\},$$
$$\mathbf{E}' = \{\{\}\}.$$

Considérer le cas *constant* nous permet aussi de nous concentrer sur d'autres critères que le seul nombre d'extensions de $\mathcal{G}$ et $\mathcal{G}'$, ou la possible vacuité de ces mêmes extensions. Nous pouvons ainsi prendre en compte les inclusions éventuelles entre les différentes extensions (de $\mathcal{G}$ vers $\mathcal{G}'$ et vice-versa).

Définition 4.17 (**Changement c-conservatif**). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est* c-conservatif *si et seulement si* $\mathbf{E} = \mathbf{E}'$.

Exemple 4.13. *Sous la sémantique préférée, le changement* $\ominus$ *avec* z *et* $\mathcal{R}_z = \{(a, z), (b, z)\}$ *est* c-conservatif *car :*

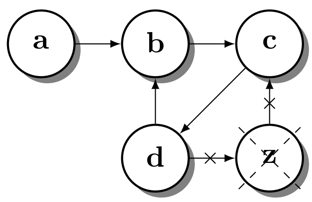


$$\mathbf{E} = \{\{a\}, \{b\}\},$$
$$\mathbf{E}' = \{\{a\}, \{b\}\}.$$

Définition 4.18 (Changement c-expansif). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est* c-expansif *si et seulement si*

1. $|\mathbf{E}| = |\mathbf{E}'| > 0$,
2. $\forall \mathcal{E}_i \in \mathbf{E}, \exists \mathcal{E}'_j \in \mathbf{E}', \varnothing \neq \mathcal{E}_i \subset \mathcal{E}'_j$ *et*
3. $\forall \mathcal{E}'_j \in \mathbf{E}', \exists \mathcal{E}_i \in \mathbf{E}, \varnothing \neq \mathcal{E}_i \subset \mathcal{E}'_j$.

Exemple 4.14. *Sous la sémantique préférée, le changement* $\ominus$ *avec z et* $\mathcal{R}_z = \{(z,c),(d,z)\}$ *est* c-expansif *car :*



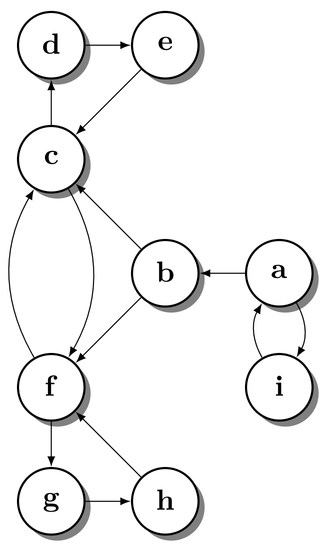
$$\mathbf{E} = \{\{a\}\},$$
$$\mathbf{E}' = \{\{a,c\}\}.$$

Proposition 4.1. *Dans le cadre d'un changement par addition ou suppression d'un argument z, en sémantique préférée, stable ou basique, la condition 3 de la définition 4.18 est conséquence des conditions 1 et 2 de cette même définition.*

La démonstration de la proposition 4.1 est disponible dans l'annexe C.1 page 197.

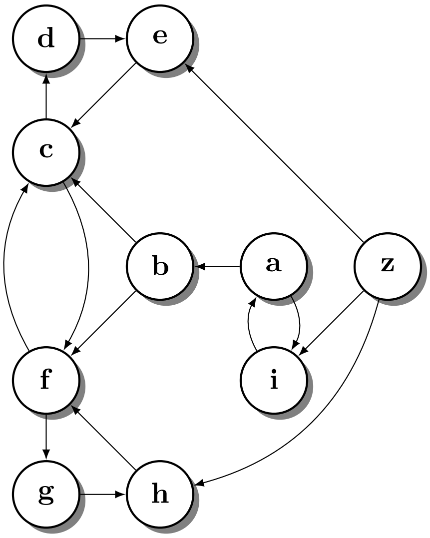
Notons que les conditions 2 et 3 ne sont pas équivalentes, même en présence de la condition 1. Voici un contre-exemple.

Exemple 4.15.



Sous la sémantique préférée, $\mathbf{E} = \{\{a\},\{i,b,d,g\}\}$.

Supposons qu'un changement ait lieu par l'addition de z et de $\mathcal{R}_z = \{(z,i),(z,h),(z,e)\}$, *nous obtenons :*



Sous la sémantique préférée, $\mathbf{E}' = \{\{a, z, c, g\}, \{a, z, f, d\}\}$.

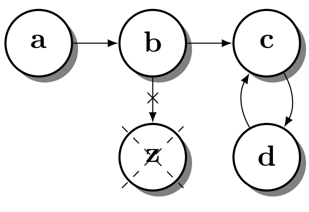
Ce changement satisfait les conditions 1 et 3, mais pas la condition 2.

Le changement *c-limitatif* est défini de façon similaire au changement *c-expansif* ; la définition suivante est donc le pendant de la définition 4.18 page précédente. Par conséquent, il existe une proposition similaire, que nous verrons un peu plus loin, à la proposition 4.1 page précédente concernant le changement *c-limitatif*.

Définition 4.19 (**Changement c-limitatif**)**.** *Le changement de* $\mathcal{G}$ *à* $\mathcal{G}'$ *est* c-limitatif *si et seulement si*

1. $|\mathbf{E}| = |\mathbf{E}'| > 0$,
2. $\forall \mathcal{E}_i \in \mathbf{E}, \exists \mathcal{E}'_j \in \mathbf{E}', \varnothing \neq \mathcal{E}'_j \subset \mathcal{E}_i$ *et*
3. $\forall \mathcal{E}'_j \in \mathbf{E}', \exists \mathcal{E}_i \in \mathbf{E}, \varnothing \neq \mathcal{E}'_j \subset \mathcal{E}_i$.

Exemple 4.16. *Sous les sémantiques préférée et stable, le changement* $\ominus$ *avec* z *et* $\mathcal{R}_z = \{(b, z)\}$ *est* c-limitatif *car :*



$\mathbf{E} = \{\{a, c, z\}, \{a, d, z\}\}$,
$\mathbf{E}' = \{\{a, c\}, \{a, d\}\}$.

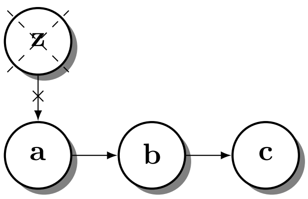
Proposition 4.2. *Dans le cadre d'un changement par addition ou suppression d'un argument* z*, en sémantique préférée, stable ou basique, la condition 2 de la définition 4.19 est conséquence des conditions 1 et 3 de cette même définition.*

La démonstration de la proposition 4.2 est disponible dans l'annexe C.1 page 197.

Remarquons que l'exemple 4.15 page précédente peut être lu à l'envers en considérant que le système contenant z est le système de départ et qu'on réalise sur ce système une suppression d'argument. Dans ce cas, cela illustre le fait que les conditions 1 et 2 de la definition 4.19 page précédente ne suffisent pas à impliquer la condition 3 (chaque extension de départ contient une extension d'arrivée et pourtant il existe une extension d'arrivée qui n'est incluse dans aucune extension de départ).

Définition 4.20 (**Changement c-modifiant**). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est* c-modifiant *si et seulement si* $|\mathbf{E}| = |\mathbf{E}'|$ *et qu'il n'est ni* c-conservatif*, ni* c-expansif*, ni* c-limitatif*, ni* c_{1nv-1v}*, ni* c_{1v-1nv}.

Exemple 4.17. *Sous les sémantiques préférée et stable, le changement* $\ominus$ *avec* z *et* $\mathcal{R}_z = \{(z, a)\}$ *est* c-modifiant *car :*



$$\mathbf{E} = \{\{z, b\}\},$$
$$\mathbf{E}' = \{\{a, c\}\}.$$

Les différentes correspondances entre les propriétés de Cayrol *et al.* (2010c) et les nôtres sont répertoriées dans la table 4.1.

<table>
<tr><th>Cayrol et al. (2010c)</th><th>Nouvelle dénomination</th><th>Cayrol et al. (2010c)</th><th>Nouvelle dénomination</th></tr>
<tr><td rowspan="3">Décisif (“Decisive”)</td><td>e_{0-1nv}</td><td rowspan="4">Destructif (“Destructive”)</td><td>r_{k-0}</td></tr>
<tr><td>c_{1v-1nv}</td><td>r_{k-1v}</td></tr>
<tr><td>r_{k-1nv}</td><td>r_{1nv-0}</td></tr>
<tr><td></td><td></td><td>c_{1nv-1v}</td></tr>
<tr><td rowspan="2">Restrictif (“Restrictive”)</td><td rowspan="2">r_{k-j}</td><td rowspan="2">Expansif (“Expansive”)</td><td>c-expansif</td></tr>
<tr><td>c-modifiant</td></tr>
<tr><td rowspan="4">Questionnant (“Questioning”)</td><td>e_{0-k}</td><td rowspan="2">Conservatif (“Conservative”)</td><td rowspan="2">c-conservatif</td></tr>
<tr><td>e_{1v-k}</td></tr>
<tr><td>e_{1nv-k}</td><td rowspan="2">Modifiant (“Altering”)</td><td>c-limitatif</td></tr>
<tr><td>e_{j-k}</td><td>c-modifiant</td></tr>
</table>

Table 4.1 – *Récapitulatif des correspondances entre les propriétés définies par Cayrol* et al. *(2010c) et celles définies dans ce travail.*

4.2.2 Propriétés concernant les ensembles d'arguments

Nous avons vu dans la section précédente qu'un changement peut occasionner une modification structurelle de l'ensemble des extensions, dont la taille peut augmenter, diminuer ou rester constante. Néanmoins, un changement peut également avoir un impact sur l'acceptabilité d'ensembles d'arguments. Dans cette section, nous nous intéressons à cet impact particulier en définissant une certaine notion de continuité dans l'acceptabilité d'ensembles d'arguments.

Ainsi, lors d'un dialogue, par exemple, il peut être souhaitable que l'addition ou la suppression d'un argument n'influe pas sur l'acceptation des autres arguments. Une première forme de la propriété de *monotonie*, la "monotonie de $\mathcal{G}$ vers $\mathcal{G}'$", traduit ce desiderata en exprimant le fait que tous les ensembles d'arguments acceptés conjointement avant le changement sont toujours acceptés conjointement après. Ainsi, aucun argument accepté n'est perdu et il y a donc une *expansion* (non stricte) de l'acceptabilité.

Définition 4.21 (**Monotonie expansive simple**). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ satisfait la propriété de* monotonie expansive simple[2] *si et seulement si* $\forall \mathcal{E}_i \in \mathbf{E}, \exists \mathcal{E}'_j \in \mathbf{E}',\ \mathcal{E}_i \subseteq \mathcal{E}'_j$.

Par ailleurs, il peut également être désirable que l'addition ou la suppression d'un argument ne permette pas à d'autres arguments d'être nouvellement acceptés, par exemple lors d'un débat avec un adversaire. Ceci est reflété par une autre forme de la propriété de *monotonie*, la "monotonie de $\mathcal{G}'$ vers $\mathcal{G}$", formulant le fait que tous les ensembles d'arguments acceptés conjointement après le changement étaient déjà acceptés conjointement avant. Ainsi, aucun argument accepté n'est gagné et il y a donc une *restriction* (non stricte) de l'acceptabilité.

Définition 4.22 (**Monotonie restrictive simple**). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ satisfait la propriété de* monotonie restrictive simple *si et seulement si* $\forall \mathcal{E}'_j \in \mathbf{E}', \exists \mathcal{E}_i \in \mathbf{E},\ \mathcal{E}'_j \subseteq \mathcal{E}_i$.

Ces propriétés de *monotonie expansive simple* et *monotonie restrictive simple* peuvent prendre des formes plus précises suivant le critère d'acceptabilité des arguments. Ainsi, la *monotonie expansive crédule* correspond au fait que tout argument accepté *crédulement* avant le changement l'est toujours après, et la *monotonie restrictive crédule* traduit le fait que tout argument accepté *crédulement* après le changement l'était déjà avant.

Définition 4.23 (**Monotonie expansive crédule**). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ satisfait la propriété de* monotonie expansive crédule *si et seulement si :*

$$\bigcup_{1 \leq i \leq |\mathbf{E}|} \mathcal{E}_i \subseteq \bigcup_{1 \leq j \leq |\mathbf{E}'|} \mathcal{E}'_j.$$

2. Notons que la propriété de *monotonie expansive simple* correspond à la notion de *monotonie* de Cayrol *et al.* (2010c)

Définition 4.24 (**Monotonie restrictive crédule**). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ satisfait la propriété de* monotonie restrictive crédule *si et seulement si :*

$$\bigcup_{1 \leq j \leq |\mathbf{E}'|} \mathcal{E}'_j \subseteq \bigcup_{1 \leq i \leq |\mathbf{E}|} \mathcal{E}_i.$$

De façon analogue, la *monotonie expansive sceptique* correspond au fait que tout argument accepté *sceptiquement* avant le changement l'est toujours après et la *monotonie restrictive sceptique* correspond au fait que tout argument accepté *sceptiquement* après le changement l'était déjà avant.

Définition 4.25 (**Monotonie expansive sceptique**). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ satisfait la propriété de* monotonie expansive sceptique *si et seulement si :*

$$\bigcap_{1 \leq i \leq |\mathbf{E}|} \mathcal{E}_i \subseteq \bigcap_{1 \leq j \leq |\mathbf{E}'|} \mathcal{E}'_j.$$

Définition 4.26 (**Monotonie restrictive sceptique**). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ satisfait la propriété de* monotonie restrictive sceptique *si et seulement si :*

$$\bigcap_{1 \leq j \leq |\mathbf{E}'|} \mathcal{E}'_j \subseteq \bigcap_{1 \leq i \leq |\mathbf{E}|} \mathcal{E}_i.$$

Notons le lien entre monotonies simples et monotonies crédules :

Proposition 4.3.

- *Si le changement de $\mathcal{G}$ à $\mathcal{G}'$ satisfait la propriété de* monotonie expansive simple *alors il satisfait la propriété de* monotonie expansive crédule.
- *Si le changement de $\mathcal{G}$ à $\mathcal{G}'$ satisfait la propriété de* monotonie restrictive simple *alors il satisfait la propriété de* monotonie restrictive crédule.

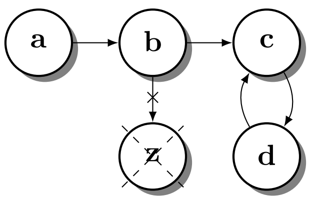
La démonstration de la proposition 4.3 est disponible dans l'annexe C.1 page 197.

Cette proposition est illustrée par les exemples 4.18 à 4.20 pages 71–72. D'autre part, les exemples 4.19 et 4.20 page suivante montrent que ce lien n'existe pas entre monotonies simples et monotonies sceptiques.

Exemple 4.18. *Sous les sémantiques préférée et stable, le changement $\ominus$ avec z et $\mathcal{R}_z = \{(b, z)\}$ est*

- *monotone restrictif simple mais pas monotone expansif simple*
- *donc monotone restrictif crédule et*
- *monotone restrictif sceptique mais pas monotone expansif sceptique*

car :



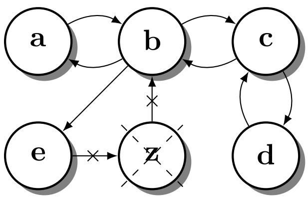
Où :

$$\mathbf{E} = \{\{a,c,z\},\{a,d,z\}\}, \qquad \mathbf{E}' = \{\{a,c\},\{a,d\}\},$$
$$\bigcup_{1\le i\le|\mathbf{E}|} \mathcal{E}_i = \{\{a,c,d,z\}\}, \qquad \bigcup_{1\le j\le|\mathbf{E}'|} \mathcal{E}'_j = \{\{a,c,d\}\},$$
$$\bigcap_{1\le i\le|\mathbf{E}|} \mathcal{E}_i = \{\{a,z\}\}, \qquad \bigcap_{1\le j\le|\mathbf{E}'|} \mathcal{E}'_j = \{\{a\}\}.$$

Exemple 4.19. *Sous la sémantique préférée, le changement* $\ominus$ *avec* z *et* $\mathcal{R}_z = \{(z,b),(e,z)\}$ *est*

- *monotone expansif simple mais pas monotone restrictif simple*
- *donc monotone expansif crédule et*
- *monotone restrictif sceptique mais pas monotone expansif sceptique*

car :



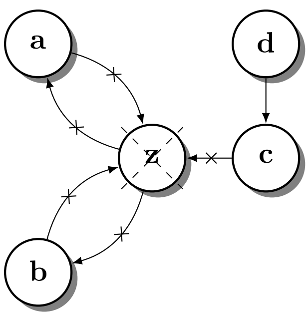
Où :

$$\mathbf{E} = \{\{a,c,e\},\{a,d,e\}\}, \qquad \mathbf{E}' = \{\{a,c,e\},\{a,d,e\},\{b,d\}\},$$
$$\bigcup_{1\le i\le|\mathbf{E}|} \mathcal{E}_i = \{\{a,c,d,e\}\}, \qquad \bigcup_{1\le j\le|\mathbf{E}'|} \mathcal{E}'_j = \{\{a,b,c,d,e\}\},$$
$$\bigcap_{1\le i\le|\mathbf{E}|} \mathcal{E}_i = \{\{a,e\}\}, \qquad \bigcap_{1\le j\le|\mathbf{E}'|} \mathcal{E}'_j = \{\{\}\}.$$

Exemple 4.20. *Sous les sémantiques préférée et stable, le changement* $\ominus$ *avec* z *et* $\mathcal{R}_z = \{(z,a),(a,z),(z,b),(b,z),(c,z)\}$ *est*

- *monotone restrictif simple mais pas monotone expansif simple*
- *donc monotone restrictif crédule et*
- *monotone expansif sceptique mais pas monotone restrictif sceptique*

car :



Où :

$$\mathbf{E} = \{\{d,z\},\{a,b,d\}\}, \qquad \mathbf{E}' = \{\{a,b,d\}\},$$
$$\bigcup_{1\leq i\leq |\mathbf{E}|} \mathcal{E}_i = \{\{a,b,d,z\}\}, \qquad \bigcup_{1\leq j\leq |\mathbf{E}'|} \mathcal{E}'_j = \{\{a,b,d\}\},$$
$$\bigcap_{1\leq i\leq |\mathbf{E}|} \mathcal{E}_i = \{\{d\}\}, \qquad \bigcap_{1\leq j\leq |\mathbf{E}'|} \mathcal{E}'_j = \{\{a,b,d\}\}.$$

4.2.3 Propriétés concernant le statut d'un argument

La section précédente nous a permis de considérer l'impact d'une opération de changement sur des ensembles d'arguments : sur une extension du système d'argumentation, sur l'union des extensions ou encore sur l'intersection des extensions. Une opération de changement peut également avoir une influence sur le statut des arguments, même si cet argument n'est pas celui concerné directement par l'opération de changement. Par exemple, la suppression d'un argument z peut avoir un effet drastique sur le statut des arguments attaqués par z.

Par conséquent, nous nous intéressons dans cette section aux propriétés traduisant l'altération ou la conservation du statut d'un argument particulier x lorsqu'une opération de changement modifie le système d'argumentation. Nous supposons ici que $x \in \mathcal{G} \cap \mathcal{G}'$, c'est-à-dire que l'argument x appartient à l'ensemble des arguments de $\mathcal{G}$ et à l'ensemble des arguments de $\mathcal{G}'$; par conséquent, nous avons $x \neq z$.

Définition 4.27 (**Instauration d'acceptabilité**). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est une* instauration d'acceptabilité *pour un argument x si et seulement si* $\forall \mathcal{E}_i \in \mathbf{E}, x \notin \mathcal{E}_i$ *et*

1. *(*instauration d'acceptabilité uniquement partielle*)* $\exists \mathcal{E}'_j \in \mathbf{E}', x \in \mathcal{E}'_j$ *et* $\exists \mathcal{E}'_l \in \mathbf{E}', x \notin \mathcal{E}'_l$.
2. *(*instauration d'acceptabilité totale*)* $\forall \mathcal{E}'_j \in \mathbf{E}', x \in \mathcal{E}'_j$.

Définition 4.28 (**Élimination d'acceptabilité**). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$* élimine l'acceptabilité *pour un argument x si et seulement si* $\forall \mathcal{E}'_j \in \mathbf{E}', x \notin \mathcal{E}'_j$ *et*

1. *(*élimination d'acceptabilité uniquement crédule*)* $\exists \mathcal{E}_i \in \mathbf{E}, x \in \mathcal{E}_i$ *et* $\exists \mathcal{E}_k \in \mathbf{E}, x \notin \mathcal{E}_k$.
2. *(*élimination d'acceptabilité sceptique*)* $\forall \mathcal{E}_i \in \mathbf{E}, x \in \mathcal{E}_i$.

Outre la pure apparition ou disparition de l'acceptabilité d'un argument, ce dernier peut subir une modification moins “brutale” de son statut d'acceptabilité, c'est-à-dire passer d'une acceptation crédule à une acceptation sceptique, et inversement ; par exemple, un argument n'étant présent que dans quelques extensions avant un changement, peut l'être dans toutes après le changement.

Définition 4.29 (**Diffusion générale d'acceptabilité**). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$ est une* diffusion générale d'acceptabilité *pour un argument x si et seulement si* $\exists \mathcal{E}_i \in \mathbf{E}, x \in \mathcal{E}_i$, $\exists \mathcal{E}_k \in \mathbf{E}, x \notin \mathcal{E}_k$ *et* $\forall \mathcal{E}'_j \in \mathbf{E}', x \in \mathcal{E}'_j$.

Définition 4.30 (Dégradation partielle d'acceptabilité). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$* dégrade partiellement l'acceptabilité *pour un argument x si et seulement si $\forall \mathcal{E}_i \in \mathbf{E}, x \in \mathcal{E}_i$, $\exists \mathcal{E}'_j \in \mathbf{E}', x \in \mathcal{E}'_j$ et $\exists \mathcal{E}'_l \in \mathbf{E}', x \notin \mathcal{E}'_l$.*

Enfin, l'acceptabilité d'un argument peut ne pas accuser de modification ; l'argument conservera alors le statut qu'il possédait avant le changement ; ainsi, un argument accepté (crédulement ou sceptiquement) ou rejeté le sera toujours après un changement.

Notons que les définitions suivantes correspondent à des affinements de la propriété de *monotonie partielle* proposée par Cayrol *et al.* (2010c) ; elles sont donc intimement liées aux définitions des différentes formes de monotonie, dont elles reprennent le principe au niveau d'un argument particulier.

Définition 4.31 (Conservation du statut). *Le changement de $\mathcal{G}$ à $\mathcal{G}'$* conserve le statut *d'un argument x si et seulement si*

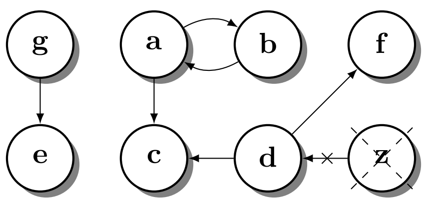
1. *(*conservation de l'acceptabilité uniquement crédule*) $\exists \mathcal{E}_i \in \mathbf{E}, x \in \mathcal{E}_i$, $\exists \mathcal{E}_k \in \mathbf{E}, x \notin \mathcal{E}_k$, $\exists \mathcal{E}'_j \in \mathbf{E}', x \in \mathcal{E}'_j$ et $\exists \mathcal{E}'_l \in \mathbf{E}', x \notin \mathcal{E}'_l$.*
2. *(*conservation de l'acceptabilité sceptique*) $\forall \mathcal{E}_i \in \mathbf{E}, x \in \mathcal{E}_i$ et $\forall \mathcal{E}'_j \in \mathbf{E}', x \in \mathcal{E}'_j$.*
3. *(*conservation du rejet*) $\forall \mathcal{E}_i \in \mathbf{E}, x \notin \mathcal{E}_i$ et $\forall \mathcal{E}'_j \in \mathbf{E}', x \notin \mathcal{E}'_j$.*

Notons que la propriété de *conservation de l'acceptabilité* (uniquement crédule ou sceptique) pour x ne signifie pas que les arguments acceptés avec x sont les mêmes avant le changement et après le changement (ce qui était le cas avec les propriétés de monotonie présentées en section 4.2.2 page 70). Les exemples 4.21 et 4.22 page suivante illustrent les propriétés définies ci-dessus.

Exemple 4.21. *Sous les sémantiques préférée et stable, le changement $\ominus$ avec z et $\mathcal{R}_z = \{(z, d)\}$*

- instaure l'acceptabilité *totale pour d,*
- élimine l'acceptabilité *sceptique pour f,*
- élimine l'acceptabilité *uniquement crédule pour c,*
- conserve l'acceptabilité *sceptique pour g,*
- conserve le rejet *pour e,*
- conserve l'acceptabilité *uniquement crédule pour a et b*

car :

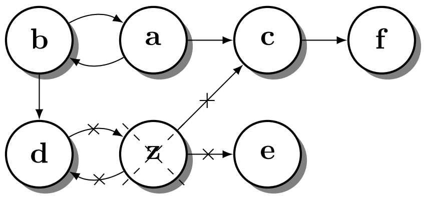


$$\mathbf{E} = \{\{a, f, g, z\}, \{b, c, f, g, z\}\},$$
$$\mathbf{E}' = \{\{a, d, g\}, \{b, d, g\}\}.$$

Exemple 4.22. *Sous les sémantiques préférée et stable, le changement* $\ominus$ *avec* z *et* $\mathcal{R}_z = \{(z, c), (z, d), (z, e), (d, z)\}$

- conserve l'acceptabilité *uniquement crédule pour a, b et d*
- instaure l'acceptabilité *uniquement partielle pour c,*
- dégrade partiellement l'acceptabilité *pour f,*
- *est une* diffusion générale d'acceptabilité *pour e*

car :



$$\mathbf{E} = \{\{a, d, e, f\}, \{a, f, z\}, \{b, z, f\}\},$$
$$\mathbf{E}' = \{\{a, d, e, f\}, \{b, c, e\}\}.$$

4.3 Discussion

Synthèse

Tout au long de ce chapitre, nous avons présenté plusieurs propriétés du changement, et ce à différents niveaux.

Le premier niveau abordé est celui de la **structure générale des extensions** (section 4.2.1 page 61), où sont étudiées les diverses modifications possibles du nombre d'extensions. Nous avons alors identifié trois classes générales de propriétés, les cas *extensif* (définition 4.2 page 62), *restrictif* (définition 4.8 page 64) et *constant* (définition 4.14 page 66), traduisant respectivement une augmentation, une diminution et une conservation du nombre d'extensions. La table 4.2 page suivante présente ces différentes classes.

Au sein de chacune de ces classes, nous avons distingué plusieurs sous-cas particuliers, suivant la structure du système de départ et du système d'arrivée.

Ainsi, le cas général *extensif* a été raffiné en cinq sous-cas, à savoir :

- le changement e_{0-1nv} (définition 4.3 page 62),
- le changement e_{0-k} (définition 4.4 page 62),
- le changement e_{1v-k} (définition 4.5 page 63),
- le changement e_{1nv-k} (définition 4.6 page 63),
- le changement e_{j-k} (définition 4.7 page 63).

De façon symétrique, le cas général *restrictif* a également été raffiné en cinq sous-cas :

- le changement r_{1nv-0} (définition 4.9 page 64),
- le changement r_{k-0} (définition 4.10 page 64),

<table>
<tr><th rowspan="2">Système initial</th><th colspan="3">Système final</th></tr>
<tr><th>$\lvert\mathbf{E}'\rvert = 0$</th><th>$\lvert\mathbf{E}'\rvert = 1$</th><th>$\lvert\mathbf{E}'\rvert > 1$</th></tr>
<tr><td>$\lvert\mathbf{E}\rvert = 0$</td><td>Constant</td><td>Extensif</td><td>Extensif</td></tr>
<tr><td>$\lvert\mathbf{E}\rvert = 1$</td><td>Restrictif</td><td>Constant</td><td>Extensif</td></tr>
<tr><td rowspan="3">$\lvert\mathbf{E}\rvert > 1$</td><td rowspan="3">Restrictif</td><td rowspan="3">Restrictif</td><td>Si $\lvert\mathbf{E}\rvert < \lvert\mathbf{E}'\rvert$:
Extensif</td></tr>
<tr><td>Si $\lvert\mathbf{E}\rvert = \lvert\mathbf{E}'\rvert$:
Constant</td></tr>
<tr><td>Si $\lvert\mathbf{E}\rvert > \lvert\mathbf{E}'\rvert$:
Restrictif</td></tr>
</table>

Table 4.2 – *Classes de propriétés.*

- le changement r_{k-1v} (définition 4.11 page 65),
- le changement r_{k-1nv} (définition 4.12 page 65),
- le changement r_{k-j} (définition 4.13 page 65).

Enfin, le cas général *constant* a été raffiné en six sous-cas :

- le changement c_{1v-1nv} (définition 4.15 page 66),
- le changement c_{1nv-1v} (définition 4.16 page 66),
- le changement *c-conservatif* (définition 4.17 page 66),
- le changement *c-expansif* (définition 4.18 page 67),
- le changement *c-limitatif* (définition 4.19 page 68),
- le changement *c-modifiant* (définition 4.20 page 69).

Les tables 4.3 et 4.4 page suivante présentent la classification de l'ensemble de ces sous-cas.

Le deuxième niveau abordé est celui des **ensembles d'arguments** (section 4.2.2 page 70), où nous nous intéressons à la notion de *monotonie*, c'est-à-dire au fait que tout ensemble d'arguments précédemment accepté l'est toujours après un changement ou, au contraire, que tout ensemble d'arguments accepté après un changement l'était déjà avant.

Ainsi, nous distinguons deux grands types de monotonie, chacun d'entre eux possédant trois formes plus précises suivant le critère d'acceptabilité des arguments :

- la monotonie expansive qui recouvre 3 cas :
  - la *monotonie expansive simple* (définition 4.21 page 70),
  - la *monotonie expansive crédule* (définition 4.23 page 70),

Système initial	Système final $\|\mathbf{E}'\| = 0$	$\|\mathbf{E}'\| = 1$, $\mathcal{E}' = \varnothing$	$\|\mathbf{E}'\| = 1$, $\mathcal{E}' \neq \varnothing$	$\|\mathbf{E}'\| > 1$
$\|\mathbf{E}\| = 0$	*c-conservatif*	∇	e_{0-1nv}	e_{0-k}
$\|\mathbf{E}\| = 1$, $\mathcal{E} = \varnothing$	∇	*c-conservatif*	c_{1v-1nv}	e_{1v-k}
$\|\mathbf{E}\| = 1$, $\mathcal{E} \neq \varnothing$	r_{1nv-0}	c_{1nv-1v}	Voir table 4.4	e_{1nv-k}
$\|\mathbf{E}\| > 1$	r_{k-0}	r_{k-1v}	r_{k-1nv}	Si $\|\mathbf{E}\| < \|\mathbf{E}'\|$: e_{j-k} Si $\|\mathbf{E}\| = \|\mathbf{E}'\|$: Voir table 4.4 Si $\|\mathbf{E}\| > \|\mathbf{E}'\|$: r_{k-j}

∇. Cas ne pouvant se produire que par un changement de sémantique ou en autorisant un système d'argumentation vide et des arguments s'attaquant eux-mêmes, ce que nous ne considérons pas ici.

Table 4.3 – *Classification des propriétés.*

Toutes les extensions restent les mêmes	***c-conservatif***
Toutes les extensions sont augmentées de quelques arguments et n'étaient pas vides au départ	***c-expansif***
Toutes les extensions sont diminuées de quelques arguments mais sans devenir vides	***c-limitatif***
Pour tout autre cas de changement constant non encore répertorié	***c-modifiant***

Table 4.4 – *Récapitulatif des propriétés structurelles lorsque* $|\mathbf{E}| = |\mathbf{E}'|$.

  - la *monotonie expansive sceptique* (définition 4.25 page 71),
- la monotonie restrictive qui recouvre 3 cas :
  - la *monotonie restrictive simple* (définition 4.22 page 70),
  - la *monotonie restrictive crédule* (définition 4.24 page 71),
  - la *monotonie restrictive sceptique* (définition 4.26 page 71).

La table 4.5 page suivante résume les différentes propriétés de monotonie.

	Monotonie expansive	Monotonie restrictive
Simple	$\forall \mathcal{E}_i \in \mathbf{E}, \exists \mathcal{E}'_j \in \mathbf{E}',$ $\mathcal{E}_i \subseteq \mathcal{E}'_j$	$\forall \mathcal{E}'_j \in \mathbf{E}', \exists \mathcal{E}_i \in \mathbf{E},$ $\mathcal{E}'_j \subseteq \mathcal{E}_i$
Crédule	$\bigcup\limits_{1 \leq i \leq \lvert\mathbf{E}\rvert} \mathcal{E}_i \subseteq \bigcup\limits_{1 \leq j \leq \lvert\mathbf{E}'\rvert} \mathcal{E}'_j$	$\bigcup\limits_{1 \leq j \leq \lvert\mathbf{E}'\rvert} \mathcal{E}'_j \subseteq \bigcup\limits_{1 \leq i \leq \lvert\mathbf{E}\rvert} \mathcal{E}_i$
Sceptique	$\bigcap\limits_{1 \leq i \leq \lvert\mathbf{E}\rvert} \mathcal{E}_i \subseteq \bigcap\limits_{1 \leq j \leq \lvert\mathbf{E}'\rvert} \mathcal{E}'_j$	$\bigcap\limits_{1 \leq j \leq \lvert\mathbf{E}'\rvert} \mathcal{E}'_j \subseteq \bigcap\limits_{1 \leq i \leq \lvert\mathbf{E}\rvert} \mathcal{E}_i$

Table 4.5 – *Récapitulatif des différentes propriétés de monotonie.*

Le troisième et dernier niveau abordé est celui de **l'argument lui-même** (section 4.2.3 page 73), où nous considérons spécifiquement le statut d'un argument particulier et étudions l'évolution de son acceptabilité (ou de son rejet), *i.e.*, s'il conserve son statut ou passe d'un statut à un autre.

Nous avons dénombré neuf cas différents, suivant que l'argument étudié appartient à aucune, plusieurs ou toutes les extensions avant et après le changement. Ainsi, nous avons :

- l'*instauration d'acceptabilité* uniquement partielle (définition 4.27.1 page 73),
- l'*instauration d'acceptabilité* totale (définition 4.27.2 page 73),
- l'*élimination d'acceptabilité* uniquement crédule (définition 4.28.1 page 73),
- l'*élimination d'acceptabilité* sceptique (définition 4.28.2 page 73),
- la *diffusion générale d'acceptabilité* (définition 4.29 page 73),
- la *dégradation partielle d'acceptabilité* (définition 4.30 page 74),
- la *conservation de l'acceptabilité* uniquement crédule (définition 4.31.1 page 74),
- la *conservation de l'acceptabilité* sceptique (définition 4.31.2 page 74),
- la *conservation du rejet* (définition 4.31.3 page 74).

La table 4.6 page suivante présente la classification des propriétés concernant le statut d'un argument.

Travaux connexes

Notons que l'obtention d'un système d'argumentation à partir de la description des extensions souhaitées a été étudiée par Woltran (2013); Linsbichler (2013). Il s'avère que pour un ensemble d'extensions donné, il n'est pas toujours possible de trouver un système d'argumentation qui corresponde. En plaçant ces recherches dans notre typologie, cela signifie donc qu'il existe des cas où aucun changement ne permet d'obtenir une modification particulière. Il pourrait être intéressant de modifier notre typologie pour tenir compte de ce genre d'informations.

Système initial	Système final		
	$\lvert\mathbf{E}'_x\rvert = 0$	$0 < \lvert\mathbf{E}'_x\rvert < \lvert\mathbf{E}'\rvert$	$0 < \lvert\mathbf{E}'_x\rvert = \lvert\mathbf{E}'\rvert$
$\lvert\mathbf{E}_x\rvert = 0$	*Conservation du rejet*	*Instauration d'acceptabilité* uniquement partielle	*Instauration d'acceptabilité* totale
$0 < \lvert\mathbf{E}_x\rvert < \lvert\mathbf{E}\rvert$	*Élimination d'acceptabilité* uniquement crédule	*Conservation de l'acceptabilité* uniquement crédule	*Diffusion générale d'acceptabilité*
$0 < \lvert\mathbf{E}_x\rvert = \lvert\mathbf{E}\rvert$	*Élimination d'acceptabilité* sceptique	*Dégradation partielle d'acceptabilité*	*Conservation de l'acceptabilité* sceptique

Table 4.6 – *Classification des différentes propriétés concernant le statut d'un argument particulier.*

Maintenant que les propriétés du changement sont clairement définies et organisées, il est temps de les étudier. Ainsi, le chapitre suivant est consacré à leur caractérisation. Pour cela, outre des caractérisations directes pour chacune des opérations d'addition et de suppression d'un argument, nous allons utiliser le lien unissant ces deux opérations pour définir une méthode permettant de réutiliser des résultats concernant une opération pour obtenir de nouveaux résultats concernant l'autre opération.

Chapitre 5

Caractérisations

Le serpent qui ne peut changer de peau, meurt. Il en va de même des esprits que l'on empêche de changer d'opinion : ils cessent d'être esprit.

Friedrich Nietzsche,
Aurore.

Sommaire

Nous nous intéressons maintenant à l'étude des propriétés du changement et, en particulier, à leur caractérisation. Pour cela, nous fournissons des conditions sous lesquelles ces propriétés sont satisfaites étant donné une opération de changement.

Dans ce cadre, il nous a paru important d'exploiter le lien entre addition et suppression afin de compléter la caractérisation de la suppression grâce au travail fait précédemment sur l'addition, et inversement. L'exemple suivant montre que quelques connaissances à propos de ce lien peuvent permettre de bénéficier de résultats déjà connus :

> *Monsieur Pink sait que l'énonciation d'un simple argument pourrait détruire l'argumentation de Monsieur White. Mais cet argument lui manque. Une autre façon de gagner ce débat serait de supprimer un des arguments de Monsieur White ; il pourrait prétexter qu'il est fallacieux. Malheureusement, Monsieur Pink ne connaît pas les conséquences de cette suppression.*

Aussi, dans ce chapitre, nous nous attelons à étudier le lien entre addition et suppression d'un argument, lien très peu traité à notre connaissance, à travers diverses notions de dualité dans la section 5.2 page 84 qui nous serviront à enrichir nos caractérisations. Nous présentons ensuite les résultats de caractérisation obtenus directement pour l'opération d'addition (section 5.3.1 page 89) et l'opération de suppression (section 5.3.2 page 92). Nous faisons de même pour les résultats obtenus grâce à la dualité pour l'opération d'addition (section 5.4.1 page 94) ainsi que pour l'opération de suppression (section 5.4.2 page 96). La section 5.5 page 99 propose quant à elle des tables synthétisant tous nos résultats de caractérisation. Enfin, la section 5.6 page 103 conclut ce chapitre en décrivant leurs utilisations possibles. Notons que le lecteur trouvera dans l'annexe C.2 page 199 les preuves des résultats de caractérisation.

Remarquons, à titre d'avant-propos, que ce que nous appelons *opération de changement* dans ce chapitre correspond en fait à une *opération élémentaire de changement* telle qu'elle apparaît dans la définition 2.6 page 16. En outre, ce chapitre reprend et augmente les résultats de Bisquert (2011); Bisquert *et al.* (2012a,b, 2013b).

5.1 Résultats préliminaires

Dans cette section, nous proposons des résultats généraux de caractérisation pour l'addition et la suppression d'un argument dans le cadre de plusieurs sémantiques (certains résultats sont repris de Cayrol *et al.* (2010c)). Le premier de ces résultats est dû au fait que l'extension basique existe toujours et est unique (voir la propriété 2.1.1 page 14).

Proposition 5.1. *Dans le cadre de la sémantique basique, un changement (addition ou suppression) n'est jamais* extensif *(*e_{0-1nv}*,* e_{0-k}*,* e_{1v-k}*,* e_{1nv-k}*,* e_{j-k}*) ni* restrictif *(*r_{k-0}*,* r_{1nv-0}*,* r_{k-j}*,* r_{k-1v}*,* r_{k-1nv}*).*

Le second résultat est dû au fait qu'il existe toujours une extension préférée (voir la propriété 2.1.1 page 14).

Proposition 5.2. *Dans le cadre de la sémantique préférée, un changement (addition ou suppression) n'est jamais* e_{0-1nv}*,* e_{0-k}*,* r_{k-0} *ou* r_{1nv-0}*.*

La sémantique stable est prise en compte dans le troisième résultat, grâce à la propriété 2.1.8 page 14 et à la supposition que l'ensemble des arguments du système n'est pas vide.

Proposition 5.3. *Dans le cadre de la sémantique stable, un changement (addition ou suppression) n'est jamais* c_{1v-1nv}*,* c_{1nv-1v}*,* e_{1v-k} *ou* r_{k-1v}*.*

La proposition suivante est due, ici aussi, au fait que l'extension basique est unique.

Proposition 5.4.

- *Dans le cadre de la sémantique basique, la* monotonie expansive sceptique *et la* monotonie expansive crédule *correspondent à la* monotonie expansive simple*.*
- *Dans le cadre de la sémantique basique, la* monotonie restrictive sceptique *et la* monotonie restrictive crédule *correspondent à la* monotonie restrictive simple*.*

Selon les définitions de la monotonie données dans la section 4.2 page 61, la proposition suivante est vraie pour chacune des sémantiques considérées dans ce travail.

Proposition 5.5.

- *La* monotonie expansive simple *implique la* conservation de l'acceptabilité uniquement crédule*.*
- *La* monotonie expansive sceptique *implique la* conservation de l'acceptabilité sceptique*.*

La notation et les lemmes suivants seront utilisées pour établir les propositions données dans les sections 5.3.1 à 5.4.2 pages 89–96. Les démonstrations de ces lemmes sont disponibles dans l'annexe C.2.1 page 199.

Soit $\mathcal{N}_z$ l'ensemble d'arguments non attaqués de $\mathcal{G}'$, c'est-à-dire l'ensemble d'arguments différents de z non attaqués par $\mathcal{G} \setminus \{z\}$.

Notation 5.1. *Soit* $z \in \mathcal{G}$, $\mathcal{N}_z = \{x \in \mathcal{G} \setminus \{z\}$ *tel que* x *n'est pas attaqué dans* $\mathcal{G} \setminus \{z\}\}$*.*

Lemme 5.1. *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, z n'attaque pas $\mathcal{E}'$ dans $\mathcal{G}$ si et seulement si $\forall x \in \mathcal{G}'$, si z attaque x dans $\mathcal{G}$ alors (x est attaqué par $\mathcal{G} \setminus \{z\}$ et x n'est pas défendu indirectement par $\mathcal{N}_z$ dans $\mathcal{G} \setminus \{z\}$).*

Lemme 5.2. *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si z n'attaque pas $\mathcal{E}'$, alors l'équivalence suivante est vraie : $z \in \bigcup_{i \geq 1} \mathcal{F}^i(\mathcal{N}_z)$ si et seulement si $\mathcal{E}'$ défend z dans $\mathcal{G}$.*

Lemme 5.3.

1. *Lors de l'addition d'un argument z, z n'attaque pas $\mathcal{G}'$ (resp. n'est pas attaqué par $\mathcal{G}'$) si et seulement si z n'attaque pas $\mathcal{G}$ (resp. n'est pas attaqué par $\mathcal{G}$) dans $\mathcal{G}'$.*
2. *Lors de la suppression d'un argument z, z n'attaque pas $\mathcal{G}'$ (resp. n'est pas attaqué par $\mathcal{G}'$) dans $\mathcal{G}$ si et seulement si z n'attaque pas $\mathcal{G}$ (resp. n'est pas attaqué par $\mathcal{G}$).*

Lemme 5.4.

1. *Lors de l'addition d'un argument z dans $\mathcal{G}$, soit $x \neq z$ un argument de $\mathcal{G}$. Si x est attaqué dans $\mathcal{G}$ alors x est aussi attaqué dans $\mathcal{G}'$ par un argument distinct de z. Si x est attaqué dans $\mathcal{G}'$ alors x est soit attaqué dans $\mathcal{G}$, soit attaqué seulement par z dans $\mathcal{G}$.*
2. *Lors de la suppression d'un argument z de $\mathcal{G}$, soit $x \neq z$ un argument de $\mathcal{G}$. Si x est attaqué dans $\mathcal{G}'$ alors x est aussi attaqué dans $\mathcal{G}$ par un argument distinct de z. Si x est attaqué dans $\mathcal{G}$ alors x est soit attaqué dans $\mathcal{G}'$, soit attaqué seulement par z dans $\mathcal{G}'$.*

5.2 Dualités

À notre connaissance, la suppression d'un argument et, *a fortiori*, le lien entre addition et suppression d'un argument sont des points qui n'ont été que très peu abordés. Pourtant, il nous semble dommage de considérer addition et suppression de façon indépendante. Nous pensons en effet qu'il peut être profitable de tirer parti des liens entre ces opérations pour l'étude des propriétés caractérisant les changements qui peuvent impacter un système d'argumentation. Pour ce faire, nous nous aiderons de la notion de *dualité* que nous définissons dans la section 5.2.1 et pour laquelle nous donnons quelques résultats directs dans la section 5.2.2 page suivante. Enfin, nous proposons dans la section 5.2.3 page 87 une méthode utilisant la notion dualité pour aider à la caractérisation des propriétés du changement.

5.2.1 Deux définitions de la dualité

Nous nous intéressons à deux concepts de dualité. Nous définissons en premier lieu une dualité au niveau des opérations de changement, la *dualité basée sur la* *notion d'inverse*, traduisant le fait que deux opérations peuvent être considérées inverses l'une de l'autre, puis une dualité au niveau des propriétés d'un changement, la *dualité basée sur la notion de symétrie*, traduisant une correspondance entre deux propriétés.

Définition 5.1 (**Dualité basée sur la notion d'inverse**). *Deux opérations de changement o et o' sont* inverses *l'une de l'autre si et seulement si :*
$$\forall \mathcal{G}, \forall \mathcal{G}',\ o(\mathcal{G}) = \mathcal{G}' \text{ si et seulement si } o'(\mathcal{G}') = \mathcal{G}.$$

Selon la définition 5.1, il est clair que les opérations d'addition et de suppression d'un argument au sens de la définition 2.6 page 16 sont inverses l'une de l'autre. Pour l'illustrer, reprenons l'exemple 2.3 page 17 :

Exemple 2.3 (cont.). *Soit $o_1 = \ominus(z, \mathcal{R}_z)$ avec $\mathcal{R}_z = \{(z, x), (z, y)\}$ et $o'_1 = \oplus(z, \mathcal{R}_z)$ avec $\mathcal{R}_z = \{(z, x), (z, y)\}$; o_1 et o'_1 sont inverses l'une de l'autre pour tout graphe (rappelons que chaque opération spécifie totalement l'argument et les interactions ajoutées ou supprimées).*

Définition 5.2 (**Dualité basée sur la notion de symétrie**). *Deux propriétés π et π' sont* symétriques *si et seulement si :*
$$\forall \mathcal{G},\ \forall \mathcal{G}',\ \pi'(\mathcal{G}', \mathcal{G}) \text{ si et seulement si } \pi(\mathcal{G}, \mathcal{G}').$$

De ces définitions, nous pouvons tirer une condition pour la satisfaction d'une propriété par une opération de changement.

Proposition 5.6. *Soit o et o' deux opérations de changement inverses l'une de l'autre et π et π' deux propriétés symétriques. o satisfait π si et seulement si o' satisfait π'.*

La démonstration de la proposition 5.6 est disponible dans l'annexe C.2.2 page 201.

5.2.2 Résultats de dualité

Nous reprenons dans cette section les propriétés du changement de la section 4.2 page 61 et nous les étudions sous l'angle des deux concepts de dualité. Toutes les démonstrations se trouvent en annexe C.2.2 page 201.

5.2.2.1 Propriétés concernant l'ensemble des extensions

Proposition 5.7.

1. *Un changement est* restrictif *si et seulement si le changement inverse est* extensif.
2. *Un changement est r_{1nv-0} si et seulement si le changement inverse est e_{0-1nv}.*
3. *Un changement est r_{k-0} si et seulement si le changement inverse est e_{0-k}.*
4. *Un changement est r_{k-1v} si et seulement si le changement inverse est e_{1v-k}.*

5. *Un changement est* r_{k-1nv} *si et seulement si le changement inverse est* e_{1nv-k}.
6. *Un changement est* r_{k-j} *si et seulement si le changement inverse est* e_{j-k}.

Proposition 5.8.

1. *Un changement est* constant *si et seulement si le changement inverse l'est aussi.*
2. *Un changement est* c_{1nv-1v} *si et seulement si le changement inverse est* c_{1v-1nv}.
3. *Un changement est* c-conservatif *si et seulement si le changement inverse l'est aussi.*
4. *Un changement est* c-limitatif *si et seulement si le changement inverse est* c-expansif.
5. *Un changement est* c-modifiant *si et seulement si le changement inverse l'est aussi.*

La figure 5.1 récapitule toutes les propositions précédentes.

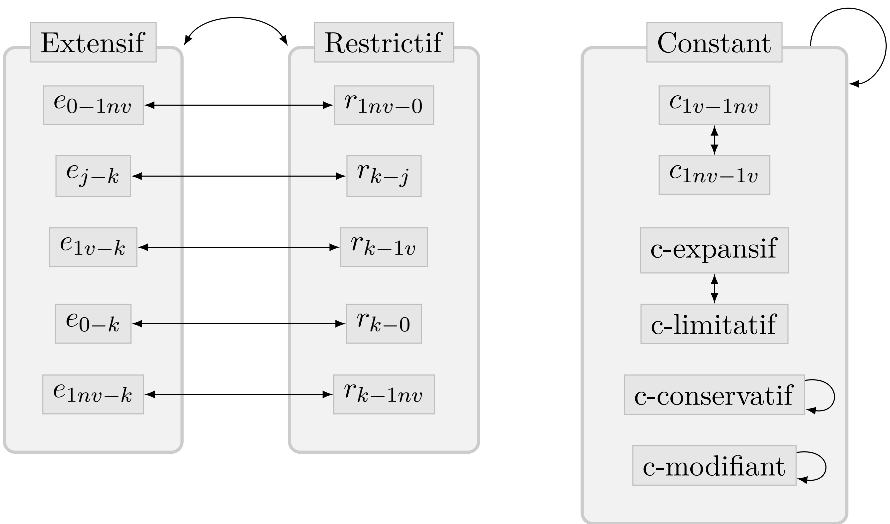


Figure 5.1 – *Présentation de la dualité entre les changements* restrictif *et* extensif *d'une part, et* constant *d'autre part, ainsi que leurs cas particuliers (la flèche signifie "est le dual de").*

5.2.2.2 Propriétés concernant les ensembles d'arguments

Proposition 5.9.

1. *Un changement satisfait la* monotonie restrictive simple *si et seulement si le changement inverse satisfait la* monotonie expansive simple.
2. *Un changement satisfait la* monotonie restrictive crédule *si et seulement si le changement inverse satisfait la* monotonie expansive crédule.

3. *Un changement satisfait la* monotonie restrictive sceptique *si et seulement si le changement inverse satisfait la* monotonie expansive sceptique.

La figure 5.2 récapitule les résultats sur la dualité concernant la monotonie.

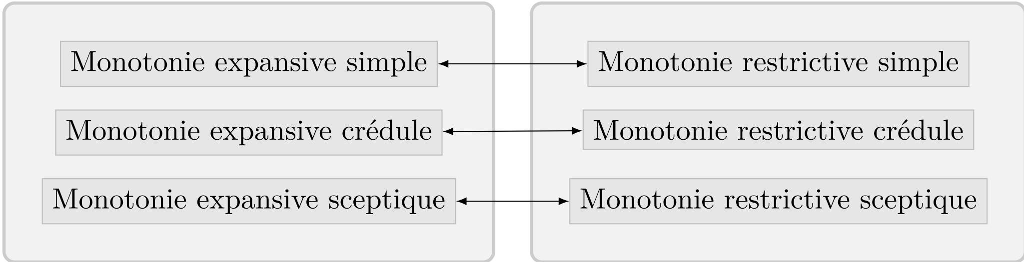


Figure 5.2 – *Récapitulatif des différentes relations de dualité entre les propriétés concernant le statut d'ensembles d'arguments (la flèche signifie "est le dual de").*

5.2.2.3 Propriétés concernant le statut d'un argument

Proposition 5.10.

1. *Un changement* instaure l'acceptabilité uniquement partielle *pour un argument x si et seulement si le changement inverse* élimine l'acceptabilité uniquement crédule *de x.*
2. *Un changement* instaure l'acceptabilité totale *pour un argument x si et seulement si le changement inverse* élimine l'acceptabilité sceptique *de x.*

Proposition 5.11. *Un changement est une* diffusion générale d'acceptabilité *pour un argument x si et seulement si le changement inverse* dégrade partiellement l'acceptabilité *pour x.*

Proposition 5.12.

1. *Un changement* conserve l'acceptabilité uniquement crédule *pour un argument x si et seulement si le changement inverse* conserve l'acceptabilité uniquement crédule *pour x.*
2. *Un changement* conserve l'acceptabilité sceptique *pour un argument x si et seulement si le changement inverse* conserve l'acceptabilité sceptique *pour x.*
3. *Un changement* conserve le rejet *pour un argument x si et seulement si le changement inverse* conserve le rejet *pour x.*

La figure 5.3 page suivante récapitule les propositions précédentes.

5.2.3 Méthodologie d'utilisation de la dualité

Cette partie décrit comment utiliser la dualité de façon à obtenir de nouvelles propositions pour l'opération de suppression, à partir des propositions concernant

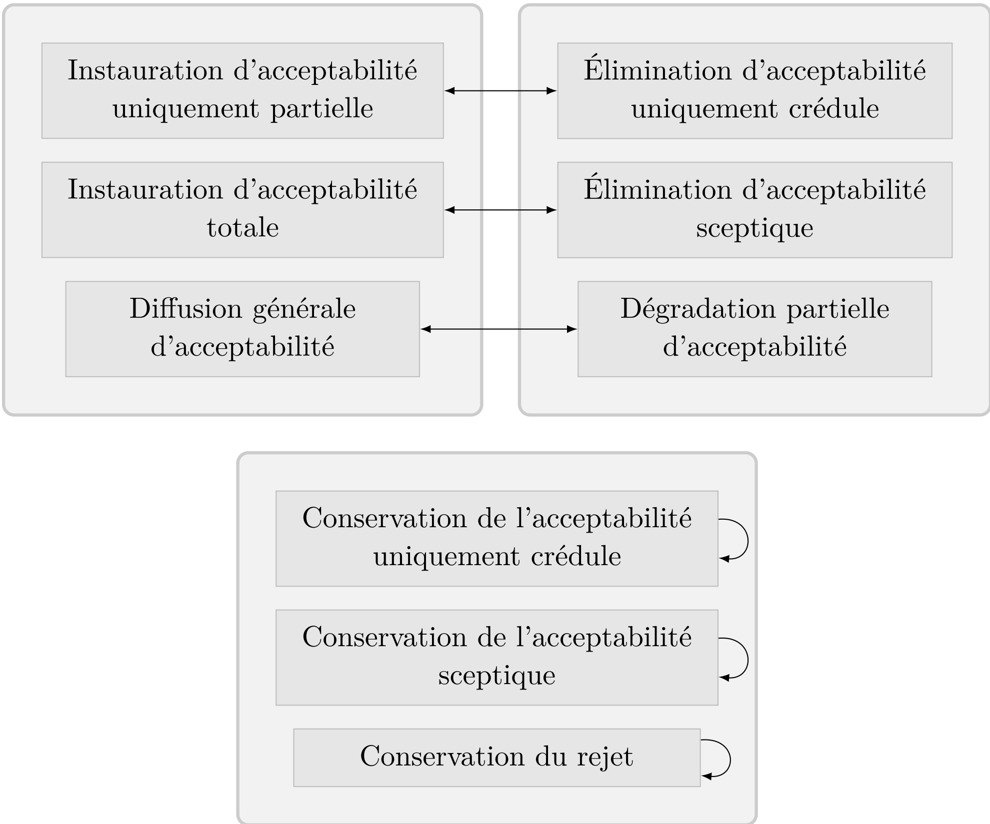


Figure 5.3 – *Récapitulatif des différentes relations de dualité entre les propriétés concernant le statut d'un argument particulier (la flèche signifie "est le dual de").*

l'addition (ou vice versa). Notons que nous présentons cette méthodologie dans le cadre de la sémantique basique. Décrivons-là en utilisant la proposition suivante. Celle-ci assure que lorsque l'on ajoute un argument (dans le cadre de la sémantique basique) qui ne défend pas un argument n'appartenant à aucune extension, ce dernier n'appartiendra toujours à aucune extension après le changement. La démonstration de cette nouvelle proposition est disponible dans l'annexe C.2.2.

Proposition 5.13. *Lors de l'addition d'un argument z dans le cadre de la sémantique basique, $\forall x \in \mathcal{G}$, si $x \notin \mathcal{E}$ et z ne défend pas indirectement x, alors $x \notin \mathcal{E}'$.*

De façon à clarifier cette présentation, les graphes et extensions sont renommés en ajoutant deux lettres majuscules en indice - EA, SA, ES et SS - représentant respectivement le système d'**E**ntrée pour l'**A**ddition, le système de **S**ortie pour l'**A**ddition, le système d'**E**ntrée pour la **S**uppression et le système de **S**ortie pour la **S**uppression. Ainsi, la proposition 5.13 peut être réécrite comme suit :

Proposition 5.13.1. *Lors de l'addition d'un argument z dans le cadre de la sémantique basique, si $x \notin \mathcal{E}_{EA}$ et z ne défend pas indirectement x, alors $x \notin \mathcal{E}_{SA}$.*

Soit π une propriété du changement et π^{-1} sa propriété symétrique. Grâce à la proposition 5.6 page 85, il est vrai que : $\oplus$ *satisfait* π *ssi* $\ominus$ *satisfait* π^{-1}.

Et grâce à la définition 4.1 page 60, nous savons qu'une opération de changement o satisfait π si et seulement si $\forall \mathcal{G}$, il est vrai que $\pi(\mathcal{G}, o(\mathcal{G}))$. Donc :

$$\forall \mathcal{G}_{EA}, \pi(\mathcal{G}_{EA}, \bigoplus(\mathcal{G}_{EA})) \textit{ est vraie ssi } \forall \mathcal{G}_{ES},\ \pi^{-1}(\mathcal{G}_{ES}, \bigominus(\mathcal{G}_{ES})) \textit{ est vraie.}$$

De plus, grâce à la définition 5.2 page 85, nous avons :

$$\forall \mathcal{G}_{ES}, \pi^{-1}(\mathcal{G}_{ES}, \bigominus(\mathcal{G}_{ES})) \textit{ est vraie ssi } \pi(\bigominus(\mathcal{G}_{ES}), \mathcal{G}_{ES}) \textit{ est vraie.}$$

Ainsi, nous avons :

$$\forall \mathcal{G}_{EA},\ \forall \mathcal{G}_{ES},\ \pi(\mathcal{G}_{EA}, \bigoplus(\mathcal{G}_{EA})) \textit{ est vraie ssi } \pi(\bigominus(\mathcal{G}_{ES}), \mathcal{G}_{ES}) \textit{ est vraie.}$$

Soit $\mathcal{G}_{SA} = \bigoplus(\mathcal{G}_{EA})$ et $\mathcal{G}_{SS} = \bigominus(\mathcal{G}_{ES})$. Puisque nous savons que π est satisfaite pour l'opération d'addition, nous pouvons la réécrire pour la suppression :

Proposition 5.13.2. *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si $x \notin \mathcal{E}_{SS}$ et z ne défend pas indirectement x, alors $x \notin \mathcal{E}_{ES}$.*

Ce qui est équivalent à :

Proposition 5.13.3. *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si $x \in \mathcal{E}_{ES}$ et z ne défend pas indirectement x, alors $x \in \mathcal{E}_{SS}$.*

Ainsi, pour l'opération de suppression, nous obtenons une proposition analogue à la proposition 5.13 page ci-contre notée proposition $5.13^{\ominus}$; dans la suite de ce travail, l'exposant ($^{\oplus}$ ou $^{\ominus}$) représentera la correspondance entre une proposition et celle obtenue en appliquant la méthodologie de la dualité :

Proposition $5.13^{\ominus}$. *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si $x \in \mathcal{E}$ et z ne défend pas indirectement x, alors $x \in \mathcal{E}'$.*

Dans les sections suivantes, après avoir donné des propositions concernant les opérations d'addition et de suppression d'un argument obtenues directement, nous utiliserons cette méthodologie de concert avec les lemmes présentés dans la section 5.1 page 82 de façon à obtenir de nouveaux résultats.

5.3 Résultats directs

5.3.1 Caractérisations dans le cas de l'addition d'argument

Parmi les résultats donnés ici, les propositions 5.13 page ci-contre, 5.20 page 91 et 5.21 page 91 sont nouvelles.

Les propositions 5.17 page suivante, 5.18 page suivante et 5.19 page 91 proposent une généralisation des propositions 10, 11.1, 11.2.1, 11.2.2 et 13 de Cayrol *et al.* (2010c). La preuve fournie dans l'annexe C.2.3 page 202 prend en compte diverses propriétés concernant l'addition de l'argument z dans le cas spécial où $\mathcal{E} = \varnothing$. Ces propriétés nous permettent de supprimer la condition $\mathcal{E} \neq \varnothing$ des propositions initiales.

Enfin, les propositions 5.14, 5.15 et 5.16 sont directement reprises de Cayrol *et al.* (2010c) (la numérotation originelle de ces propositions est donnée entre parenthèses).

Les résultats donnés ici ne concernent que deux sémantiques : la sémantique basique et la sémantique préférée. Les démonstrations sont données dans l'annexe C.2.3 page 202.

5.3.1.1 Résultats pour la sémantique basique

Soit $x \in \mathcal{G} \cap \mathcal{G}'$, la *conservation du rejet* de x est caractérisée par la proposition suivante (utilisée comme base pour illustrer notre méthodologie dans la section 5.2.3 page 87).

Proposition 5.13. *Lors de l'addition d'un argument z dans le cadre de la sémantique basique, $\forall x \in \mathcal{G}$, si $x \notin \mathcal{E}$ et z ne défend pas indirectement x, alors $x \notin \mathcal{E}'$.*

Proposition 5.14 (Proposition 7 de Cayrol *et al.* (2010c))**.** *Dans le cadre de la sémantique basique, si $x \in \mathcal{E}$ et z n'attaque pas indirectement x, alors $\oplus$* conserve le statut *de x (*i.e. *$x \in \mathcal{E}'$).*

Proposition 5.15 (Proposition 8 de Cayrol *et al.* (2010c))**.** *Dans le cadre de la sémantique basique, si z n'est pas attaqué par $\mathcal{G}$, alors $\oplus$ satisfait la* priorité à la nouveauté *(*i.e. *$z \in \mathcal{E}'$).*

Proposition 5.16 (Proposition 9 de Cayrol *et al.* (2010c))**.** *Dans le cadre de la sémantique basique, dans le cas d'une addition,*

1. *si $\mathcal{E} = \{\}$, alors il est vrai que : $\mathcal{E}' = \{\}$ si et seulement si z est attaqué par $\mathcal{G}$, de plus,*
2. *si $\mathcal{E} = \{\}$ et z n'est pas attaqué par $\mathcal{G}$, alors $\mathcal{E}' = \{z\} \cup \bigcup_{i \geq 1} \mathcal{F}'^i(\{z\})$.*

Notons que la proposition 5.16.1 implique que "si z est attaqué par $\mathcal{G}$ et $\mathcal{E} = \{\}$, alors $\mathcal{E}' = \{\}$". Ainsi, "z est attaqué par $\mathcal{G}$ et $\mathcal{E} = \{\}$" est une condition suffisante pour avoir un changement *c-conservatif*. Les deux propositions suivantes sont des extensions de propositions de Cayrol *et al.* (2010c) permettant de couvrir le cas $\mathcal{E} = \emptyset$.

Proposition 5.17. *Dans le cadre de la sémantique basique, si z n'attaque pas $\mathcal{E}$, alors $\oplus$ satisfait la* monotonie expansive simple *(*i.e. *$\mathcal{E} \subseteq \mathcal{E}'$).*

Proposition 5.18. *Dans le cadre de la sémantique basique, si z n'attaque pas $\mathcal{E}$, nous avons :*

1. *si $\mathcal{E}$ ne défend pas z, alors $\mathcal{E}' = \mathcal{E}$ (le changement $\oplus$ est* c-conservatif*).*
2. *si $\mathcal{E}$ défend z, alors $\mathcal{E}' = \mathcal{E} \cup \{z\} \cup \bigcup_{i \geq 1} \mathcal{F}'^i(\{z\})$.*
3. *De plus, si $\mathcal{E}$ défend z et z n'attaque pas $\mathcal{G}$, alors $\mathcal{E}'$ se réduit à $\mathcal{E} \cup \{z\}$.*

Dans les deux derniers cas, le changement $\oplus$ est c-expansif *si $\mathcal{E} \neq \{\}$, c_{1v-1nv} sinon.*

Soit $x \in \mathcal{G} \cap \mathcal{G}'$, une conséquence de la proposition 5.18 page précédente donne la caractérisation (condition suffisante) de l'*instauration d'acceptabilité* de x :

Conséquence 5.1. *Lors de l'addition d'un argument z dans le cadre de la sémantique basique, si z n'attaque pas $\mathcal{E}$ et $\mathcal{E}$ défend z et z défend indirectement x et $x \notin \mathcal{E}$, alors $x \in \mathcal{E}'$.*

Proposition 5.19. *Dans le cadre de la sémantique basique, si z attaque chaque argument non attaqué de $\mathcal{G}$ et z est attaqué par $\mathcal{G}$, alors le changement $\bigoplus$ est c_{1nv-1v}.*

Soit $x \in \mathcal{G} \cap \mathcal{G}'$, les résultats suivants donnent respectivement une caractérisation de l'*instauration d'acceptabilité* de x et une caractérisation de l'*élimination d'acceptabilité* de x.

Proposition 5.20. *Lors de l'addition d'un argument z dans le cadre de la sémantique basique, si z n'est pas attaqué par $\mathcal{G}$ et z défend indirectement x et $x \notin \mathcal{E}$, alors $x \in \mathcal{E}'$.*

Proposition 5.21. *Lors de l'addition d'un argument z dans le cadre de la sémantique basique, si $\mathcal{E} \setminus \{x\}$ n'attaque pas z et z attaque x et $x \in \mathcal{E}$, alors $x \notin \mathcal{E}'$.*

5.3.1.2 Résultats pour la sémantique préférée

Proposition 5.22. *Dans le cadre de la sémantique préférée, si z n'est pas attaqué par $\mathcal{G}$, alors $\bigoplus$ satisfait la* priorité sceptique à la nouveauté *(*i.e. $\forall i$, $z \in \mathcal{E}'_i$*).*

Proposition 5.23. *Lors de l'addition d'un argument z dans le cadre de la sémantique préférée,*

1. *si z n'attaque pas $\mathcal{E}_i$, alors $\mathcal{E}_i$ reste admissible dans $\mathcal{G}'$;*
2. *si z n'attaque pas $\mathcal{E}_i$ et $\mathcal{E}_i$ défend z dans $\mathcal{G}'$, alors $\mathcal{E}_i \cup \{z\}$ est admissible dans $\mathcal{G}'$.*

Proposition 5.24. *Lors de l'addition d'un argument z dans le cadre de la sémantique préférée, si $\mathbf{E} = \{\{\}\}$ et z n'est pas attaqué par $\mathcal{G}$ et il n'existe aucun cycle de longueur paire dans $\mathcal{G}$, alors $\mathbf{E}' = \{\mathcal{E}'\}$ et $z \in \mathcal{E}'$ (donc $\bigoplus$ est c_{1v-1nv}).*

Proposition 5.25. *Lors de l'addition d'un argument z dans le cadre de la sémantique préférée, si z n'attaque aucun argument de $\mathcal{G}$ et $\mathbf{E} = \{\{\}\}$, alors $\mathbf{E}' = \{\{\}\}$; ou, de façon équivalente, si $\mathbf{E} = \{\{\}\}$, le changement $\bigoplus$ par z est c_{1v-1nv} seulement si z attaque $\mathcal{G}$.*

Notons que la proposition précédente donne une condition suffisante pour avoir un changement *c-conservatif* : "z n'attaque aucun argument de $\mathcal{G}$ et $\mathbf{E} = \{\{\}\}$".

Proposition 5.26. *Lors de l'addition d'un argument z dans le cadre de la sémantique préférée, si z n'attaque aucun argument de $\mathcal{G}$ et $\mathbf{E} \neq \{\{\}\}$, alors pour tout i :*

1. *si $\mathcal{E}_i$ défend z, alors $\mathcal{E}_i \cup \{z\}$ est une extension de $\mathcal{G}'$;*
2. *si $\mathcal{E}_i$ ne défend pas z, alors $\mathcal{E}_i$ est une extension de $\mathcal{G}'$.*

De plus, $\mathcal{G}$ et $\mathcal{G}'$ ont le même nombre d'extensions ($|\mathbf{E}| = |\mathbf{E}'|$), donc le changement est constant*.*

Notons encore que la proposition 5.26.2 donne une condition suffisante pour avoir un changement *c-conservatif* : “z n'attaque aucun argument de $\mathcal{G}$ et $\mathbf{E} \neq \{\{\}\}$ et pour tout i, $\mathcal{E}_i$ ne défend pas z”.

Proposition 5.27. *Dans le cadre de la sémantique préférée, si z n'attaque aucune extension de $\mathcal{G}$, alors le changement $\bigoplus$ satisfait la* monotonie expansive simple*.*

Proposition 5.28. *Dans le cadre de la sémantique préférée, supposons que $\mathcal{G}$ ne contienne aucun argument controversé*[1]*. Si z n'attaque pas $\bigcap_{i\geq 1} \mathcal{E}_i$, alors le changement $\bigoplus$ satisfait la* monotonie expansive sceptique*, c'est-à-dire $\bigcap_{i\geq 1} \mathcal{E}_i \subseteq \bigcap_{i\geq 1} \mathcal{E}'_i$.*

Proposition 5.29. *Dans le cadre de la sémantique préférée, si $\mathbf{E} \neq \{\{\}\}$ et il n'existe aucun cycle de longueur paire dans $\mathcal{G}'$ et chaque argument non attaqué de $\mathcal{G}$ est attaqué dans $\mathcal{G}'$ et z est attaqué dans $\mathcal{G}'$, alors le changement $\bigoplus$ est c_{1nv-1v}.*

5.3.2 Caractérisations dans le cas de la suppression d'argument

5.3.2.1 Résultats pour les trois sémantiques

Les résultats donnés ici concernent les trois sémantiques classiques de l'argumentation abstraite : les sémantiques préférée, basique et stable. Tous les résultats sont nouveaux, et leurs démonstrations sont données dans l'annexe C.2.4 page 205.

Proposition 5.30. *Lors de la suppression d'un argument z,*

1. *si $\mathcal{E}$ est une extension préférée de $\mathcal{G}$ et $z \notin \mathcal{E}$, alors $\mathcal{E}$ est admissible dans $\mathcal{G}'$ et donc il existe une extension préférée $\mathcal{E}'$ de $\mathcal{G}'$ telle que $\mathcal{E} \subseteq \mathcal{E}'$.*
2. *si $\mathcal{E}$ est une extension stable de $\mathcal{G}$ et $z \notin \mathcal{E}$, alors $\mathcal{E}$ est stable dans $\mathcal{G}'$.*
3. *si $\mathcal{E}$ est l'extension basique de $\mathcal{G}$ et $z \notin \mathcal{E}$, alors $\mathcal{E} \subseteq \mathcal{E}'$ où $\mathcal{E}'$ est l'extension basique de $\mathcal{G}'$.*

Le résultat suivant donne une condition nécessaire et suffisante pour la *monotonie expansive simple*.

Proposition 5.31. *Lors de la suppression d'un argument z dans le cadre de la sémantique préférée, stable ou basique, il est vrai que :*

$$(\forall \mathcal{E} \in \mathbf{E},\ z \notin \mathcal{E}) \text{ si et seulement si } (\forall \mathcal{E} \in \mathbf{E},\ \exists \mathcal{E}' \in \mathbf{E}' \text{ tel que } \mathcal{E} \subseteq \mathcal{E}').$$

La proposition suivante concerne la notion de *monotonie expansive faible* (*i.e.* une forme de monotonie dans laquelle z n'est pas pris en compte).

1. Un argument u est controversé si et seulement s'il existe au moins un argument v tel que u est à la fois un attaquant indirect (voir définition 2.2 page 12) et un défenseur indirect (voir définition 2.3 page 13) de v.

Proposition 5.32. *Lors de la suppression d'un argument z, si z n'attaque aucun argument dans $\mathcal{G}$, alors*

1. *$\forall \mathcal{E}$ une extension préférée de $\mathcal{G}$, $\mathcal{E} \setminus \{z\}$ est admissible dans $\mathcal{G}'$ et donc $\exists \mathcal{E}'$ une extension préférée de $\mathcal{G}'$ telle que $\mathcal{E} \setminus \{z\} \subseteq \mathcal{E}'$.*
2. *$\forall \mathcal{E}$ une extension stable de $\mathcal{G}$, $\mathcal{E} \setminus \{z\}$ est une extension stable de $\mathcal{G}'$.*
3. *si $\mathcal{E}$ est l'extension basique de $\mathcal{G}$, $\mathcal{E} \setminus \{z\}$ est l'extension basique de $\mathcal{G}'$.*

Proposition 5.33. *Lors de la suppression d'un argument z dans le cadre de la sémantique préférée, stable ou basique, si z n'attaque aucun argument dans $\mathcal{G}$, alors pour toute extension $\mathcal{E}$ de $\mathcal{G}$ telle que $z \notin \mathcal{E}$, $\mathcal{E}$ est une extension de $\mathcal{G}'$.*

Proposition 5.34. *Lors de la suppression d'un argument z dans le cadre de la sémantique préférée, stable ou basique, si le changement est* c-limitatif*, alors il existe une extension $\mathcal{E}$ de $\mathcal{G}$ telle que $z \in \mathcal{E}$.*

5.3.2.2 Résultats ne concernant que certaines sémantiques

Proposition 5.35. *Lors de la suppression d'un argument z dans le cadre de la sémantique préférée, si z n'attaque aucun argument dans $\mathcal{G}$, alors pour toute extension $\mathcal{E}_i$ de $\mathcal{G}$,*

1. *si $z \notin \mathcal{E}_i$ alors $\mathcal{E}_i$ est une extension préférée de $\mathcal{G}'$;*
2. *si $z \in \mathcal{E}_i$ alors $\mathcal{E}_i \setminus \{z\}$ est une extension préférée de $\mathcal{G}'$.*

De plus, $\mathcal{G}$ et $\mathcal{G}'$ ont le même nombre d'extensions ($|\mathbf{E}| = |\mathbf{E}'|$), donc le changement est constant*.*

Proposition 5.36. *Lors de la suppression d'un argument dans le cadre de la sémantique stable, un changement ne peut pas être* c-expansif*.*

Proposition 5.37. *Lors de la suppression d'un argument z dans le cadre de la sémantique préférée ou basique, si ce changement est* c-expansif*, alors*

1. *z n'appartient à aucune extension de $\mathcal{G}$ et*
2. *z attaque au moins un élément de $\mathcal{G}$.*

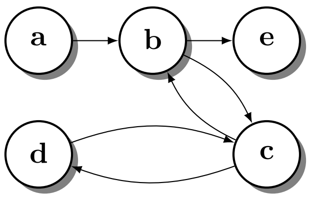
Proposition 5.38. *Lors de la suppression d'un argument z dans le cadre de la sémantique préférée ou basique, si z n'attaque aucun argument de $\mathcal{G}$ et $\forall \mathcal{E}, z \in \mathcal{E}$, alors le changement est* c-limitatif*.*

Soit $x \in \mathcal{G} \cap \mathcal{G}'$; la proposition suivante donne une caractérisation de l'*instauration d'acceptabilité* pour x :

Proposition 5.39. *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si $x \notin \mathcal{E}$ et z est l'unique attaquant de x, alors $x \in \mathcal{E}'$.*

Les conditions de la proposition précédente sont fortes. Cela est dû au fait qu'il est souvent impossible de faire accepter un argument subissant plusieurs attaques avec une seule opération de suppression d'argument, ce qui empêche l'obtention d'un résultat plus général. L'exemple suivant montre ce problème :

Exemple 5.1. *Soit le système d'argumentation suivant dans le cadre de la sémantique basique :*



$\mathbf{E} = \{\{a, e\}\}.$

Il est impossible de faire accepter l'argument b ($b \in \mathcal{E}'$) avec une seule opération de suppression d'argument ; il faut en effet supprimer les arguments a et c. De la même façon, pour faire accepter l'argument c, il est nécessaire de supprimer les arguments b et d.

Soit $x \in \mathcal{G} \cap \mathcal{G}'$, une caractérisation de l'*élimination d'acceptabilité* pour x est donnée par les deux propositions suivantes :

Proposition 5.40. *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si $x \neq z$ et $x \in \mathcal{E}$ et $z \in \mathcal{E}$ et $x \in \bigcup_{i \geq 1} \mathcal{F}^i(\{z\})$ et (il existe un attaquant y de x tel que tout chemin de longueur impaire de $\mathcal{F}(\varnothing)$ à y contient z à une position paire) et (il n'existe aucun chemin de longueur impaire de z à x), alors $x \notin \mathcal{E}'$.*

Proposition 5.41. *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si $x \in \mathcal{E}$ et x est attaqué dans $\mathcal{G}$ et $\forall \mathcal{S}$ tel que $x \in \mathcal{F}(\mathcal{S}), z \in \mathcal{S}$, alors $x \notin \mathcal{E}'$.*

5.4 Résultats par dualité

5.4.1 Caractérisations dans le cas de l'addition d'argument

La démonstration des propositions de cette partie suit la méthodologie décrite dans la section 5.2.3 page 87 ; aucune preuve supplémentaire ne sera donc fournie.

5.4.1.1 Résultats pour les trois sémantiques

Proposition 5.30$^{\oplus}$. *Lors de l'addition d'un argument z,*

1. *si $\mathcal{E}'$ est une extension préférée de $\mathcal{G}'$ et $z \notin \mathcal{E}'$, alors $\mathcal{E}'$ est admissible dans $\mathcal{G}$, donc il existe une extension préférée $\mathcal{E}$ de $\mathcal{G}$ telle que $\mathcal{E}' \subseteq \mathcal{E}$;*
2. *si $\mathcal{E}'$ est une extension stable de $\mathcal{G}'$ et $z \notin \mathcal{E}'$, alors $\mathcal{E}'$ est stable dans $\mathcal{G}$;*
3. *si $\mathcal{E}'$ est l'extension basique de $\mathcal{G}'$ et $z \notin \mathcal{E}'$, alors $\mathcal{E}' \subseteq \mathcal{E}$, où $\mathcal{E}$ est l'extension basique de $\mathcal{G}$.*

Proposition 5.31$^{\oplus}$. *Lors de l'addition d'un argument z dans le cadre de la sémantique préférée, basique ou stable, ($\forall \mathcal{E}' \in \mathbf{E}'$, $z \notin \mathcal{E}'$) si et seulement si ($\forall \mathcal{E}' \in \mathbf{E}', \exists \mathcal{E} \in \mathbf{E}$ tel que $\mathcal{E}' \subseteq \mathcal{E}$).*

Proposition 5.32$^{\oplus}$. *Lors de l'addition d'un argument z, si z n'attaque aucun argument dans $\mathcal{G}$, alors*

1. *$\forall \mathcal{E}'$ une extension préférée de $\mathcal{G}'$, $\mathcal{E}' \setminus \{z\}$ est admissible dans $\mathcal{G}$ et donc $\exists \mathcal{E}$ une extension préférée de $\mathcal{G}$ tel que $\mathcal{E}' \setminus \{z\} \subseteq \mathcal{E}$.*
2. *$\forall \mathcal{E}'$ une extension stable $\mathcal{G}'$, $\mathcal{E}' \setminus \{z\}$ est une extension stable extension de $\mathcal{G}$.*
3. *si $\mathcal{E}'$ est l'extension basique de $\mathcal{G}'$, $\mathcal{E}' \setminus \{z\}$ est l'extension basique de $\mathcal{G}$.*

La proposition 5.32$^{\oplus}$ complète les résultats donnés par les propositions 5.25 page 91, 5.26 page 91 (pour la sémantique préférée) et 5.18 page 90 (pour la sémantique basique).

Proposition 5.33$^{\oplus}$. *Lors de l'addition d'un argument z dans le cadre de la sémantique préférée, basique ou stable, si z n'attaque aucun argument dans $\mathcal{G}$, alors quelle que soit $\mathcal{E}'$ une extension de $\mathcal{G}'$ telle que $z \notin \mathcal{E}'$, $\mathcal{E}'$ est une extension de $\mathcal{G}$.*

Proposition 5.34$^{\oplus}$. *Lors de l'addition d'un argument z dans le cadre de la sémantique préférée, basique ou stable, si le changement est* c-expansif*, alors il existe une extension $\mathcal{E}'$ de $\mathcal{G}'$ telle que $z \in \mathcal{E}'$.*

5.4.1.2 Résultats ne concernant que certaines sémantiques

Proposition 5.35$^{\oplus}$. *Lors de l'addition d'un argument z dans le cadre de la sémantique préférée, si z n'attaque aucun argument dans $\mathcal{G}$, alors, pour toute extension $\mathcal{E}'_i$ de $\mathcal{G}'$,*

1. *si $z \notin \mathcal{E}'_i$ alors $\mathcal{E}'_i$ est une extension préférée de $\mathcal{G}$.*
2. *si $z \in \mathcal{E}'_i$ alors $\mathcal{E}'_i \setminus \{z\}$ est une extension préférée de $\mathcal{G}$.*

De plus, $\mathcal{G}'$ et $\mathcal{G}$ ont le même nombre d'extensions ($|\mathbf{E}'| = |\mathbf{E}|$), donc le changement est constant*.*

La proposition 5.35$^{\oplus}$ complète les résultats donnés par la proposition 5.26 page 91.

Proposition 5.36$^{\oplus}$. *Lors de l'addition d'un argument dans le cadre de la sémantique stable, un changement ne peut pas être* c-limitatif*.*

Proposition 5.37$^{\oplus}$. *Lors de l'addition d'un argument z dans le cadre de la sémantique préférée ou basique, si le changement est* c-limitatif*, alors*

1. *z n'appartient à aucune extension de $\mathcal{G}'$ et*
2. *z attaque au moins un élément de $\mathcal{G}'$.*

Proposition 5.38$^{\oplus}$. *Lors de l'addition d'un argument z dans le cadre de la sémantique préférée ou basique, si z n'attaque aucun argument dans $\mathcal{G}'$ et appartient à chaque extension de $\mathcal{G}'$, alors le changement est* c-expansif*.*

Notons que la traduction de la proposition 5.39 page 93 grâce à la méthode de la dualité n'est pas intéressante car la proposition résultante produit une condition contradictoire ($x \notin \mathcal{E}$ et z est le seul attaquant de x, z étant l'argument ajouté).

De la même façon, la proposition 5.40 page 94 n'est pas traduite car la condition résultante est contradictoire ($x \in \mathcal{E}$ et il existe un attaquant y de x tel que tout chemin de longueur impaire de $\mathcal{F}'(\varnothing)$ à y contient z à une position paire et il n'existe aucun chemin de longueur impaire de z à x).

Soit $x \in \mathcal{G} \cap \mathcal{G}'$, la proposition suivante donne une caractérisation de l'*élimination d'acceptabilité* pour x. Néanmoins, bien que la condition proposée ne soit pas contradictoire, elle n'est pas facile à vérifier (notamment car elle concerne en partie le système après le changement).

Proposition 5.41$^{\oplus}$**.** *Lors de l'addition d'un argument z dans le cadre de la sémantique basique, si $x \in \mathcal{E}$ et x est attaqué dans $\mathcal{G}'$ et $\forall \mathcal{S}$ tel que $x \in \mathcal{F}'(\mathcal{S}), z \in \mathcal{S}$, alors $x \notin \mathcal{E}'$.*

5.4.2 Caractérisations dans le cas de la suppression d'argument

La démonstration des propositions de cette partie suit la méthodologie décrite dans la section 5.2.3 page 87, et utilise les lemmes donnés dans la section 5.1 page 82 ainsi que la notation 5.1 page 83.

5.4.2.1 Resultats pour la sémantique basique

Soit $x \in \mathcal{G} \cap \mathcal{G}'$, la proposition suivante (utilisée pour illustrer notre méthodologie dans la section 5.2.3 page 87) caractérise la *conservation de l'acceptabilité* de x.

Proposition 5.13$^{\ominus}$**.** *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si $x \in \mathcal{E}$ et z ne défend pas indirectement x, alors $x \in \mathcal{E}'$.*

Soit $x \in \mathcal{G} \cap \mathcal{G}'$, la première proposition caractérise la *conservation du rejet* pour x.

Proposition 5.14$^{\ominus}$**.** *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si $x \notin \mathcal{E}$ et z n'attaque pas indirectement x, alors $x \notin \mathcal{E}'$.*

Notons que la traduction de la proposition 5.15 page 90 grâce à la méthode de la dualité donnerait un résultat trivial dans le cadre de la sémantique basique ("$z \notin \mathcal{E}$ implique z est attaqué par $\mathcal{G}$").

Proposition 5.16.1$^{\ominus}$**.** *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si $\mathcal{E} \neq \varnothing$ et z est attaqué par $\mathcal{G}$, alors $\mathcal{E}' \neq \varnothing$.*

Proposition 5.16.2$^{\ominus}$**.** *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si $\mathcal{E} \neq \{z\} \cup \bigcup_{i \geq 1} \mathcal{F}^i(\{z\})$ et z n'est pas attaqué par $\mathcal{G}$, alors $\mathcal{E}' \neq \varnothing$.*

Corollaire 5.1. *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si le changement est c_{1nv-1v}, alors z n'est pas attaqué par $\mathcal{G}$ et $\mathcal{E} = \{z\} \cup \bigcup_{i \geq 1} \mathcal{F}^i(\{z\})$.*

Proposition 5.17$^{\ominus}$**.** *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si $\forall x \in \mathcal{G}$, si z attaque x alors (x est attaqué par $\mathcal{G} \setminus \{z\}$ et x n'est pas défendu indirectement par $\mathcal{N}_z$ dans $\mathcal{G} \setminus \{z\}$), alors $\mathcal{E}' \subseteq \mathcal{E}$.*

Proposition 5.18.1$^{\ominus}$**.** *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si*

- *$\forall x \in \mathcal{G}$, si z attaque x alors (x est attaqué par $\mathcal{G} \setminus \{z\}$ et x n'est pas défendu indirectement par $\mathcal{N}_z$ dans $\mathcal{G} \setminus \{z\}$) et*
- *$z \notin \bigcup_{i \geq 1} \mathcal{F}^i(\mathcal{N}_z)$,*

alors $\mathcal{E} = \mathcal{E}'$.

La proposition suivante est une traduction de la proposition 5.18.2 page 90 grâce à la méthode de la dualité, mais ce résultat n'est pas facile à exploiter puisque la condition concerne le système après le changement.

Proposition 5.18.2$^{\ominus}$**.** *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si z n'attaque pas $\mathcal{E}'$ et $\mathcal{E}'$ défend z, alors $\mathcal{E} = \mathcal{E}' \cup \{z\} \cup \bigcup_{i \geq 1} \mathcal{F}^i(\{z\})$.*

Proposition 5.18.3$^{\ominus}$**.** *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si*

- *$z \in \bigcup_{i \geq 1} \mathcal{F}^i(\mathcal{N}_z)$ et*
- *z n'attaque pas $\mathcal{G} \setminus \{z\}$,*

alors $\mathcal{E}' = \mathcal{E} \setminus \{z\}$.

La proposition suivante donne une condition nécessaire et suffisante pour un changement c_{1v-1nv}.

Proposition 5.19$^{\ominus}$**.** *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si $\mathcal{E}' \neq \varnothing$ et z attaque chaque argument non attaqué de $\mathcal{G} \setminus \{z\}$ et z est attaqué par $\mathcal{G} \setminus \{z\}$, alors $\mathcal{E} = \varnothing$. Le contraire est vrai aussi.*

Notons que l'application de la méthode de la dualité sur les propositions 5.20 page 91 et 5.21 page 91 donne des conditions impossibles (pour la proposition 5.20 : "z n'est pas attaqué par $\mathcal{G}$, z défend indirectement x et $x \notin \mathcal{E}$", et pour la proposition 5.21 : "$\mathcal{E}' \setminus \{x\}$ n'attaque pas z, z attaque x et $x \in \mathcal{E}$").

5.4.2.2 Resultats pour la sémantique préférée

Notons que la traduction de la proposition 5.22 page 91 grâce à la méthode de la dualité donnerait un résultat trivial dans le cadre de la sémantique préférée ("z non attaqué par $\mathcal{G}$ implique z appartient à chaque $\mathcal{E}_i$").

Proposition 5.23$^{\ominus}$. *Lors de la suppression d'un argument z dans le cadre de la sémantique préférée,*

1. *si z n'attaque pas $\mathcal{E}'_i$, alors $\mathcal{E}'_i$ est admissible dans $\mathcal{G}$;*
2. *si z n'attaque pas $\mathcal{E}'_i$ et $\mathcal{E}'_i$ défend z, alors $\mathcal{E}'_i \cup \{z\}$ est admissible dans $\mathcal{G}$.*

Le résultat précédent est relié aux propositions 5.32 à 5.35 page 93 qui sont des cas spécifiques de cette proposition lorsque nous nous restreignons à la sémantique préférée.

Proposition 5.24$^{\ominus}$. *Lors de la suppression d'un argument z dans le cadre de la sémantique préférée, si $\mathbf{E}' = \{\{\}\}$ et z n'est pas attaqué par $\mathcal{G}$ et il n'existe pas de cycle de taille paire dans $\mathcal{G}'$, alors $\mathbf{E} = \{\mathcal{E}\}$ et $z \in \mathcal{E}$ (donc, $\ominus$ est c_{1nv-1v}).*

Proposition 5.25$^{\ominus}$. *Lors de la suppression d'un argument z dans le cadre de la sémantique préférée, si z n'attaque aucun argument de $\mathcal{G}$ et $\mathbf{E}' = \{\{\}\}$, alors $\mathbf{E} = \{\{\}\}$; ou, de façon équivalente, si $\mathbf{E}' = \{\{\}\}$, le changement $\ominus$ par z est c_{1nv-1v} seulement si z attaque $\mathcal{G}'$.*

La proposition suivante complète la proposition 5.35 page 93.

Proposition 5.26$^{\ominus}$. *Lors de la suppression d'un argument z dans le cadre de la sémantique préférée, si z n'attaque aucun argument de $\mathcal{G}$ et $\mathbf{E}' \neq \{\{\}\}$, alors pour tout i :*

1. *si $\mathcal{E}'_i$ défend z, alors $\mathcal{E}'_i \cup \{z\}$ est une extension de $\mathcal{G}$;*
2. *si $\mathcal{E}'_i$ ne défend pas z, alors $\mathcal{E}'_i$ est une extension $\mathcal{G}$.*

De plus, $\mathcal{G}'$ et $\mathcal{G}$ ont le même nombre d'extensions ($|\mathbf{E}'| = |\mathbf{E}|$), donc le changement est constant.

Proposition 5.27$^{\ominus}$. *Dans le cadre de la sémantique préférée, si z n'attaque aucune extension de $\mathcal{G}'$, alors le changement $\ominus$ satisfait la* monotonie restrictive simple.

Proposition 5.28$^{\ominus}$. *Dans le cadre de la sémantique préférée, supposons que $\mathcal{G}'$ ne contienne aucun argument controversé. Si z n'attaque pas $\bigcap_{i\geq 1} \mathcal{E}'_i$, alors le changement $\ominus$ satisfait la* monotonie restrictive sceptique, *c'est-à-dire $\bigcap_{i\geq 1} \mathcal{E}'_i \subseteq \bigcap_{i\geq 1} \mathcal{E}_i$.*

Notons que la proposition suivante n'est pas facile à exploiter puisque la condition concerne le système après le changement.

Proposition 5.29$^{\ominus}$. *Lors de la suppression d'un argument z dans le cadre de la sémantique préférée, si $\mathbf{E}' \neq \{\{\}\}$ et il n'existe aucun cycle de longueur paire dans $\mathcal{G}$ et tout argument non attaqué de $\mathcal{G}'$ est attaqué dans $\mathcal{G}$ et z est attaqué dans $\mathcal{G}$, alors $\mathbf{E} = \{\{\}\}$ (et donc le changement $\ominus$ est c_{1v-1nv}).*

5.5 Synthèse

Tous les résultats de caractérisation donnés dans ce chapitre sont synthétisés dans les tables suivantes. Il est important de noter que certaines CS (conditions suffisantes) et CN (conditions nécessaires) obtenues par l'application de la dualité ne sont pas vraiment utiles car elles concernent le système de sortie ; il est alors très difficile de les traduire en termes de conditions sur le système d'entrée, que ce soit pour l'addition ou la suppression (ces CS et CN sont notées *CS* et *CN* dans les tables).

Pour chaque opération de changement ($\oplus$ et $\ominus$), deux tables sont données, la première donnant les CS et CN trouvées pour les propriétés définies essentiellement[2] dans le chapitre 4 page 59, et la seconde donnant quelques propositions supplémentaires.

Notons que pour certaines propriétés et/ou certaines sémantiques, aucune caractérisation n'a pu être trouvée.

5.5.1 Synthèse pour l'addition

Propriétés du changement $\oplus$	Sémantique basique	Sémantique préférée	Sémantique stable
e_{0-1nv}, e_{0-k}, r_{k-0}, r_{1nv-0}	Jamais (Prop. 5.1)	Jamais (Prop. 5.2)	
c_{1v-1nv}	CNS : Prop. 5.16.2	CS : Prop. 5.24 CN : Prop. 5.25	Jamais (Prop. 5.3)
r_{k-1nv}	Jamais (Prop. 5.1)		
r_{k-j}	Jamais (Prop. 5.1)	CN : $\exists$ un cycle de longueur paire dans $\mathcal{G}$ + Prop. 5.23	
e_{1nv-k}	Jamais (Prop. 5.1)	CN : $\exists$ un cycle de longueur paire dans $\mathcal{G}'$ + Prop. 5.25, Prop. 5.26	
e_{1v-k}	Jamais (Prop. 5.1)	CN : $\exists$ un cycle de longueur paire dans $\mathcal{G}'$ + Prop. 5.25, Prop. 5.26	Jamais (Prop. 5.3)
e_{j-k}	Jamais (Prop. 5.1)	CN : $\exists$ un cycle de longueur paire dans $\mathcal{G}'$ + Prop. 5.25, Prop. 5.26	
c_{1nv-1v}	CNS : Prop. 5.19	CS : Prop. 5.29 CN : Prop. 2.1.5, 2.2.2	Jamais (Prop. 5.3)
r_{k-1v}	Jamais (Prop. 5.1)	CN : Prop. 2.1.5, 2.2.2	Jamais (Prop. 5.3)

2. La *monotonie expansive faible* et la *monotonie restrictive faible*, sans être définies formellement, sont caractérisées dans ce même chapitre grâce aux propositions 5.32 page 93 et 5.32.3$^{\oplus}$ page 95, respectivement.

Propriétés du changement $\oplus$	Sémantique basique	Sémantique préférée	Sémantique stable
c-expansif	CS : Prop. 5.18.2 + Prop. 5.18.3 *CN* : Prop. 5.34$^{\oplus}$ *CS* : Prop. 5.38$^{\oplus}$	CS : Prop. 5.26.1 *CN* : Prop. 5.34$^{\oplus}$ *CS* : Prop. 5.38$^{\oplus}$	*CN* : Prop. 5.34$^{\oplus}$
c-conservatif	CS : Prop. 5.16.1 CS : Prop. 5.18.1	CS : Prop. 5.25 CS : Prop. 5.26.2	
c-limitatif	CN : Prop. 5.17 *CN* : Prop. 5.37$^{\oplus}$	CN : Prop. 5.23 *CN* : Prop. 5.37$^{\oplus}$	Jamais (Prop. 5.36$^{\oplus}$)
c-modifiant	CN : Prop. 5.17	CN : Prop. 5.23	
Monotonie expansive simple	CS : $\mathcal{E} = \{\}$ CS : Prop. 5.17	CS : Prop. 5.27	
Monotonie restrictive simple	*CNS* : Prop. 5.31$^{\oplus}$	*CNS* : Prop. 5.31$^{\oplus}$	*CNS* : Prop. 5.31$^{\oplus}$
Monotonie expansive faible			
Monotonie restrictive faible	*CNS* : Prop. 5.32.3$^{\oplus}$	*CNS* : Prop. 5.32.1$^{\oplus}$	*CNS* : Prop. 5.32.2$^{\oplus}$
Priorité à la nouveauté	CS : Prop. 5.15 CS : Prop. 5.18	CS : Prop. 5.22 CS : Prop. 5.24	
Monotonie expansive sceptique	*cf. monotonie expansive simple*	CS : Prop. 5.28	
Monotonie restrictive sceptique	*cf. monotonie restrictive simple*		
Monotonie expansive crédule	*cf. monotonie expansive simple*	*cf. monotonie expansive simple*	*cf. monotonie expansive simple*
Monotonie restrictive crédule	*cf. monotonie restrictive simple*	*cf. monotonie restrictive simple*	*cf. monotonie restrictive simple*
Conservation du rejet	CS : Prop. 5.13		
Conservation de l'acceptabilité	CS : Prop. 5.14 et *cf. monotonie expansive simple*	*cf. monotonie expansive simple*	*cf. monotonie expansive simple*
Conservation de l'acceptabilité sceptique	CS : Prop. 5.14	*cf. monotonie expansive sceptique*	*cf. monotonie expansive sceptique*
Conservation de l'acceptabilité uniquement crédule	CS : Prop. 5.14	*cf. monotonie expansive simple*	*cf. monotonie expansive simple*

Propriétés du changement $\oplus$	Sémantique basique	Sémantique préférée	Sémantique stable
Instauration d'acceptabilité	CS : Conseq. 5.1 CS : Prop. 5.20		
Élimination d'acceptabilité	CS : Prop. 5.21 *CS* : Prop. 5.41$^{\oplus}$		

Prop. numéro	Pour $\oplus$, correspond à	Sémantique
Prop. 5.33$^{\oplus}$	*CS* pour la "conservation d'une extension finale" (*i.e.* dans le cas où une extension après le changement était déjà une extension avant le changement)	P, S, B
Prop. 5.35$^{\oplus}$	*CS* pour la conservation (éventuellement faible) d'une extension finale (*i.e.* dans le cas où une extension après le changement ne contenant pas z était déjà une extension avant le changement)	P

5.5.2 Synthèse pour la suppression

Propriétés du changement $\ominus$	Sémantique basique	Sémantique préférée	Sémantique stable
e_{0-1nv}, e_{0-k}, r_{k-0}, r_{1nv-0}	Jamais (Prop. 5.1)	Jamais (Prop. 5.2)	
c_{1v-1nv}	CNS : Prop. 5.19$^{\ominus}$	*CS* : Prop. 5.29$^{\ominus}$ CN : Prop. 2.1.5 + Prop. 2.2.2	Jamais (Prop. 5.3)
r_{k-1nv}	Jamais (Prop. 5.1)	*CN* : $\exists$ un cycle de longueur paire dans $\mathcal{G}$ + Prop. 5.25$^{\ominus}$, Prop. 5.26$^{\ominus}$	
r_{k-j}	Jamais (Prop. 5.1)	*CN* : $\exists$ un cycle de longueur paire dans $\mathcal{G}$ + Prop. 5.25$^{\ominus}$, Prop. 5.26$^{\ominus}$	
e_{1nv-k}	Jamais (Prop. 5.1)		
e_{1v-k}	Jamais (Prop. 5.1)	CN : Prop. 2.1.5 + Prop. 2.2.2	Jamais (Prop. 5.3)
e_{j-k}	Jamais (Prop. 5.1)	*CN* : $\exists$ un cycle de longueur paire dans $\mathcal{G}'$ + Prop. 5.23$^{\ominus}$	
c_{1nv-1v}	CN : Prop. 5.16.1$^{\ominus}$ CN : Prop. 5.16.2$^{\ominus}$ CN : Corollaire 5.1	*CS* : Prop. 5.24$^{\ominus}$ *CN* : Prop. 5.25$^{\ominus}$	Jamais (Prop. 5.3)
r_{k-1v}	Jamais (Prop. 5.1)	*CN* : $\exists$ un cycle de longueur paire dans $\mathcal{G}$ + Prop. 5.25$^{\ominus}$, Prop. 5.26$^{\ominus}$	Jamais (Prop. 5.3)

Propriétés du changement ⊜	Sémantique basique	Sémantique préférée	Sémantique stable
c-expansif	CN : Prop. 5.37	CN : Prop. 5.37 *CN* : Prop. 5.23⊜	Jamais (Prop. 5.36)
c-conservatif	CS : Prop. 5.18.1⊜	*CS* : Prop. 5.25⊜ *CS* : Prop. 5.26.2⊜	
c-limitatif	CN : Prop. 5.34 CS : Prop. 5.38 *CS* : Prop. 5.18.2⊜ CS : Prop. 5.18.3⊜	CN : Prop. 5.34 CS : Prop. 5.38 *CS* : Prop. 5.26.1⊜	CN : Prop. 5.34
c-modifiant	CN : Prop. 5.17⊜	*CN* : Prop. 5.23⊜	
Monotonie expansive simple	CNS : Prop. 5.31	CNS : Prop. 5.31	CNS : Prop. 5.31
Monotonie restrictive simple	CS : $\mathcal{E}' = \{\}$ CS : Prop. 5.17⊜	*CS* : Prop. 5.27⊜	
Monotonie expansive faible	CNS : Prop. 5.32.3	CNS : Prop. 5.32.1	CNS : Prop. 5.32.2
Monotonie restrictive faible			
Priorité à la nouveauté	Jamais (def. *priorité à la nouveauté*)	Jamais (def. *priorité à la nouveauté*)	Jamais (def. *priorité à la nouveauté*)
Monotonie expansive sceptique	*cf. monotonie expansive simple*		
Monotonie restrictive sceptique	*cf. monotonie restrictive simple*	*CS* : Prop. 5.28⊜	
Monotonie expansive crédule	*cf. monotonie expansive simple*	*cf. monotonie expansive simple*	*cf. monotonie expansive simple*
Monotonie restrictive crédule	*cf. monotonie restrictive simple*	*cf. monotonie restrictive simple*	*cf. monotonie restrictive simple*
Conservation du rejet	CS : Prop. 5.14⊜		
Conservation de l'acceptabilité	CS : Prop. 5.13⊜ et *cf. monotonie expansive simple*	*cf. monotonie expansive simple*	
Conservation de l'acceptabilité sceptique	*cf. monotonie expansive sceptique*	*cf. monotonie expansive sceptique*	*cf. monotonie expansive sceptique*

Propriétés du changement $\ominus$	Sémantique basique	Sémantique préférée	Sémantique stable
Conservation de l'acceptabilité uniquement crédule	*cf. monotonie expansive simple*	*cf. monotonie expansive simple*	*cf. monotonie expansive simple*
Instauration d'acceptabilité	CS : Prop. 5.39		
Élimination d'acceptabilité	CS : Prop. 5.40 CS : Prop. 5.41		

Prop. numéro	Pour $\ominus$, correspond à	Sémantique
Prop. 5.33	CS pour la "conservation d'une extension initiale" (*i.e.* dans le cas où une extension avant le changement est toujours une extension après le changement)	P, S, B
Prop. 5.35	CS pour la conservation (éventuellement faible) d'une extension initiale (*i.e.* dans le cas où une extension avant le changement avec éventuellement l'addition de z est toujours une extension après le changement)	P

5.6 Discussion

Synthèse

Dans ce chapitre, nous avons présenté une étude détaillée de deux opérations de changement en argumentation, à savoir l'addition et la suppression d'un argument (et de ses interactions). Plus précisément, nous avons donné, pour chacune de ces opérations, des ensembles de conditions (nécessaires et/ou suffisantes) permettant de s'assurer qu'une propriété du changement sera effectivement satisfaite après un changement.

Toutefois, face au nombre important de résultats de caractérisation, le lecteur pourrait se demander lesquels sont les plus utiles. Pour répondre à cette question, trois critères peuvent être pris en compte :

- Le type de changement concerné : certains changements peuvent être considérés comme utiles selon le rôle de l'utilisateur, *e.g.* le modérateur d'un débat peut souhaiter préciser ou élargir le dialogue suivant le temps restant, alors qu'un orateur peut avoir une ou plusieurs stratégies de dialogue particulières et vouloir se concentrer sur des arguments en particulier (le lecteur pourra se reporter au chapitre 3 page 29 et à l'annexe D.1 page 221 pour de plus amples illustrations). Examinons certaines de ces propriétés :

- Un changement “décisif” (*e.g.*, c_{1v-1nv}) est utile pour faire diminuer l’ignorance puisqu’après ce changement, il ne reste qu’une et une seule extension. Cela peut être utilisé par un modérateur pour conclure le débat.
- Un changement “expansif” (*e.g.*, *c-expansif*) augmente le nombre d’arguments acceptés tout en conservant ceux qui l’étaient déjà. Cela peut également être utilisé par un modérateur ou par un orateur dans le but de convaincre un auditoire plus large à propos du point de vue actuel du débat.
- Un changement “conservatif” (*e.g.*, *c-conservatif*) peut représenter une attitude neutre que peut adopter un modérateur ou un orateur dans le cas où il ne veut pas apporter de nouvelles informations, mais veut tout de même participer au débat. Un bel exemple de *parler pour ne rien dire.*
- Les différentes “monotonies” et les propriétés du changement basées sur le statut d’un argument (*e.g.*, *instauration d’acceptabilité*) permettent de se concentrer sur certains arguments particuliers et peuvent être utilisées stratégiquement.
- Les changements “ouvrant” (*e.g.*, e_{j-k}) et “destructif” (*e.g.*, r_{k-1v}) accroissent l’ignorance soit par l’augmentation des points de vue possibles, soit par la destruction de tout point de vue cohérent. Cela peut être utilisé en désespoir de cause par un orateur voulant interdire toute prise de décision.
- Un changement “modifiant” (*e.g.*, *c-modifiant*) permet de complètement changer le ou les points de vue actuels. Cela peut également être utilisé pour changer le cours du débat.

• La nature de la caractérisation obtenue en termes de temps de calcul requis pour vérifier sa condition (*e.g.*, vérifier si un argument n’attaque aucun autre argument est plus facile que de vérifier s’il appartient à une extension).

• La nature de la caractérisation obtenue en termes de “typicalité” : la condition est-elle souvent réalisée dans un système d’argumentation ordinaire, ou est-elle peu rencontrée ?

Au-delà du point de vue théorique, ces résultats de caractérisation peuvent être utilisés dans une dimension plus concrète. Nous verrons dans le chapitre 8 page 155 comment il est possible de les exploiter dans un outil permettant de calculer une opération de changement réalisant un but.

Il est néanmoins important de noter que certaines caractérisations obtenues grâce à la méthodologie de la dualité se sont révélées problématiques dans leur utilisation. En effet, l’application naïve de cette méthodologie peut amener les conditions à porter sur le système d’argumentation résultant du changement. Dans ce cas, il est difficile de se servir directement de telles caractérisations et nous avons donc dû fournir un travail supplémentaire pour s’assurer que ces résultats aient un sens ; nous avons notamment cherché à donner des conditions équivalentes sur le système d'argumentation avant le résultat, par exemple au moyen des lemmes 5.1 page 84 et 5.2 page 84.

Pour résumer l'intérêt de ce chapitre, revenons à l'exemple donné au début de celui-ci. Pour Monsieur Pink, ajouter un nouvel argument attaquant un argument spécifique de Monsieur White sans menacer ses propres arguments acceptés correspond à la proposition 5.14 page 90. La proposition 5.13⊜ page 96, d'un autre côté, lui permet de s'assurer que la suppression de l'argument de son adversaire donne le même résultat si cet argument ne porte assistance à aucun de ses propres arguments acceptés. Ainsi, au lieu d'utiliser la proposition 5.14 page 90, Monsieur Pink peut bénéficier de la proposition 5.13⊜ page 96, obtenue grâce à notre méthodologie (le lecteur peut se rapporter aux lignes concernant la "*conservation de l'acceptabilité*" dans les tables de la section 5.5 page 99).

Travaux connexes

Notons l'approche similaire de Boella *et al.* (2009a,b,c) qui propose, à la manière de Cayrol *et al.* (2010c), d'étudier le changement et les modifications qui en découlent. L'objectif du travail est ici d'étudier comment la conclusion d'un système d'argument peut rester inchangée (un équivalent de notre propriété du changement *c-conservatif*) malgré la modification de ce système. Ces travaux considèrent uniquement le cas où il n'existe qu'une seule extension (par exemple, dans le cadre de la sémantique basique) et ils fournissent des *principes* ("principles"), ressemblant à nos caractérisations, qui décrivent des conditions sur le système d'argumentation et le changement le modifiant. Ainsi, si les conditions sont satisfaites, alors nous sommes assurés que l'unique extension ne sera pas modifiée.

Notons que les auteurs considèrent l'addition d'argument et d'attaque dans Boella *et al.* (2009b,c) et la suppression d'argument et d'attaque dans Boella *et al.* (2009a).

Dans ce chapitre, nous avons étudié *comment* une modification particulière d'un système d'argumentation peut être provoquée. Une question découlant naturellement de cette étude est "pourquoi" une telle modification peut être souhaitée. Ainsi, en utilisant les résultats présentés dans cette partie, nous introduisons dans la prochaine partie un cadre permettant à un agent de choisir l'opération adéquate pour satisfaire un but dans un système d'argumentation (chapitre 6 page 109).

Troisième partie

Spécialisation et axiomatisation du changement

CHAPITRE 6

Un cadre contraint pour la prise en compte des acteurs du changement

Le plus heureux des hommes est celui qui désire le moins le changement de son état.

MARQUISE DU CHÂTELET,
Discours sur le bonheur.

Sommaire

Grâce aux chapitres 4 page 59 et 5 page 81, nous savons respectivement quelles sont les modifications qu'un système d'argumentation peut subir, et les conditions permettant leur apparition. Pour compléter ces études, nous souhaitons maintenant savoir ce qui est susceptible de provoquer ces modifications, ce qui les motive. Intuitivement, il paraît probable que l'on souhaite modifier un système d'argumentation pour que ce dernier nous convienne. Par ailleurs, il paraît tout aussi probable qu'il soit impossible de le modifier comme il nous plaît ; il est impératif de posséder les bons arguments, par exemple. Ainsi, dans ce chapitre, nous introduisons un nouveau cadre formel permettant de prendre en compte ces intuitions.

S'inscrivant dans le domaine de la dynamique en argumentation, ce cadre est proche de la planification [1] classique. On désire modéliser le raisonnement d'un agent qui possède ses propres connaissances, un objectif, et qui a la possibilité d'agir sur un système cible. Ici, les connaissances de l'agent sont représentées sous la forme d'un système d'argumentation. La cible sur laquelle il peut agir est aussi un système d'argumentation. Par exemple, lors d'un débat, les arguments échangés publiquement ainsi que leurs interactions peuvent constituer le système cible ; nous avons pu le voir dans l'exemple de l'audience de tribunal (section 3.1 page 30). Notons à ce propos que le protocole présenté dans la section 3.1.2 page 36 met en évidence deux types de changement :

- changement opéré par un agent sur un système cible autre que le sien,
- changement opéré par un agent sur son propre système.

Ici, nous nous concentrons uniquement sur l'étude du premier type de changement.

L'originalité de notre formalisation du changement réside dans l'utilisation de deux systèmes d'argumentation : un pour l'agent et un pour la cible. Cela introduit des limitations aux actions possibles de l'agent (puisqu'il ne peut pas, par exemple, ajouter des arguments ou des attaques qu'il ne connaît pas) et rend le problème de la réalisation de ses objectifs moins trivial.

Par ailleurs, une extension naturelle de la dynamique d'un système d'argumentation, et donc de notre cadre, est la possibilité de choisir un changement optimal au regard d'un critère particulier. Aussi, dans ce chapitre, nous présentons une approche destinée à répondre à la question : "*Entre deux changements, lequel est le meilleur ?*" Nous nous proposons donc d'examiner comment comparer des changements.

Dans la section 6.1, nous présentons notre cadre théorique qui s'appuie sur les notions qui ont été présentées et étendues dans les chapitres 2 page 11, 4 page 59 et 5 page 81 tout en les modifiant. Puis, nous abordons le problème de la comparaison et du choix d'un changement dans la section 6.2 page 121. Enfin, la section 6.3 page 131 conclut cette partie.

Notons que ce chapitre détaille le cadre présenté dans Bisquert *et al.* (2013c,d).

6.1 Nouvelles définitions

Cette section nous permet d'introduire formellement certaines notions utiles à notre travail, notions que nous tentons d'exemplifier au fur et à mesure. À ce titre, nous reprenons et modifions des notions que nous avons eu l'occasion de présenter dans le chapitre 2 page 11 de façon à les rendre cohérentes avec l'utilisation de deux systèmes d'argumentation.

1. La planification est la discipline ayant pour objectif la création automatisée de plans (des listes d'actions à entreprendre) permettant la résolution d'un problème spécifique.

6.1.1 Univers

Avant d'aller plus loin, il convient de définir dans quelle mesure un agent a accès aux arguments. Pour cela, nous avons besoin d'une notion pouvant représenter les arguments qu'un agent connaît et parmi lesquels il peut choisir ceux qu'il va avancer dans un dialogue, par exemple. Ceci nous permettra de nous situer dans un cadre potentiellement concret tout en restant dans le domaine de l'argumentation abstraite.

Ainsi, dans ce travail, nous introduisons la notion d'**univers de référence** représentant un ensemble fini ou infini d'arguments manipulables dans un domaine (ou contexte) donné. [2] Notons que nous ne précisons pas, à dessein, l'origine de ces arguments, ce qui nous permettra d'être plus flexible quant à leur obtention.

Néanmoins, à titre d'exemple, ce domaine peut être celui d'une base de connaissances ; l'univers de référence est alors l'ensemble de tous les arguments constructibles à partir de cette base. De la même façon, le domaine peut également être un texte argumentatif, tel un essai ; l'univers de référence est alors constitué des différents arguments (sous forme de phrases, par exemple) qui ont été extraits de ce texte. Nous pouvons enfin, comme dans l'exemple qui suit, considérer que l'univers de référence est fourni explicitement.

Exemple 6.1. *Revenons à l'audience de tribunal où un procureur et un avocat s'affrontent pour déterminer le sort du prévenu M. X. Chacun des deux orateurs a un but, à savoir, pour le procureur, de faire condamner M. X (en faisant accepter l'argument x_1), et pour l'avocat, de le faire innocenter (en faisant accepter l'argument x_0). Les différents arguments pouvant être mis en jeu sont présentés dans la table* *3.1 page 32* *; ils constituent l'*univers de référence *dans ce contexte particulier.*

La structure interne des arguments n'étant pas ce qui nous intéresse ici, nous procédons à **l'abstraction de l'univers de référence** en usant du cadre formel de Dung (1995). Ceci nous permet de nous concentrer sur les interactions entre arguments (représentés par des symboles x_i) et la structure qui en découle.

Définition 6.1 (Abstraction de l'univers de référence). *Une* abstraction d'univers de référence $\mathbf{U}$ *est une paire* $(\mathcal{A}_\mathbf{U}, \mathcal{R}_\mathbf{U})$*, où* $\mathcal{A}_\mathbf{U}$ *est l'ensemble non vide de tous les arguments de cet univers et* $\mathcal{R}_\mathbf{U} \subseteq \mathcal{A}_\mathbf{U} \times \mathcal{A}_\mathbf{U}$ *est une relation binaire sur* $\mathcal{A}_\mathbf{U}$*.* [3]

Cette définition nous permet aisément de représenter graphiquement un univers de référence.

Exemple 6.1 (cont.). *En abstrayant l'univers de référence représenté par la table* *3.1 page 32**, nous obtenons le graphe illustré par la figure* *6.1 page suivante**.*

2. Il peut donc exister plusieurs univers.
3. Dans la suite de ce document, $\mathbf{U}$ dénotera autant un univers de référence que son abstraction.

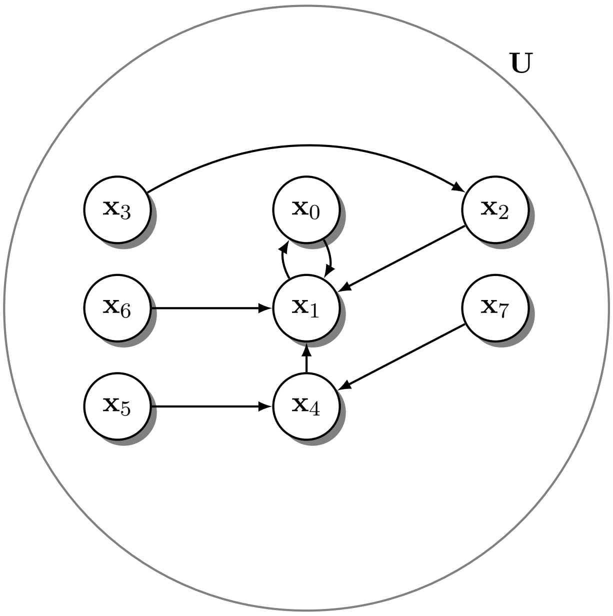


Figure 6.1 – *Illustration de la notion d'abstraction de l'univers de référence.*

De la notion d'abstraction de l'univers de référence découle directement une nouvelle définition du système d'argumentation. Celle-ci se distingue de la définition de Dung (1995) par le fait que le système d'argumentation est relatif à un univers de référence.

Définition 6.2 (**Système d'argumentation**). *Étant donné un univers de référence* $\mathbf{U}$*, un* système d'argumentation *issu de* $\mathbf{U}$ *est une paire* $(\mathcal{A}, \mathcal{R})$*, où* $\mathcal{A} \subseteq \mathcal{A}_{\mathbf{U}}$ *est un ensemble non vide fini d'arguments et* $\mathcal{R} \subseteq \mathcal{R}_{\mathbf{U}} \ \cap \ \mathcal{A} \times \mathcal{A}$ *est une relation binaire sur* $\mathcal{A}$*.* $(\mathcal{A}, \mathcal{R})$ *est représenté par un* graphe d'argumentation $\mathcal{G}$ *dont les sommets et les arcs correspondent respectivement à* $\mathcal{A}$ *et* $\mathcal{R}$*. L'ensemble des graphes d'argumentation issu de* $\mathbf{U}$ *est noté* $\mathbf{G}_{\mathbf{U}}$*.*[4]

Dans la suite de ce travail, un système d'argumentation sera le plus souvent associé à un agent. Soit k un agent, nous désignerons par $\mathbf{U}_k = (\mathcal{A}_k, \mathcal{R}_k)$ le système d'argumentation de l'agent k issu de l'univers de référence $\mathbf{U}$, et représentant la partie de l'univers de référence connue par l'agent k. Nous noterons $\mathcal{G}_k$ le graphe représentant ce système. Il est à noter que lorsque plusieurs agents interagissent, nous supposons que les systèmes d'argumentation leur étant associés sont issus du même univers.

Exemple 6.1 (cont.). *Pour déterminer la culpabilité du prévenu, deux agents, l'avocat de la défense et le procureur s'affrontent oralement. Ces derniers n'étant pas omniscients, ils ne connaissent pas tous les arguments utilisables (et qui plus est, ils ne partagent pas nécessairement les mêmes arguments). Ainsi, à chacun de*

4. Notons que dans le cas où un seul univers de référence est utilisé, nous parlerons de "système d'argumentation" en lieu et place de "système d'argumentation issu de $\mathbf{U}$".

ces agents correspond un système d'argumentation, tel que le montre la figure 6.2 *page suivante* *où apparaissent :*

- **U** *l'univers de référence, dans lequel se trouvent les arguments* x_0 *à* x_7.
- $\mathbf{U}_{proc}$ *le système d'argumentation du procureur, contenant les arguments* x_0, x_1, x_3, x_4, x_5 *et* x_6.
- $\mathbf{U}_{avoc}$ *le système d'argumentation de l'avocat, contenant les arguments* x_0, x_1, x_2, x_4 *et* x_7.

Notons qu'il est possible que certains arguments et interactions soient absents des systèmes d'argumentation de tous les agents. Ceci est représenté graphiquement

- *pour les arguments* : par des sommets du graphe représentant l'univers n'étant contenus dans aucun système d'argumentation,
- *pour les interactions* : par des arcs de l'univers n'apparaissant entièrement dans aucun système d'argumentation (telle l'attaque de l'argument x_3 vers l'argument x_2, qui n'est contenue entièrement ni dans $\mathbf{U}_{proc}$, ni dans $\mathbf{U}_{avoc}$ sur la figure 6.2 page suivante).

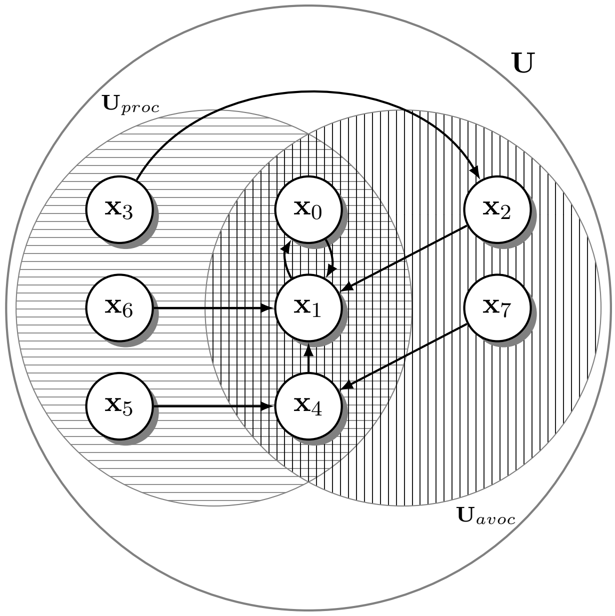
Dans le cas contraire, l'univers de référence se définit par l'union des arguments, et l'union des interactions les concernant, connus par les agents. Dans notre exemple, ceci aurait été le cas si l'argument x_3 n'attaquait pas l'argument x_2.

6.1.2 Opérations de changement

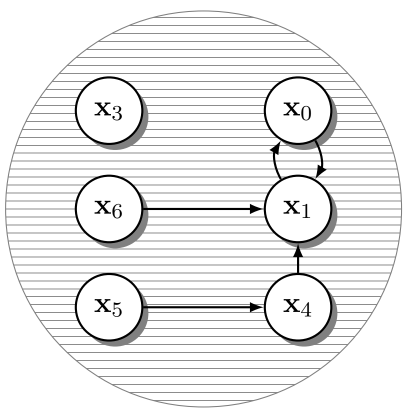
L'étude du changement dans un système d'argumentation commence par la définition d'une opération de changement. Pour ce faire, nous nous inspirons librement du travail de Cayrol *et al.* (2010c). Nous considérons des opérations de changement *élémentaires* (voir section 2.2 page 16), c'est-à-dire n'agissant que sur un argument ou une attaque appartenant à un univers particulier.

Les travaux de Cayrol *et al.* (2010c), que nous avons étendus dans les chapitres 4 page 59 et 5 page 81, seront utilisés dans un contexte particulier, où les opérations seront contraintes : un agent n'a pas forcément accès à toutes les opérations possibles dans un contexte particulier, ce qui correspond à une restriction des opérations permises à cet agent. La figure 6.3 page 115 illustre cet ajout par la présence du système d'argumentation de l'agent entreprenant l'opération ($\mathbf{U}_{agent}$). Par ailleurs, dans notre travail, une opération effectuée par un agent ne l'est pas forcément sur son propre système d'argumentation ; elle sera généralement destinée à modifier le système d'un autre agent, et plus généralement un système cible.

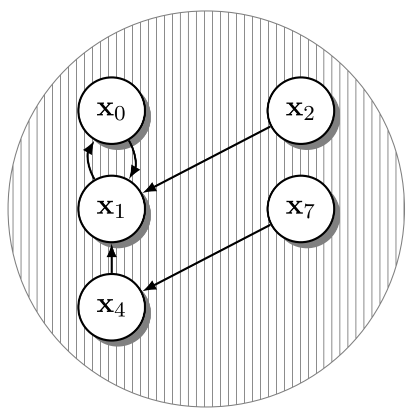
Ainsi, nous raffinons la notion d'opération élémentaire au sens de Cayrol *et al.* (2010c) en quatre points : en premier lieu nous nous penchons de manière abstraite sur cette notion en redéfinissant sa *syntaxe* ; puis nous la ramenons à un agent pour déterminer si une opération est *autorisée* ou non pour celui-ci ; nous la restreignons ensuite en fonction du contexte, ce qui permet de déterminer si une opération peut être *exécutée* par un agent sur un système d'argumentation cible ; enfin, nous étudions l'*impact* d'une opération sur un système d'argumentation.



(a) *Représentation graphique des arguments mis en jeu lors de l'audience.* **U** *symbolise l'univers de référence,* $\mathbf{U}_{proc}$ *le système d'argumentation du procureur et* $\mathbf{U}_{avoc}$ *le système d'argumentation de l'avocat.*



(b) *Système d'argumentation du procureur.*



(c) *Système d'argumentation de l'avocat.*

Figure 6.2 – *Illustration de la notion de système d'argumentation issu d'un univers de référence.*

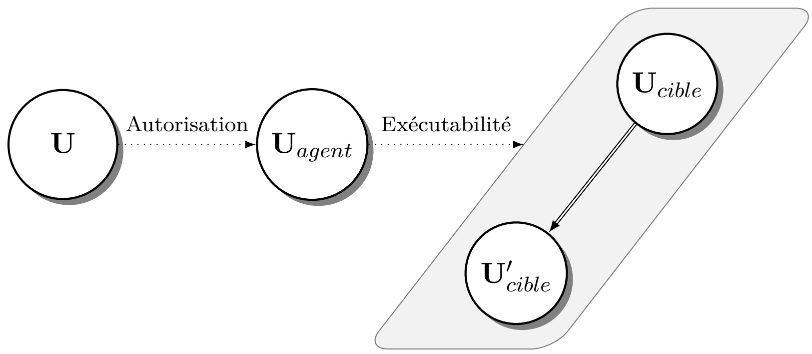


Figure 6.3 – *Illustration des contraintes sur les opérations. La zone grisée représente les travaux sur lesquels nous nous basons, c'est-à-dire la modification d'un système d'argumentation (système cible : $\mathbf{U}_{cible}$) menant à un nouveau système (système résultat : $\mathbf{U}'_{cible}$) par l'intermédiaire d'une opération de changement. Nous ajoutons à cela une notion de restriction sur les opérations de changement disponibles pour modifier $\mathbf{U}_{cible}$ par le fait qu'elles ne doivent se baser que sur des arguments et attaques existant dans le système d'argumentation de l'agent les effectuant (ce qui est représenté par le lien entre l'univers $\mathbf{U}$ et $\mathbf{U}_{agent}$) ; elles doivent de plus correspondre aux arguments et attaques de $\mathbf{U}_{cible}$ (ce qui est représenté par le lien entre le système de l'agent entreprenant l'opération ($\mathbf{U}_{agent}$) et l'ensemble des opérations exécutables sur le système cible ($\mathbf{U}_{cible}$).*

Définition 6.3 (**Opération élémentaire**). *Soit* $\mathbf{U} = (\mathcal{A}_\mathbf{U}, \mathcal{R}_\mathbf{U})$ *un univers de référence. Un triplet* $o = (op, arg, att)$ *est appelé* opération élémentaire *si et seulement si* $o.arg \subseteq \mathcal{A}_\mathbf{U}$, $o.att \subseteq \mathcal{R}_\mathbf{U}$ *et*

- *soit* $o.op = \bigoplus$ *avec* $|o.arg| = 1$ *et* $\forall(x, y) \in o.att,\ (x \neq y)$ *et* $(x \in o.arg$ *ou* $y \in o.arg)$,
- *soit* $o.op = \bigominus$ *avec* $|o.arg| = 1$ *et* $o.att = \varnothing$,
- *soit* $o.op \in \{\oplus, \ominus\}$ *avec* $o.arg = \varnothing$ *et* $|o.att| = 1$.

L'ensemble des opérations élémentaires issues de $\mathbf{U}$ *est noté* $\mathcal{O}_\mathbf{U}$.[5]

Cette définition, bien qu'elle s'en rapproche, compte quelques différences de forme par rapport à la définition 2.6 page 16. Ainsi, outre la définition sous forme de triplet, nous laissons ici entrevoir la définition possible d'opération en général ; en effet, *o.arg* et *o.att* sont des ensembles et peuvent donc contenir plusieurs arguments ou attaques ; relâcher les contraintes $|o.arg| = 1$ et $|o.att| = 1$ nous permettrait de considérer des opérations concernant plusieurs arguments et attaques à la fois. Notons néanmoins que ne considérer que les opérations élémentaires ne fait pas perdre en généralité puisqu'une opération quelconque peut se ramener à une séquence d'opérations élémentaires.

Mais la différence principale tient du fait qu'ici, lors de la suppression d'un argument, nous ne précisons pas l'ensemble des attaques à supprimer ($o.att = \varnothing$)

5. Par souci de simplicité, on utilisera de manière équivalente les notations $\mathcal{O}_\mathbf{U}$ et $\mathcal{O}$ pour désigner l'ensemble des opérations élémentaires issues de $\mathbf{U}$ lorsqu'il n'y a pas d'ambiguïté.

puisqu'il peut être directement déterminé en considérant toutes les attaques qui concernent l'argument à supprimer ($o.arg$) ; une opération élémentaire au sens de la définition 2.6 page 16 impose quant à elle que l'ensemble des attaques à supprimer ($\mathcal{R}_z$) soit renseigné. [6]

Exemple 6.2. *À partir de l'univers de référence représenté par la figure 6.1 page 112, plusieurs opérations élémentaires sont syntaxiquement correctes. En voici une liste non exhaustive :*

- $(\bigoplus, \{x_1\}, \varnothing)$,
- $(\bigominus, \{x_4\}, \varnothing)$,
- $(\bigoplus, \{x_2\}, \{(x_2, x_1)\})$,
- $(\bigominus, \{x_7\}, \varnothing)$.
- $(\ominus, \varnothing, \{(x_6, x_1)\})$,
- $(\oplus, \varnothing, \{(x_5, x_4)\})$,
- $(\oplus, \varnothing, \{(x_3, x_2)\})$,
- $(\ominus, \varnothing, \{(x_7, x_4)\})$,

Nous considérons maintenant les notions d'*opération autorisée* et d'*opération interdite* pour un agent : une opération élémentaire est dite autorisée si elle porte sur un argument et/ou des attaques connus de l'agent. À l'inverse, une opération élémentaire est dite interdite si elle porte sur un argument et/ou des attaques inconnus de l'agent. Cette nouvelle notion est illustrée sur la figure 6.3 page précédente par le lien entre l'univers $\mathbf{U}$ et le nœud représentant le système d'argumentation de l'agent réalisant l'opération ($\mathbf{U}_{agent}$).

Définition 6.4 (Opération autorisée pour un agent). *Soit k un agent et $\mathbf{U}_k = (\mathcal{A}_k, \mathcal{R}_k)$ son système d'argumentation. Une opération élémentaire o est* autorisée *pour k ssi*

- *$o.arg \subseteq \mathcal{A}_k$ et*
- *$o.att \subseteq \mathcal{R}_k$.*

L'ensemble des opérations autorisées pour k est noté $\mathcal{O}_k$.

Il est important de noter que dans le cas particulier d'une addition d'argument, il peut n'être fourni qu'une partie des attaques connues ; il est donc possible pour un agent d'effectuer un "mensonge par omission". Il n'est par contre pas autorisé de fournir des attaques n'étant pas connues et donc de mentir de manière "active" ; ceci pourra être l'objet de travaux futurs.

Exemple 6.3. *Parmi les quelques opérations élémentaires listées dans l'exemple 6.2, le procureur n'est pas autorisé à utiliser celles concernant des arguments ou des attaques qu'il ne connaît pas. Ainsi, il ne pourra pas utiliser les opérations suivantes :*

6. Ne pas avoir à spécifier les attaques à supprimer n'est qu'une simplification syntaxique. En effet, le fait que $\mathcal{R}_z$ doive être renseigné ne veut pas dire que l'on peut supprimer uniquement les attaques que l'on souhaite (la définition 2.6 page 16 le précise). Que ce soit dans le cas de la définition 2.6 page 16 ou dans celui de la définition 6.3 page précédente, il n'y aurait aucun sens à laisser dans le système d'argumentation des attaques concernant un argument n'étant plus dans le système ; les attaques supprimées sont donc, dans les deux cas, toutes celles qui concernent l'argument supprimé.

- $(\bigoplus, \{x_2\}, \{(x_2, x_1)\})$,
- $(\bigominus, \{x_7\}, \varnothing)$.
- $(\oplus, \varnothing, \{(x_3, x_2)\})$,
- $(\ominus, \varnothing, \{(x_7, x_4)\})$,

Remarquons que si le procureur vient à connaître, grâce à une opération de son adversaire et une mise à jour de son système, des arguments ou des attaques lui faisant défaut jusque-là, la ou les opérations correspondantes seront alors autorisées.

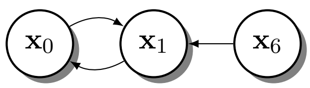
Une opération, bien qu'autorisée pour un agent, n'est pas nécessairement adaptée au système d'argumentation cible. Pour représenter cette nuance, nous définissons la notion d'opération exécutable. L'exécutabilité d'une opération est illustrée sur la figure 6.3 page 115 par le lien entre le système de l'agent entreprenant l'opération ($\mathbf{U}_{agent}$) et la zone grise contenant le système cible ($\mathbf{U}_{cible}$). Les conditions de l'exécutabilité d'une opération dépendent de sa nature (addition ou suppression d'un argument ou d'une attaque). Ainsi, l'addition d'un argument nécessite que cet argument ne soit pas dans le système cible, et que les éventuelles interactions associées concernent des arguments appartenant, eux, au système cible. La suppression d'un argument exige que celui-ci soit présent dans le système cible. L'addition d'une interaction réclame qu'elle ne soit pas dans le système cible, mais que les deux arguments concernés le soient. Enfin, la suppression d'une interaction requiert que cette interaction appartienne au système cible.

Définition 6.5 (**Opération exécutable par un agent sur un système d'argumentation**)**.** *Soit k un agent et $\mathcal{G} = (\mathcal{A}, \mathcal{R})$ un système d'argumentation quelconque. Une opération* exécutable par k sur $\mathcal{G}$ *est une opération $o = (op, arg, att)$ autorisée pour k telle que :*

- *si $o.op = \bigoplus$, alors $o.arg \not\subset \mathcal{A}$ et $\forall (x, y) \in o.att,\ (x \in \mathcal{A}$ ou $y \in \mathcal{A})$.*
- *si $o.op = \bigominus$, alors $o.arg \subseteq \mathcal{A}$.*
- *si $o.op = \oplus$, alors $o.att \not\subset \mathcal{R}$ et $\forall (x, y) \in o.att,\ (x \in \mathcal{A}$ et $y \in \mathcal{A})$.*
- *si $o.op = \ominus$, alors $o.att \subseteq \mathcal{R}$.*

L'ensemble des opérations exécutables par k sur $\mathcal{G}$ est noté $\mathcal{O}_k^{\mathcal{G}}$. Par définition, $\mathcal{O}_k^{\mathcal{G}} \subseteq \mathcal{O}_k$.

Exemple 6.4. *Au cours de l'audience, les différentes actions des protagonistes ont fait évoluer le système d'argumentation des jurés, de sorte que ce dernier soit le suivant :*



C'est maintenant au tour du procureur de prendre la parole et d'essayer de convaincre les jurés en modifiant leur système. Il sait qu'il n'a pas les éléments nécessaires pour faire en sorte que l'argument x_1 soit accepté sceptiquement et se rabat en conséquence sur une acceptabilité crédule. Pour cela, il n'a à sa disposition que quatre opérations autorisées sur celles vues dans l'exemple 6.2 page ci-contre :

- $o_1 = (\textcircled{\oplus}, \{x_1\}, \varnothing)$,
- $o_2 = (\ominus, \varnothing, \{(x_6, x_1)\})$,
- $o_3 = (\textcircled{\ominus}, \{x_4\}, \varnothing)$,
- $o_4 = (\oplus, \varnothing, \{(x_5, x_4)\})$.

Mais o_1 n'est pas exécutable, puisque x_1 est déjà dans le système d'argumentation des jurés ; après tout, il ne servirait à rien de répéter une chose qu'ils savent déjà. De la même manière, les opérations o_3 et o_4 ne peuvent être exécutées du fait de l'absence des arguments x_4 et x_5 dans le système des jurés. Il ne lui reste donc plus qu'à tenter de faire supprimer l'attaque de x_6 vers x_1 grâce à o_2.

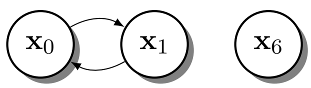
Une opération exécutable sur un système d'argumentation a un impact sur ce dernier. Cet impact est représenté sur la figure 6.3 page 115 par le lien entre $\mathbf{U}_{cible}$ et le système résultat ($\mathbf{U}'_{cible}$).

Définition 6.6 (Impact d'opération exécutable par un agent sur un système d'argumentation). *Soit k un agent, $\mathcal{G} = (\mathcal{A}, \mathcal{R})$ un système d'argumentation quelconque et $o = (op, arg, att)$ une opération exécutable par k sur $\mathcal{G}$. o fournit un nouveau système d'argumentation $\mathcal{G}' = o(\mathcal{G}) = (\mathcal{A}', \mathcal{R}')$ tel que :*

- $o.op = \textcircled{\oplus}$:
$$\mathcal{G}' = (\mathcal{A} \cup o.arg, \mathcal{R} \cup o.att)$$
- $o.op = \textcircled{\ominus}$:
$$\mathcal{G}' = (\mathcal{A} \setminus o.arg, \mathcal{R} \setminus \{(x, y) \in \mathcal{R} \mid x \in o.arg \text{ ou } y \in o.arg\})$$
- $o.op = \oplus$:
$$\mathcal{G}' = (\mathcal{A}, \mathcal{R} \cup o.att)$$
- $o.op = \ominus$:
$$\mathcal{G}' = (\mathcal{A}, \mathcal{R} \setminus o.att)$$

Notons que dans le cas de la suppression d'un argument, l'impact porte donc sur *toutes* les interactions dans $\mathcal{G}$ concernant l'argument appartenant à $o.arg$.

Exemple 6.5. *En exécutant l'opération $o_2 = (\ominus, \varnothing, \{(x_6, x_1)\})$, le procureur a modifié le système d'argumentation des jurés en supprimant l'attaque de l'argument x_6 vers l'argument x_1 :*



Ainsi, la menace de l'argument x_6 est pour le moment écartée !

Enfin, nous considérons les suites d'opérations exécutables par un agent sur un système d'argumentation, nommées *programmes*, donnant la possibilité à un agent d'effectuer plusieurs opérations élémentaires en séquence.

Définition 6.7 (Programme exécutable par un agent sur un système d'argumentation). *Soit k un agent et $\mathcal{G}$ un système d'argumentation quelconque. Un* programme *p exécutable par k sur $\mathcal{G}$ est une suite ordonnée finie de n opérations $\langle o_1, \cdots, o_n\rangle$ telle que :*

- *Pour $n = 1$: o_1 est exécutable par k sur $\mathcal{G}$ (*i.e. *$o_1 \in \mathcal{O}_k^{\mathcal{G}}$). On a alors $p(\mathcal{G}) = o_1(\mathcal{G})$.*
- *Pour $n > 1$: $\langle o_1, \cdots, o_{n-1}\rangle$ est un programme p' exécutable par k sur $\mathcal{G}$ tel que $p'(\mathcal{G}) = \mathcal{G}'$ et $o_n \in \mathcal{O}_k^{\mathcal{G}'}$ est une opération exécutable par k sur $\mathcal{G}'$. On a alors $p(\mathcal{G}) = o_n(\mathcal{G}')$.*
- *Par extension, une suite vide est aussi un programme. On a alors $p(\mathcal{G}) = \mathcal{G}$.*

n est appelé cardinal *de p, noté $|p|$. Par abus de notation, on écrira $o \in p$ pour $\exists o_i \in p = \langle o_1, \cdots, o_n\rangle$ telle que $o = o_i$. On notera $\mathcal{P}_k^{\mathcal{G}}$ l'ensemble de tous les programmes exécutables par l'agent k sur $\mathcal{G}$.*

Dans la section suivante, nous raffinons cette définition pour y inclure la notion de programme réalisant un ensemble de buts.

6.1.3 Prise en compte de buts

Dans notre travail, au-delà de la notion de changement et des modifications qu'il implique, nous nous intéressons à ce qui peut provoquer ce changement. Il est donc intéressant de se demander, outre le comment, pourquoi un système d'argumentation cible est modifié. Nous nous plaçons ici dans le contexte d'un dialogue entre agents, il est ainsi naturel de penser que les buts découlent d'un désir particulier sur la discussion en cours.

Comme nous l'avons déjà souligné, un agent a la possibilité d'agir sur un système cible pour atteindre des objectifs. Ces objectifs seront formellement traduits par des buts. Un but est représenté par une expression d'un langage qui exprime des conditions à satisfaire dans des systèmes d'argumentation (celui de la cible et éventuellement celui de l'agent). Pour représenter ces buts, nous utiliserons les symboles apparaissant dans la typologie des propriétés du changement présentée dans le chapitre 4 page 59, notamment :

- les arguments (x, y, z, etc.),
- les extensions ($\mathcal{E}_i$ et $\mathcal{E}'_i$),
- les ensembles d'extensions ($\mathbf{E}$ et $\mathbf{E}'$) ainsi que leur cardinal ($|\mathbf{E}|$ et $|\mathbf{E}'|$),
- les ensembles d'extensions contenant un argument particulier x ($\mathbf{E}_x$ et $\mathbf{E}'_x$) et leur cardinal ($|\mathbf{E}_x|$ et $|\mathbf{E}'_x|$),
- les opérateurs de comparaison classique ($=$, $<$, $>$, etc.),
- les quantificateurs $\forall$ et $\exists$,
- l'appartenance ($\in$) et l'inclusion ($\subseteq$),
- l'union ($\cup$) et l'intersection ($\cap$) d'ensembles,

- les opérateurs logiques classiques ($\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\neg$).

Dans la suite de ce document, nous noterons $\mathcal{B}_k$ l'ensemble des buts de l'agent k.[7]

Un but **absolu** ne prend pas en compte le système d'argumentation initial (appelé système cible), mais uniquement le système d'argumentation obtenu après modification du système cible (appelé système résultat). Formellement, il ne porte donc que sur $\mathcal{G}'$. Un tel but peut être utilisé pour représenter un désir à long terme, que l'on a tout au long de la discussion.

Ainsi, nous aurons au sein de cette catégorie des buts tels que *"Je veux que l'argument x soit rejeté"*, formalisé par $|\mathbf{E}'_x| = 0$, ou *"Je veux qu'il n'existe qu'une seule extension non vide"*, formalisé par $|\mathbf{E}'| = 1,\ \mathcal{E}' \neq \varnothing$. Notons que, puisque le système cible n'est pas pris en compte, un agent peut avoir un ou plusieurs buts déjà satisfaits ; dans ce cas, il n'aura donc aucun changement à entreprendre pour voir ces buts satisfaits.

Un but **relatif** prend, quant à lui, en compte le système d'argumentation cible. Formellement, il porte sur le couple $(\mathcal{G}, \mathcal{G}')$, rassemblant un système cible dans son état initial et ce système après le changement. Un tel but permet de représenter l'évolution que l'on souhaite voir sur un système à un moment précis de la discussion.

Cette catégorie regroupe des buts tels que *"Je veux recentrer la discussion sur un nombre réduit d'options"*, formalisé par $|\mathbf{E}| > |\mathbf{E}'|$, ou *"Je veux qu'aucun nouvel argument ne soit accepté"*, formalisé par $\forall \mathcal{E}'_j \in \mathbf{E}', \exists \mathcal{E}_i \in \mathbf{E}, \mathcal{E}'_j \subseteq \mathcal{E}_i$. Ici, il faudra nécessairement que la vérification de la condition soit effectuée en partie sur le système d'argumentation cible initial.

Le lecteur pourra se référer à l'annexe D.1 page 221 (ainsi qu'en partie au chapitre 3 page 29) pour une liste plus complète de buts, ainsi qu'une intuition pour chacun d'entre eux.

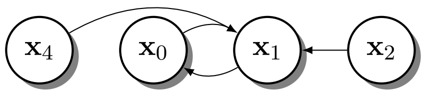
Nous pouvons maintenant définir la notion de programme réalisant un ensemble de buts.

Définition 6.8 (Programme d'un agent sur un système d'argumentation réalisant un ensemble de buts). *Soit k un agent, $\mathcal{B}_k$ l'ensemble de buts de k et $\mathcal{G}$ un système d'argumentation quelconque. Un* programme p de k sur $\mathcal{G}$ réalisant $\mathcal{B}_k$ *est un programme exécutable par k sur $\mathcal{G}$ tel que $p(\mathcal{G}) = \mathcal{G}'$ et tous les buts $b \in \mathcal{B}_k$ sont satisfaits dans $\mathcal{G}'$. On notera $\mathcal{P}^{\mathcal{G}}_{\mathcal{B}_k}$ l'ensemble de tous les programmes de l'agent k sur $\mathcal{G}$ réalisant l'ensemble de buts $\mathcal{B}_k$. Par définition, $\mathcal{P}^{\mathcal{G}}_{\mathcal{B}_k} \subseteq \mathcal{P}^{\mathcal{G}}_k$.*

Exemple 6.6. *Les choses ne se sont pas passées comme prévu pour le procureur. Son adversaire, qui a très bien défendu son client, a réussi à défaire toutes les tentatives de notre représentant du ministère public de satisfaire son but (faire accepter*

7. Notons que la notion de but est ici volontairement laissée vague puisque sa formalisation nécessite un langage logique qui ne sera introduit que dans le chapitre 7 page 135 ; la notion formelle de but sera donc introduite dans la définition 7.5 page 144.

crédulement x_1 *:* $\mathcal{B}_{proc} = \{\exists \mathcal{E} \in \mathbf{E}, x_1 \in \mathcal{E}\}$*). Suite à cela, le système d'argumentation des jurés est le suivant :*



Le procureur semble en mauvaise posture, mais de nouvelles informations lui ont remis le pied à l'étrier ! Il a notamment appris l'existence de l'argument x_2 *et s'est rendu compte que son argument* x_3 *pouvait le contrer. Il a désormais à sa disposition, entre autres choses, deux opérations qui collent parfaitement à la situation :*

- $o_5 = (\oplus, \{x_3\}, \{(x_3, x_2)\})$
- $o_6 = (\oplus, \{x_5\}, \{(x_5, x_4)\})$

Grâce à ces deux opérations, il peut revenir dans la partie, en une prise de parole contrer les deux arguments de l'avocat et réaliser son but.

6.2 Choix des opérations de changement et changement optimal

Il est désormais temps d'étudier plus en profondeur le cœur de ce chapitre, c'est-à-dire le choix à proprement parler d'un changement à effectuer.

En effet, le changement, sujet principal de ce travail, est le fait, pour un agent, d'effectuer une modification sur un système d'argumentation par le biais d'un programme, en particulier pour satisfaire un ou plusieurs buts. Mais il existe potentiellement plusieurs changements différents pour arriver à ses fins. Comment choisir alors le changement à accomplir parmi tous ceux permettant de satisfaire un même but ? Il semble nécessaire de pouvoir comparer ces changements, et ce, selon différents “critères”. Ainsi, pour un ou plusieurs critères donnés, on cherche un “meilleur” changement, c'est-à-dire un programme réalisant les buts de l'agent et optimisant ce critère.

Intuitivement, un critère marque une préférence sur le *moyen* d'atteindre son but : un agent choisissant un critère particulier annonce implicitement qu'il préfère satisfaire son but grâce à un changement respectant ce critère.

Nous pouvons distinguer trois critères principaux, qu'il est possible de raffiner en critères plus précis, portant sur :

- le coût d'un programme,
- la distance structurelle entre deux systèmes d'argumentation,
- la distance sémantique entre deux systèmes d'argumentation (c'est-à-dire basée sur leurs extensions).

Nous nous proposons, dans la suite de cette partie, de définir plusieurs de ces critères indépendamment de la notion de satisfaction de but. Nous définirons ensuite plusieurs relations de comparaison grâce à ces mêmes critères. Pour illustrer cela, nous utiliserons tout au long des définitions un exemple où le procureur doit choisir au sein d'un ensemble de programmes celui qu'il va utiliser.

Exemple 6.6 (cont.). *Le procureur a donc choisi d'utiliser le programme* $\langle(\bigoplus, \{x_3\}, \{(x_3, x_2)\}), (\bigoplus, \{x_5\}, \{(x_5, x_4)\})\rangle$. *Mais comment en est-il venu à faire ce choix ? Qu'est-ce qui l'a poussé à énoncer cela plutôt qu'autre chose de façon à faire accepter crédulement l'argument* x_1 *?*

En effet, avant sa prise de parole, le procureur, sachant que les jurés utilisent la sémantique préférée pour délibérer, avait pour but d'assurer au moins un point de vue dans lequel le client de l'avocat n'est pas innocent (donc que $|\mathbf{E}'_{x_1}| \neq 0$*).*

Ainsi, divers choix possibles s'offraient à lui ; la table 6.1 en propose une partie en considérant que l'opération $(\bigoplus, \{x_3\}, \{(x_3, x_2)\})$*, notée* o_5*, est présente dans chacun des programmes étudiés. En outre, les systèmes d'argumentation qui auraient été obtenus par l'utilisation de ces programmes sont illustrés par la figure 6.4 page ci-contre.*

	Programmes	Extensions des systèmes obtenus
p_1	$\langle o_5, (\bigoplus, \{x_5\}, \{(x_5, x_4)\})\rangle$	$\{\{x_0, x_5\}, \{x_1, x_5\}\}$
p_2	$\langle o_5, (\bigoplus, \{x_5\}, \{(x_5, x_4)\}), (\ominus, \varnothing, \{(x_0, x_1)\})\rangle$	$\{\{x_1, x_5\}\}$
p_3	$\langle o_5, (\bigoplus, \{x_5\}, \{(x_5, x_4)\}), (\bigominus, \{x_4\}, \varnothing)\rangle$	$\{\{x_0, x_5\}, \{x_1, x_5\}\}$
p_4	$\langle o_5, (\bigoplus, \{x_5\}, \{(x_5, x_4)\}), (\bigominus, \{x_5\}, \varnothing), (\bigominus, \{x_4\}, \varnothing)\rangle$	$\{\{x_0\}, \{x_1\}\}$
p_5	$\langle o_5, (\bigominus, \{x_4\}, \varnothing)\rangle$	$\{\{x_0\}, \{x_1\}\}$
p_6	$\langle o_5, (\ominus, \varnothing, \{(x_4, x_1)\})\rangle$	$\{\{x_0, x_4\}, \{x_1, x_4\}\}$
p_7	$\langle o_5, (\ominus, \varnothing, \{(x_4, x_1)\}), (\ominus, \varnothing, \{(x_0, x_1)\})\rangle$	$\{\{x_1, x_4\}\}$
p_8	$\langle o_5, (\bigominus, \{x_4\}, \varnothing), (\ominus, \varnothing, \{(x_0, x_1)\})\rangle$	$\{\{x_1\}\}$

Table 6.1 – *Liste des divers programmes que le procureur peut exécuter (les extensions sont calculées dans le cadre de la sémantique préférée.*

Ayant listé ses options, il ne manquait au procureur qu'un moyen de leur attribuer une valeur pour pouvoir les comparer et choisir la meilleure d'entre elles.

Notons que la plupart des critères que nous utiliserons dans ce travail représentent une distance entre deux graphes, et portent donc sur deux objets (à savoir un premier graphe, et le graphe résultant de la modification du premier par un programme). Bien qu'il soit possible de le faire uniquement grâce à $p(\mathcal{G})$ (le système

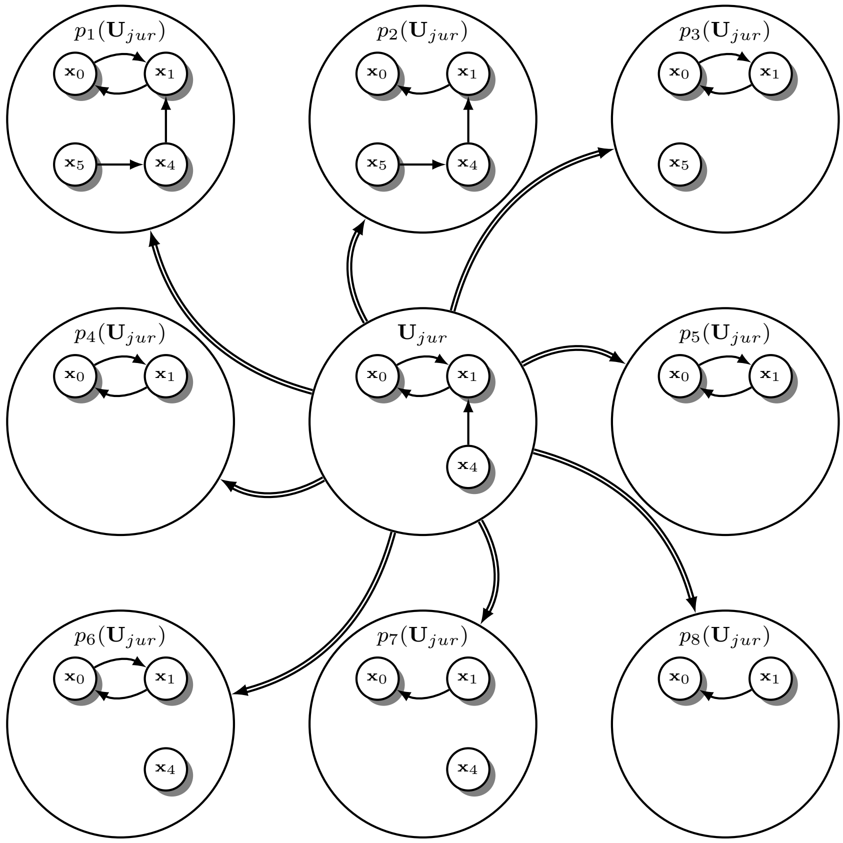


Figure 6.4 – *Illustration des différentes (et arbitraires) possibilités de changement du système d'argumentation des jurés (*$\mathbf{U}_{jur}$*).*

résultant du programme p effectué sur $\mathcal{G}$), il nous paraît plus intuitif dans ce cas de définir une valeur associée au critère grâce au couple $(\mathcal{G}, p(\mathcal{G}))$.

Voyons maintenant quelques critères possibles.

6.2.1 Coût de programmes

Le critère de coût des programmes se base, comme son nom l'indique, sur le coût engendré par un programme, et ce sans tenir compte des systèmes d'argumentation ou des modifications qu'ils subissent. Il est bien entendu lié au coût des opérations qui le constituent. Par exemple, il est possible d'utiliser une fonction de coût d'une opération (élémentaire) pour un agent.

Définition 6.9 (**Coût d'une opération**). *Soit* Coût *une fonction telle que*

$$\begin{aligned} Co\hat{u}t: \ & Agents \times \mathcal{O} \to \mathbb{N} \cup \{\infty\} \\ & (k, o) \mapsto \infty \ si \ o \notin \mathcal{O}_k \end{aligned}$$

où Agents est l'ensemble des agents, $\mathcal{O}$ *est l'ensemble des opérations élémentaires et* $\mathcal{O}_k$ *est l'ensemble des opérations autorisées pour l'agent* k.

Le critère de coût peut également être modulé par tout ce qui a trait à la notion de programme, comme la taille, la nature des opérations, etc. Nous nous proposons ici de définir le coût d'un programme à partir de sa taille.

Taille de programme

La taille d'un programme, notée t, est le nombre total d'opérations entreprises par l'agent lors de ce programme. Ainsi, ne faire qu'une opération sera moins coûteux qu'en faire deux, ce qui peut par exemple permettre d'économiser son temps de parole, et d'un point de vue computationnel minimiser la complexité du changement. Notons que nous supposons pour le moment que chaque opération a le même coût. Ce dernier pourrait néanmoins être modulé grâce à la fonction *Coût* définie précédemment.

Exemple 6.7. *Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table 6.1 page 122 :*

- $|p_1| = 1$
- $|p_2| = 2$
- $|p_3| = 2$
- $|p_4| = 3$
- $|p_5| = 1$
- $|p_6| = 1$
- $|p_7| = 2$
- $|p_8| = 2$

6.2.2 Distance structurelle

Le critère de distance structurelle se base sur le nombre de différences entre les systèmes d'argumentation (et donc leurs graphes) avant et après le changement. Il peut être raffiné en quatre nouveaux critères, que nous listons ci-dessous.

Distance de perte d'arguments.

Ce critère, noté pa, se base sur le nombre d'arguments appartenant au graphe avant le changement et n'appartenant pas au graphe après le changement. Ceci pourrait également être modulé suivant l'importance de certains arguments.

Pour ce faire, nous considérons la fonction permettant d'obtenir le nombre d'arguments perdus entre deux systèmes d'argumentation.

Définition 6.10 (Distance de perte d'arguments). *La distance d'arguments perdus entre deux systèmes d'argumentation $(\mathcal{A}, \mathcal{R})$ et $(\mathcal{A}', \mathcal{R}')$ est un entier naturel défini par la fonction suivante :*

$$f_{pa} : \quad \mathbf{G_U} \times \mathbf{G_U} \to \mathbb{N}$$
$$((\mathcal{A}, \mathcal{R}), (\mathcal{A}', \mathcal{R}')) \mapsto |\mathcal{A} \setminus \mathcal{A}'|.$$

Exemple 6.8. *Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table 6.1 page 122 :*

- $f_{pa}(\mathbf{U}_{jur}, p_1(\mathbf{U}_{jur})) = 0$
- $f_{pa}(\mathbf{U}_{jur}, p_2(\mathbf{U}_{jur})) = 0$
- $f_{pa}(\mathbf{U}_{jur}, p_3(\mathbf{U}_{jur})) = 1$
- $f_{pa}(\mathbf{U}_{jur}, p_4(\mathbf{U}_{jur})) = 1$
- $f_{pa}(\mathbf{U}_{jur}, p_5(\mathbf{U}_{jur})) = 1$
- $f_{pa}(\mathbf{U}_{jur}, p_6(\mathbf{U}_{jur})) = 0$
- $f_{pa}(\mathbf{U}_{jur}, p_7(\mathbf{U}_{jur})) = 0$
- $f_{pa}(\mathbf{U}_{jur}, p_8(\mathbf{U}_{jur})) = 1$

Distance de gain d'arguments.

Ce critère, noté *ga* se base, à l'inverse du précédent, sur le nombre d'arguments n'appartenant pas au graphe avant le changement et appartenant au graphe après le changement. Ceci pourrait également être modulé suivant l'importance de certains arguments.

Pour ce faire, nous considérons la fonction permettant d'obtenir le nombre d'arguments gagnés entre deux systèmes d'argumentation.

Définition 6.11 (**Distance de gain d'arguments**). *La distance d'arguments gagnés entre deux systèmes d'argumentation* $(\mathcal{A}, \mathcal{R})$ *et* $(\mathcal{A}', \mathcal{R}')$ *est un entier naturel défini par la fonction suivante :*

$$\begin{aligned} f_{ga} : \quad & \mathbf{G_U} \times \mathbf{G_U} \to \mathbb{N} \\ & ((\mathcal{A}, \mathcal{R}), (\mathcal{A}', \mathcal{R}')) \mapsto |\mathcal{A}' \setminus \mathcal{A}|. \end{aligned}$$

Exemple 6.9. *Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table* 6.1 page 122 *:*

- $f_{ga}(\mathbf{U}_{jur}, p_1(\mathbf{U}_{jur})) = 1$
- $f_{ga}(\mathbf{U}_{jur}, p_2(\mathbf{U}_{jur})) = 1$
- $f_{ga}(\mathbf{U}_{jur}, p_3(\mathbf{U}_{jur})) = 1$
- $f_{ga}(\mathbf{U}_{jur}, p_4(\mathbf{U}_{jur})) = 0$
- $f_{ga}(\mathbf{U}_{jur}, p_5(\mathbf{U}_{jur})) = 0$
- $f_{ga}(\mathbf{U}_{jur}, p_6(\mathbf{U}_{jur})) = 0$
- $f_{ga}(\mathbf{U}_{jur}, p_7(\mathbf{U}_{jur})) = 0$
- $f_{ga}(\mathbf{U}_{jur}, p_8(\mathbf{U}_{jur})) = 0$

Distance de perte d'attaques.

Ce critère, noté *pi*, se base sur le nombre d'attaques appartenant au graphe avant le changement et n'appartenant pas au graphe après le changement. Ceci pourrait également être modulé suivant l'importance de certaines attaques.

Pour ce faire, nous considérons la fonction permettant d'obtenir le nombre d'attaques perdues entre deux systèmes d'argumentation.

Définition 6.12 (**Distance de perte d'attaques**). *La distance d'attaques perdues entre deux systèmes d'argumentation* $(\mathcal{A}, \mathcal{R})$ *et* $(\mathcal{A}', \mathcal{R}')$ *est un entier naturel défini par la fonction suivante :*

$$\begin{aligned} f_{pi} : \quad & \mathbf{G_U} \times \mathbf{G_U} \to \mathbb{N} \\ & ((\mathcal{A}, \mathcal{R}), (\mathcal{A}', \mathcal{R}')) \mapsto |\mathcal{R} \setminus \mathcal{R}'|. \end{aligned}$$

Exemple 6.10. *Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table* 6.1 page 122 *:*

- $f_{pi}(\mathbf{U}_{jur}, p_1(\mathbf{U}_{jur})) = 0$
- $f_{pi}(\mathbf{U}_{jur}, p_2(\mathbf{U}_{jur})) = 1$
- $f_{pi}(\mathbf{U}_{jur}, p_3(\mathbf{U}_{jur})) = 1$
- $f_{pi}(\mathbf{U}_{jur}, p_4(\mathbf{U}_{jur})) = 1$
- $f_{pi}(\mathbf{U}_{jur}, p_5(\mathbf{U}_{jur})) = 1$
- $f_{pi}(\mathbf{U}_{jur}, p_6(\mathbf{U}_{jur})) = 1$
- $f_{pi}(\mathbf{U}_{jur}, p_7(\mathbf{U}_{jur})) = 2$
- $f_{pi}(\mathbf{U}_{jur}, p_8(\mathbf{U}_{jur})) = 2$

Distance de gain d'attaques.

Ce critère, noté *gi*, se base, à l'inverse du précédent, sur le nombre d'attaques n'appartenant pas au graphe avant le changement et appartenant au graphe après le changement. Ceci pourrait également être modulé suivant l'importance de certaines attaques.

Pour ce faire, nous considérons la fonction permettant d'obtenir le nombre d'attaques gagnées entre deux systèmes d'argumentation.

Définition 6.13 (**Distance de gain d'attaques**). *La distance d'attaques gagnées entre deux systèmes d'argumentation* $(\mathcal{A}, \mathcal{R})$ *et* $(\mathcal{A}', \mathcal{R}')$ *est un entier naturel défini par la fonction suivante :*

$$\begin{aligned} f_{gi} : \quad & \mathbf{G_U} \times \mathbf{G_U} \to \mathbb{N} \\ & ((\mathcal{A}, \mathcal{R}), (\mathcal{A}', \mathcal{R}')) \mapsto |\mathcal{R}' \setminus \mathcal{R}|. \end{aligned}$$

Exemple 6.11. *Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table* *6.1 page 122* *:*

- $f_{gi}(\mathbf{U}_{jur}, p_1(\mathbf{U}_{jur})) = 1$
- $f_{gi}(\mathbf{U}_{jur}, p_2(\mathbf{U}_{jur})) = 1$
- $f_{gi}(\mathbf{U}_{jur}, p_3(\mathbf{U}_{jur})) = 0$
- $f_{gi}(\mathbf{U}_{jur}, p_4(\mathbf{U}_{jur})) = 0$
- $f_{gi}(\mathbf{U}_{jur}, p_5(\mathbf{U}_{jur})) = 0$
- $f_{gi}(\mathbf{U}_{jur}, p_6(\mathbf{U}_{jur})) = 0$
- $f_{gi}(\mathbf{U}_{jur}, p_7(\mathbf{U}_{jur})) = 0$
- $f_{gi}(\mathbf{U}_{jur}, p_8(\mathbf{U}_{jur})) = 0$

6.2.3 Distance sémantique

Le critère de distance sémantique se base sur le nombre de différences entre les extensions avant et après le changement. Il peut être raffiné en deux nouveaux critères que nous listons ci-dessous.

Distance de perte d'acceptabilité.

Ce critère, noté *pe*, se base sur le nombre d'arguments étant acceptés crédulement avant le changement et étant rejetés après le changement. Ceci pourrait également être modulé suivant l'importance de certains arguments.

Pour ce faire, nous considérons la fonction qui, selon deux systèmes d'argumentation, permet d'obtenir le nombre d'arguments étant acceptés crédulement dans le premier système et rejetés dans le second, c'est-à-dire le nombre d'arguments acceptés crédulement qui ont été perdus dans le changement.

Définition 6.14 (**Distance de perte d'acceptabilité**). *La distance d'arguments acceptés crédulement et perdus entre deux systèmes d'argumentation* $(\mathcal{A}, \mathcal{R})$ *et* $(\mathcal{A}', \mathcal{R}')$ *est un entier naturel défini par la fonction suivante :*

$$\begin{aligned} f_{pe} : \quad & \mathbf{G_U} \times \mathbf{G_U} \rightarrow \mathbb{N} \\ ((\mathcal{A}, \mathcal{R}), (\mathcal{A}', \mathcal{R}')) & \mapsto \mid \bigcup_{\mathcal{E} \in \mathbf{E}} \mathcal{E} \setminus \bigcup_{\mathcal{E}' \in \mathbf{E}'} \mathcal{E}' \mid . \end{aligned}$$

Notons qu'il est possible de moduler cette distance en considérant les arguments acceptés sceptiquement. Pour ce faire, il faudra utiliser l'intersection des extensions plutôt que l'union.

Exemple 6.12. *Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table* *6.1 page 122* *:*

- $f_{pe}(\mathbf{U}_{jur}, p_1(\mathbf{U}_{jur})) = 1$
- $f_{pe}(\mathbf{U}_{jur}, p_2(\mathbf{U}_{jur})) = 2$
- $f_{pe}(\mathbf{U}_{jur}, p_3(\mathbf{U}_{jur})) = 1$
- $f_{pe}(\mathbf{U}_{jur}, p_4(\mathbf{U}_{jur})) = 1$
- $f_{pe}(\mathbf{U}_{jur}, p_5(\mathbf{U}_{jur})) = 1$
- $f_{pe}(\mathbf{U}_{jur}, p_6(\mathbf{U}_{jur})) = 0$
- $f_{pe}(\mathbf{U}_{jur}, p_7(\mathbf{U}_{jur})) = 1$
- $f_{pe}(\mathbf{U}_{jur}, p_8(\mathbf{U}_{jur})) = 2$

Distance de gain d'acceptabilité.

À l'inverse du précédent, ce critère, noté *ge*, se base sur le nombre d'arguments étant rejetés avant le changement et étant acceptés crédulement après le changement. Ceci pourrait également être modulé suivant l'importance de certains arguments.

Pour ce faire, nous considérons la fonction qui, selon deux systèmes d'argumentation, permet d'obtenir le nombre d'arguments étant rejetés dans le premier système et acceptés crédulement dans le second, c'est-à-dire le nombre d'arguments acceptés crédulement qui ont été gagnés dans le changement.

Définition 6.15 (**Distance de gain d'acceptabilité**). *La distance d'arguments acceptés crédulement et gagnés entre deux systèmes d'argumentation* $(\mathcal{A}, \mathcal{R})$ *et* $(\mathcal{A}', \mathcal{R}')$ *est un entier naturel défini par la fonction suivante :*

$$\begin{aligned} f_{ge} : \quad & \mathbf{G_U} \times \mathbf{G_U} \rightarrow \mathbb{N} \\ ((\mathcal{A}, \mathcal{R}), (\mathcal{A}', \mathcal{R}')) & \mapsto \mid \bigcup_{\mathcal{E}' \in \mathbf{E}'} \mathcal{E}' \setminus \bigcup_{\mathcal{E} \in \mathbf{E}} \mathcal{E} \mid . \end{aligned}$$

Notons qu'ici aussi, il est possible de moduler cette distance en considérant les arguments acceptés sceptiquement. Pour ce faire, il faudra utiliser l'intersection des extensions plutôt que l'union.

Exemple 6.13. *Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table* *6.1 page 122* *:*

- $f_{ge}(\mathbf{U}_{jur}, p_1(\mathbf{U}_{jur})) = 1$
- $f_{ge}(\mathbf{U}_{jur}, p_2(\mathbf{U}_{jur})) = 2$
- $f_{ge}(\mathbf{U}_{jur}, p_3(\mathbf{U}_{jur})) = 2$
- $f_{ge}(\mathbf{U}_{jur}, p_4(\mathbf{U}_{jur})) = 1$
- $f_{ge}(\mathbf{U}_{jur}, p_5(\mathbf{U}_{jur})) = 1$
- $f_{ge}(\mathbf{U}_{jur}, p_6(\mathbf{U}_{jur})) = 1$
- $f_{ge}(\mathbf{U}_{jur}, p_7(\mathbf{U}_{jur})) = 1$
- $f_{ge}(\mathbf{U}_{jur}, p_8(\mathbf{U}_{jur})) = 1$

6.2.4 Comparaison de programmes

Maintenant que les critères sont définis, nous pouvons les utiliser pour comparer à proprement parler les changements. En effet, pour deux programmes et plusieurs critères, un agent doit pouvoir dire quel programme est au moins aussi bon que l'autre.

Il est important de noter en premier lieu qu'il existe deux types de critères, à savoir les critères *cardinaux* et les critères *ordinaux* (Grabisch et Perny (2003)). Les critères cardinaux correspondent, comme leur nom l'indique, à ce qui peut être quantifié (par exemple, le nombre d'arguments disparaissant, le nombre d'attaques apparaissant, etc.). Les critères ordinaux, quant à eux, correspondent à ce qui ne peut être représenté de façon numérique (par exemple, l'inclusion de programmes [8]).

Les critères définis ci-avant sont du premier type ; les critères ordinaux ne sont pas traités dans ce travail. Notons que pour ces derniers, un changement optimal correspond implicitement à un changement minimal, ce qui revient à considérer le programme ayant la valeur la plus basse pour un de ces critères. Aussi, un programme p_1 est meilleur qu'un programme p_2 au regard d'un critère particulier si p_1 a une valeur plus basse que p_2 pour ce critère. D'autres critères sont possibles. On pourrait notamment penser à ceux qu'il faut maximiser ; ces derniers peuvent être ramenés à un changement minimal en faisant en sorte que la fonction leur correspondant donne un entier négatif.

Ainsi, dans ce travail, nous nous concentrons uniquement sur des critères cardinaux à minimiser et nous laissons le reste à des travaux futurs.

Nous souhaitons pouvoir combiner les critères cardinaux. Pour ce faire, il est possible d'associer aux critères des relations de préférence.

Définition 6.16 (Relation de préférence entre critères). *Soit k un agent, $\mathcal{G} \in \mathbf{G_U}$ un système d'argumentation quelconque, $\gamma \in \{t, pa, ga, pi, gi, pe, ge\}$ un critère de comparaison, f_γ la fonction correspondant au critère γ, $p_1 \in \mathcal{P}_k^{\mathcal{G}}$ et $p_2 \in \mathcal{P}_k^{\mathcal{G}}$ deux programmes exécutables par k sur $\mathcal{G}$. Soit $\preccurlyeq^\gamma$ la relation de préférence sur $\mathcal{P}_k$ telle que $p_1 \preccurlyeq^\gamma p_2$ (p_1 est au moins aussi bon que p_2) si et seulement si*

$$f_\gamma(\mathcal{G}, p_1(\mathcal{G})) \leq f_\gamma(\mathcal{G}, p_2(\mathcal{G})).$$

8. L'inclusion de programmes est un critère ordinal s'intéressant au fait qu'un programme constitue une "sous-séquence" d'un autre programme, c'est-à-dire si toutes les opérations de changements du premier programme apparaissent (dans le même ordre) dans le second, ce qui ne peut être quantifié. Il est à noter que considérer un critère ordinal peut amener à une situation où les programmes ne peuvent pas nécessairement être tous comparés (par exemple, dans le cas de l'inclusion de programmes, lorsque un programme ne peut être inclus dans (ni inclure) un autre programme).

Notons que les définitions des relations $\approx^\gamma$ et $\prec^\gamma$ se font de manière classique à partir de $\preccurlyeq^\gamma$ (voir annexe B page 195).

Exemple 6.14. *Étant désormais capable de comparer les huit programmes disponibles, le procureur peut faire un classement de ces derniers suivant un critère particulier et les valeurs calculées pour chacun d'entre eux. Ainsi, s'il ne désirait que privilégier le critère portant sur la taille des programmes (t), il obtiendrait :*

$$\begin{array}{llllllll} & p_1 & \approx^t & p_5 & \approx^t & p_6 & & \\ \prec^t & p_2 & \approx^t & p_3 & \approx^t & p_7 & \approx^t & p_8 \\ \prec^t & p_4 & & & & & & \end{array}$$

Les trois meilleurs changements au regard du critère t sont donc les programmes p_1, p_5 et p_6.

Si sa préoccupation première avait été de minimiser le nombre d'arguments n'étant plus acceptés après sa prise de parole, le procureur aurait considéré le critère portant sur la perte d'acceptabilité (pe). Il aurait alors obtenu :

$$\begin{array}{llllllll} & p_6 & & & & & & \\ \prec^{pe} & p_1 & \approx^{pe} & p_3 & \approx^{pe} & p_5 & \approx^{pe} & p_7 \\ \prec^{pe} & p_2 & \approx^{pe} & p_8 & & & & \end{array}$$

Le meilleur changement au regard du critère pe aurait donc été le programme p_6.

6.2.5 Combinaison de critères

Un agent n'ayant pas forcément qu'une seule préoccupation, il peut souhaiter considérer plusieurs critères. Notamment, si cet agent souhaite optimiser à la fois la perte et le gain d'arguments au sein du système d'argumentation, c'est-à-dire optimiser la distance symétrique entre $\mathcal{A}$ et $\mathcal{A}'$, il lui faudra prendre en compte les deux critères *pa* et *ga* à la fois. Dans ce cas, il lui faut alors être capable de combiner ces divers critères.

Il existe deux approches pour combiner des critères (Grabisch et Perny (2003)) ; étant donné deux objets (au sens large) et n critères, on peut :

- soit agréger les n critères sur chacun des objets, puis comparer l'agrégat (approche dite **AC**),
- soit comparer les objets sur chacun des n critères, puis agréger les différentes relations de comparaison (approche dite **CA**).

Approche AC

L'approche **AC** se propose de synthétiser pour un même objet les valeurs des différents critères en une valeur “globale” de l'objet, et ce, grâce à un opérateur d'agrégation. Deux objets sont ensuite comparés en comparant leurs valeurs globales.

Somme Dans cette approche, il est possible d'ajouter les valeurs des fonctions associées à chaque critère considéré (et éventuellement pondéré). [9] La comparaison de deux programmes est alors similaire à celle effectuée précédemment, les valeurs associées aux programmes étant simplement des additions de valeurs suivant plusieurs critères. Nous obtenons ainsi :

Définition 6.17 (Combinaison par somme). *Soit k un agent, $\mathcal{G} \in \mathbf{G_U}$ un système d'argumentation quelconque, $\Gamma \subseteq \{pa, ga, pi, gi, pe, ge\}$ un ensemble non vide de critères, f_γ la fonction correspondant au critère $\gamma \in \Gamma$ et $p \in \mathcal{P}_k$ un programme. La valeur globale de p est égale à :*

$$\sum_{\gamma \in \Gamma} f_\gamma(\mathcal{G}, p(\mathcal{G})).$$

Approche CA

L'approche **CA** se propose, elle, de comparer critère par critère les différents objets mis en jeu et d'agréger les relations de préférences. Ici, c'est donc une relation de préférence globale qui est synthétisée, à partir des préférences locales (c'est-à-dire pour chaque critère), ce qui constitue une agrégation ordinale. Plus précisément, étant donné plusieurs relations $\preccurlyeq^\gamma$, on cherche une relation $\preccurlyeq^\Gamma$, où Γ est un ensemble ou une suite de critères γ, qui synthétise les relations $\preccurlyeq^\gamma$. Nous proposons trois approches pour synthétiser ces relations, à savoir les combinaisons par unanimité, leximin et majorité, notées respectivement $\preccurlyeq^{\Gamma-una}$, $\preccurlyeq^{\Gamma-lex}$ et $\preccurlyeq^{\Gamma-maj}$.

Unanimité L'approche de l'unanimité permet une comparaison "stricte" entre deux programmes. En effet, pour qu'un programme soit préféré globalement à un autre programme, le premier doit être préféré au second pour tout critère pris en compte. La définition formelle est la suivante :

Définition 6.18 (Combinaison par unanimité). *Soit k un agent, $\Gamma \subseteq \{t, pa, ga, pi, gi, pe, ge\}$ un ensemble non vide de critères, $p_1 \in \mathcal{P}_k$ et $p_2 \in \mathcal{P}_k$ deux programmes. On définit $\preccurlyeq^{\Gamma-una}$ sur $\mathcal{P}_k$ par $p_1 \preccurlyeq^{\Gamma-una} p_2$ si et seulement si*

$$\forall \gamma \in \Gamma,\ p_1 \preccurlyeq^\gamma p_2.$$

Leximin Cette approche, raffinement de l'approche de l'unanimité, permet d'introduire une notion de priorité dans les critères considérés. En effet, si l'on suppose une comparaison entre deux programmes, une combinaison lexicographique de plusieurs critères signifie que le premier critère détermine quel programme est le meilleur parmi les deux, et dans le cas où ce premier critère ne peut départager les deux programmes, une comparaison sera faite grâce au second critère, et ainsi de suite jusqu'au dernier critère. Plus formellement, nous obtenons de nouvelles relations de comparaison :

9. Nous supposons ici que nous ne pouvons utiliser que des critères de même "nature". Ainsi, nous ne l'utiliserons ici que sur les critères de distance structurelle ou sémantique.

Définition 6.19 (**Combinaison par leximin**). *Soit k un agent, Γ une suite ordonnée de n critères distincts telle que $\forall \gamma_i \in \Gamma,\ \gamma_i \in \{t, pa, ga, pi, gi, pe, ge\}$, $p_1 \in \mathcal{P}_k$ et $p_2 \in \mathcal{P}_k$ deux programmes. On définit $\prec^{\Gamma-lex}$ sur $\mathcal{P}_k$ par $p_1 \prec^{\Gamma-lex} p_2$ si et seulement si*

$$\exists i \leq n \text{ t.q. } p_1 \prec^{\gamma_i} p_2 \text{ et}$$
$$\forall j < i,\ p_1 \approx^{\gamma_j} p_2.$$

Majorité La combinaison par majorité, comme son nom l'indique, détermine quel programme sera préféré au regard d'une majorité de critères. Plus précisément, si pour un ensemble de critères, un programme est meilleur qu'un autre pour un plus grand nombre de ces critères, alors il lui sera préféré pour cet ensemble.

Définition 6.20 (**Combinaison par majorité**). *Soit k un agent, $\Gamma \subseteq \{t, pa, ga, pi, gi, pe, ge\}$ un ensemble non vide de critères, $p_1 \in \mathcal{P}_k$ et $p_2 \in \mathcal{P}_k$ deux programmes. On définit $\preccurlyeq^{\Gamma-maj}$ sur $\mathcal{P}_k$ par $p_1 \preccurlyeq^{\Gamma-maj} p_2$ si et seulement si*

$$|\{\gamma \in \Gamma \mid p_1 \preccurlyeq^{\gamma} p_2\}|$$
$$\geq$$
$$|\{\gamma \in \Gamma \mid p_2 \preccurlyeq^{\gamma} p_1\}|.$$

Exemple 6.15. *Le critère portant sur la taille ne suffisant pas à déterminer un seul programme, le procureur décide de considérer deux critères supplémentaires. Voulant minimiser le risque que le juge rejette ses objections, le procureur se résout à considérer les critères portant sur la perte d'arguments (pa) et la perte d'interactions (pi). Et puisque la taille des programmes reste le critère le plus important pour lui, il arrête son choix sur une combinaison lexicographique (leximin).*

Ainsi, le procureur obtient :

$$\begin{aligned} & p_1 \prec^{(t,pa,pi)-lex} p_6 \prec^{(t,pa,pi)-lex} p_5 \prec^{(t,pa,pi)-lex} p_2 \\ \prec^{(t,pa,pi)-lex}\ & p_7 \prec^{(t,pa,pi)-lex} p_3 \prec^{(t,pa,pi)-lex} p_8 \prec^{(t,pa,pi)-lex} p_4 \end{aligned}$$

Le meilleur changement au regard de la suite de critères (t, pa, pi) combinés lexicographiquement est donc le programme p_1, et c'est donc ce programme qu'il va utiliser pour répondre à son adversaire.

6.3 Discussion

Synthèse

Dans ce chapitre, nous avons présenté un cadre théorique permettant de modéliser des opérations de changement de telle sorte qu'elles satisfont un ensemble de buts d'un agent sur un système d'argumentation cible à partir d'arguments ou d'attaques pris dans un système d'argumentation "source" (représentant les connaissances de cet agent). Nous avons également abordé la question du changement optimal, c'est-à-dire le fait de comparer plusieurs programmes (suites d'opérations) et de choisir celui qui optimise un, ou plusieurs, critères particuliers.

Parmi ces critères, le critère de distance structurelle (nombre de différences entre les graphes d'argumentation ; voir section 6.2.2 page 124) laisse entrevoir un lien avec la théorie de la mise à jour de croyances. En effet, la mise à jour de croyances concerne la recherche d'états du monde qui sont les plus proches d'un ensemble d'états initiaux et dans lesquels une nouvelle information est satisfaite. Cette proximité peut se calculer en termes de différences entre les états du monde. Il est alors naturel de faire un parallèle avec la recherche de graphes d'argumentation qui sont les plus proches d'un graphe initial en termes de distance structurelle, et dans lesquels des arguments particuliers sont acceptés.

De plus, le cadre que nous avons présenté suppose que l'agent voulant modifier le système cible connaît tous les arguments de ce dernier. Pour bien saisir cette hypothèse, reprenons notre exemple de l'audience de tribunal (section 3.1 page 30) où les deux orateurs tentent de persuader l'auditoire. Pour rappel, chacun d'entre eux est capable de construire le système d'argumentation correspondant au déroulement de l'audience et de calculer tous les arguments acceptés dans le cadre de cette audience. Et, pour arriver à leurs fins, c'est-à-dire faire accepter aux jurés leurs propres arguments, ils peuvent ajouter ou tenter de faire retirer un argument par le biais d'une objection.

L'hypothèse tient ici du fait que les participants sont capables de construire le système d'argumentation représentant le déroulement de l'audience. Intuitivement, ceci amène à supposer en filigrane qu'ils sont capables de connaître tous les arguments mis en jeu, et par conséquent ce que savent exactement les membres de l'auditoire, à la condition expresse, mais tacite, d'être présents durant l'intégralité de l'audience et que le système cible n'ait pas de "connaissances cachées" (des *a priori*, par exemple). Or, cette hypothèse, qui a du sens dans le cas d'une audience où toutes les personnes impliquées sont assidues et honnêtes, est très forte et n'est pas nécessairement vraie en général.

Supposons par exemple qu'un des orateurs ait dû s'absenter ou qu'il soit arrivé en cours d'audience. Il ne saurait alors plus tout à fait ce qui a été dit. Il lui manquerait peut-être une partie des arguments mis en jeu. Dans une telle éventualité, l'agent ne connaît pas exactement le système d'argumentation sur lequel il doit procéder à un changement, et ne se doute que de quelques informations à son propos. Le cadre que nous avons développé dans ce chapitre n'est alors plus adapté.

Travaux connexes

Un cadre similaire est utilisé par Bonzon et Maudet (2011) dans lequel chaque agent dispose de son propre système et peut modifier un système public commun (qui pourrait correspondre à la notion de cible dans notre cadre). Ici, il est supposé que les systèmes d'argumentation des agents contiennent tous les mêmes arguments et ne diffèrent que sur les attaques. D'autre part, ce travail consiste à fournir et à étudier un protocole visant à réglementer la mise en place d'attaques sur le système commun ; chaque agent a pour but l'acceptation, ou le rejet, par le système commun d'un argument particulier. Notre cadre a été proposé dans un objectif différent puisqu'il vise à permettre à un agent de raisonner sur les changements optimaux qu'il peut opérer sur un système cible. Par conséquent, dans notre cadre, les agents peuvent avoir des arguments différents et ont accès à la suppression d'arguments.

D'autres travaux étudient le problème de la satisfaction d'un but dans un système d'argumentation. En particulier, Booth *et al.* (2013) propose de représenter l'état de croyance d'un agent au moyen d'un système d'argumentation et d'une contrainte spécifiant la conclusion que devrait avoir ce système. Cet état mental peut subir des modifications à deux niveaux : au niveau de la contrainte, qui peut être renforcée, et au niveau du système, qui peut recevoir de nouveaux arguments et/ou attaques. La difficulté vient alors du fait qu'un changement peut rendre la contrainte et la conclusion du système incohérentes. Pour régler ce problème, les auteurs proposent deux solutions, la première étant qu'il est toujours possible d'ajouter un nombre arbitraire d'arguments pour restaurer la cohérence, la seconde consistant à ne garder qu'un sous-ensemble de croyances les plus rationnelles en fonction de la contrainte.

Notons par ailleurs que certains travaux, en particulier Zhuang (2013), s'intéressent au changement minimal à travers l'opération de suppression d'une attaque.

Remarquons en outre que le critère de la distance sémantique (nombre de différences entre les extensions des systèmes d'argumentation ; voir section 6.2.3 page 126) peut être rapproché du travail de Coste-Marquis *et al.* (2013). En partant de l'observation que le changement modifiant un système d'argumentation de façon à satisfaire un but est dépendant de la sémantique considérée, les auteurs étudient le problème du changement minimal en se basant principalement sur la distance entre les extensions du système, c'est-à-dire le nombre de différences entre les extensions avant et après le changement (ils considèrent en second lieu la distance entre les ensembles d'attaques avant et après le changement). Plus précisément, ils proposent, étant donné un système d'argumentation, une sémantique et une formule exprimant en quoi le statut des arguments doit être changé, une approche pour calculer les systèmes d'argumentations satisfaisant la formule tout en faisant en sorte que leurs extensions soient aussi proches que possible des extensions avant le changement. Notons qu'ils ne considèrent pas l'addition ou la suppression d'argument, et n'utilisent que des modifications au niveau de la relation d'attaque.

Ainsi, pour approfondir le lien entre le critère de distance structurelle et la théorie de la mise à jour et se défaire de notre hypothèse, nous considérons dans le chapitre suivant une approche axiomatique du changement en argumentation (chapitre 7 page 135), ce qui nous permet de rapprocher ce cadre du domaine du changement de croyances et d'étudier en quoi une opération de changement peut être assimilée à la mise à jour d'une base de croyances. Puis, dans le chapitre

8 page 155, nous présentons l'implémentation de ce cadre et son utilisation pour générer des opérations satisfaisant un but précisé en entrée.

Chapitre 7

Approche axiomatique

L'axiome doit être simple. Doit-il être clair ? Pas nécessairement. Simplicité et clarté, c'est deux.

Victor Hugo,
Proses philosophiques.

Sommaire

Nous avons vu, dans le chapitre précédent, un cadre permettant à un agent d'agir sur un système d'argumentation de façon à satisfaire son but.

D'un point de vue théorique, le fait de modifier un système d'argumentation pour s'assurer qu'un ensemble d'arguments est accepté (ce qui constitue un des buts possibles dans notre cadre), étant donné un ensemble de changements permis, est appelé *appartenance forcée* ("enforcement", voir Baumann et Brewka (2010)). Nous avons vu dans la section 6.2 page 121 que cette appartenance forcée peut être faite plus ou moins facilement, puisqu'elle peut mettre en jeu un ou plusieurs changements (les coûts pour ajouter et/ou supprimer des arguments peuvent d'ailleurs être pris en compte). Le but des orateurs est alors de trouver un "meilleur changement" à faire sur le système d'argumentation des jurés, ce que nous avons vu avec la notion de changement optimal dans le chapitre précédent.

Néanmoins, un tel changement est un cas particulier d'un opérateur d'appartenance forcée plus général pour lequel il est possible d'avoir des doutes sur le système initial à modifier. Dans ce cas plus général, il convient de s'assurer que le système d'argumentation après le changement satisfait un but donné quel que soit le système initial. L'appartenance forcée consiste donc en la recherche des systèmes d'argumentation qui sont les plus proches d'un système d'argumentation initial donné, et dans lesquels plusieurs arguments cibles sont acceptés.

Cette approche est similaire à la mise à jour de croyances. En effet, la mise à jour de croyances réside, intuitivement, en la recherche d'états du mondes qui sont les plus proches d'un ensemble d'états du monde initiaux (décrits par une formule initiale), et dans lesquels une nouvelle information est satisfaite.

L'idée majeure développée dans ce chapitre est donc un parallèle entre la théorie de la mise à jour de croyances (Winslett (1988); Katsuno et Mendelzon (1991)) et l'appartenance forcée en argumentation.

Ainsi, nous rappelons tout d'abord dans la section 7.1 en quoi consiste la théorie du changement de croyances de façon à éclaircir le parallèle entre la mise à jour de croyances et l'appartenance forcée. Il est important de noter que la mise à jour de croyances classique se place dans le cadre de la logique. C'est pourquoi nous nous proposons d'introduire un langage propositionnel dans lequel les informations relatives à l'appartenance forcée peuvent être exprimées (section 7.2 page 139). Ce langage nous permet de généraliser l'appartenance forcée avec une plus grande expressivité (section 7.3 page 142). Puis, en nous aidant du cadre présenté dans le chapitre 6 page 109, nous explicitons le parallèle avec la mise à jour de croyances dans la section 7.4 page 145. Enfin, la section 7.5 page 149 conclut ce chapitre tout en donnant quelques liens vers des travaux connexes existants.

Notons que ce chapitre généralise le travail présenté dans Bisquert *et al.* (2013c).

7.1 Rappels sur la théorie du changement de croyances

Dans le domaine de la théorie du changement de croyances, les travaux de Katsuno et Mendelzon (1991) ont formalisé la notion de "révision de croyances" ; le concept de révision d'un ensemble de formules avait été préalablement considéré par Alchourrón *et al.* (1985). La révision de croyances a pour but de définir comment intégrer une nouvelle information au sein d'un ensemble de croyances initiales. Les croyances sont représentées par des formules d'un langage formel. La révision consiste en l'addition d'informations tout en conservant la cohérence puisque l'incohérence mène à des informations inutilisables. Le principal intérêt des travaux d'Alchourrón *et al.* (1985) et Katsuno et Mendelzon (1991) est la définition d'un ensemble de postulats qui doivent être satisfaits par tout opérateur de révision "rationnel". Comme l'a constaté Léa Sombé (1994), ces postulats sont fondés sur trois principes :

- un principe de cohérence (le résultat doit être cohérent),
- un principe de changement minimal (les croyances doivent être modifiées le moins possible),
- un principe de priorité à la nouvelle information (la nouvelle information doit appartenir à l'ensemble résultant du processus de révision).

Une distinction très importante entre la révision de croyances et la mise à jour de croyances a été établie pour la première fois dans Winslett (1988). La différence se situe dans la nature de la nouvelle information : soit elle complète la connaissance du monde, soit elle informe qu'il y a un changement dans le monde. Plus précisément, la mise à jour est un processus qui prend en compte une évolution physique du système, alors que la révision est un processus qui prend en compte une évolution épistémique : c'est la connaissance à propos du monde qui évolue. Dans notre travail, nous faisons face à un problème de mise à jour, puisque dans l'appartenance forcée l'agent veut modifier un graphe dans le but de s'assurer que certains arguments seront acceptés (les graphes jouent le rôle des états du monde, comme expliqué dans la section 7.3 page 142). Une approche basée sur la révision s'appliquerait aux situations dans lesquelles l'agent apprend des informations à propos du système d'argumentation initial et veut corriger sa connaissance le concernant. Cela signifierait que le système d'argumentation n'a pas changé, mais les croyances de l'agent ont évolué.

Nous nous devons de procéder à quelques rappels sur la mise à jour de croyances. Un opérateur de mise à jour (Winslett (1988); Katsuno et Mendelzon (1991)) est une fonction associant à une base de connaissances φ, exprimée dans une logique propositionnelle $\mathscr{L}$ et représentant une certaine connaissance à propos d'un système dans un état initial, et une nouvelle information $\alpha \in \mathscr{L}$, une nouvelle base de connaissances $\varphi \diamond \alpha \in \mathscr{L}$ représentant le système après cette évolution.

Dans la mise à jour de croyances, l'information α doit être interprétée comme la projection des effets attendus d'un “changement explicite”, ou plus précisément, les effets attendus de l'action “rendre α vraie”. La propriété clef de la mise à jour de croyances est le postulat **U8** de Katsuno et Mendelzon (Katsuno et Mendelzon (1991)) statuant que chaque modèle de φ est mis à jour indépendamment (contrairement à la révision de croyances). Nous rappelons ici les postulats[1] de Katsuno et Mendelzon, où $\mathscr{L}$ dénote un langage propositionnel et $[\varphi]$ dénote l'ensemble des modèles de la formule φ : $\forall \varphi, \psi, \alpha, \beta \in \mathscr{L}$,

U1 : $\varphi \diamond \alpha \models \alpha$.

U2 : $\varphi \models \alpha \implies [\varphi \diamond \alpha] = [\varphi]$.

U3 : $[\varphi] \neq \varnothing$ et $[\alpha] \neq \varnothing \implies [\varphi \diamond \alpha] \neq \varnothing$.

U4 : $[\varphi] = [\psi]$ et $[\alpha] = [\beta] \implies [\varphi \diamond \alpha] = [\psi \diamond \beta]$.

U5 : $(\varphi \diamond \alpha) \wedge \beta \models \varphi \diamond (\alpha \wedge \beta)$.

U8 : $[(\varphi \vee \psi) \diamond \alpha] = [(\varphi \diamond \alpha) \vee (\psi \diamond \alpha)]$.

U9 : si $|[\varphi]| = 1$, alors $[(\varphi \diamond \alpha) \wedge \beta] \neq \varnothing \implies \varphi \diamond (\alpha \wedge \beta) \models (\varphi \diamond \alpha) \wedge \beta$.

La signification intuitive de ces postulats sera évoquée au moment de leur utilisation pour l'appartenance forcée.

Ces postulats permettent à Katsuno et Mendelzon d'écrire le théorème de représentation suivant concernant la mise à jour, à savoir un opérateur satisfaisant

1. Les postulats **U6** et **U7** ne sont pas considérés ici puisque l'ensemble **U1**-**U8** est seulement relié à une famille de préordres partiels, alors que remplacer **U6**-**U7** par **U9** assure une famille de préordres complets.

ces postulats peut être défini au moyen d'une relation de préférence ternaire sur les états du monde (l'ensemble de tous les états est noté Ω).

Théorème 7.1 (Katsuno et Mendelzon (1991)). *Il existe un opérateur $\diamond : \mathscr{L} \times \mathscr{L} \to \mathscr{L}$ satisfaisant **U1**, **U2**, **U3**, **U4**, **U5**, **U8**, **U9** si et seulement s'il existe une assignation* fidèle *qui à chaque $\omega \in \Omega$ associe un préordre complet, noté $\preccurlyeq_\omega$ tel que $\forall \varphi, \alpha \in \mathscr{L}$, $[\varphi \diamond \alpha] = \bigcup_{\omega \in [\varphi]} \{\omega' \in [\alpha]$ tel que $\forall \omega'' \in [\alpha], \omega' \preccurlyeq_\omega \omega''\}$, où une assignation d'un préordre*[2] *$\preccurlyeq_\omega$ à chaque $\omega \in \Omega$ est* fidèle *si et seulement si $\forall \omega, \omega' \in \Omega$, $\omega \prec_\omega \omega'$.*

Cet ensemble de postulats a déjà été largement étudié dans la littérature (le lecteur curieux pourra se pencher, par exemple, sur Herzig et Rifi (1999); Herzig (2005); Dubois *et al.* (1995)). **U2**, par exemple, impose l'inertie, ce qui n'est pas toujours approprié. Certaines approches ont complété ces postulats : Herzig (2005) a proposé de restreindre les mises à jour possibles en prenant en compte des contraintes d'intégrité, c'est-à-dire des formules qui doivent être satisfaites avant et après la mise à jour. Dubois *et al.* (1995) proposent de ne pas imposer l'inertie et d'autoriser l'échec de mise à jour, même dans le cas où les formules sont cohérentes. La solution consiste à introduire un monde impossible à atteindre nommé z de façon à disposer d'une borne supérieure de la proximité entre un monde courant et un monde inatteignable.

Dans la suite de ce travail, comme nous le verrons dans la section 7.3 page 142, nous souhaitons également restreindre les changements possibles. Ainsi, nous devons permettre l'échec d'une appartenance forcée. Nous faisons donc le choix d'introduire un ensemble de transitions permises $\mathcal{T}$ qui limite les appartenances forcées possibles. La conséquence directe est la nécessité d'adapter les postulats de la mise à jour de façon à restreindre les transitions possibles, ce qui sera fait dans la section 7.4 page 145.

Le rappel sur les postulats nous permet de mieux apprécier le parallèle entre appartenance forcée et mise à jour de croyances. Ainsi, les mondes correspondent aux systèmes d'argumentation, tandis que les formules représentent des connaissances à propos de ces systèmes d'argumentation. En appartenance forcée classique, cette connaissance est exprimée grâce à la description d'un système d'argumentation initial et à un ensemble d'arguments que l'on souhaite voir acceptés. La table 7.1 page suivante exprime ce parallèle entre mise à jour de croyances et appartenance forcée.

Ainsi, de manière à mettre en œuvre le parallèle et nous permettre d'étudier les travaux de la mise à jour de croyances dans le cadre de l'appartenance forcée, la première étape est d'introduire un langage logique pour représenter les graphes d'argumentation, ce que nous allons faire dans la section suivante.

2. Dans la suite de ce travail, $\prec_\omega$ est classiquement défini à partir de $\preccurlyeq_\omega$ par : $a \prec_\omega b$ si et seulement si $a \preccurlyeq_\omega b$ et $b \not\preccurlyeq_\omega a$ (voir annexe B page 195).

	Mise à jour	Appartenance forcée
Connaissance initiale	Ensemble de mondes	Ensemble de systèmes d'argumentation
Entrée	Nouvelle information	But
Contraintes	Aucune (toute mise à jour est réalisable)	Ensemble de transitions

Table 7.1 – *Résumé du parallèle existant entre mise à jour de croyances et appartenance forcée.*

7.2 Un langage logique pour représenter les graphes d'argumentation

Nous l'avons vu dans le chapitre 2 page 11, il est possible de déterminer le statut des arguments d'un système d'argumentation grâce aux sémantiques et aux extensions qui en découlent. Dans un souci de généralité, nous ne sommes intéressés ici ni par une sémantique particulière, ni par le mécanisme utilisé pour calculer le statut des arguments. Nous considérons seulement la fonction f_{acc} suivante :

Définition 7.1 (**Arguments acceptés dans un graphe**). *Étant donné un univers de référence* $\mathbf{U} = (\mathcal{A}_\mathbf{U}, \mathcal{R}_\mathbf{U})$*, soit* f_{acc} *une fonction telle que*

$$\begin{aligned} f_{acc} : \quad & \mathbf{G_U} \to 2^{\mathcal{A}_\mathbf{U}} \\ & \mathcal{G} = (\mathcal{A}, \mathcal{R}) \mapsto \mathcal{S} \subseteq \mathcal{A}. \end{aligned}$$ [3]

Cette dernière associe à tout graphe d'argumentation $\mathcal{G}$ l'ensemble $\mathcal{S}$ des arguments qui ont le statut accepté dans $\mathcal{G}$ selon une sémantique et un calcul de statut donnés. [4]

Nous allons définir un langage propositionnel $\mathscr{L}$ permettant de décrire un système d'argumentation et son ensemble d'arguments acceptés. Sa sémantique sera définie en fonction de $\mathbf{G_U}$.

Notation 7.1. $\forall \varphi \in \mathscr{L}$*, nous notons par* $[\varphi]$ *l'ensemble de graphes d'argumentation tel que* φ *est vraie (au sens de la définition* *7.2 page suivante**) dans ces graphes, à savoir* $[\varphi] = \{\mathcal{G} \in \mathbf{G_U}$ *tel que* φ *est vraie dans* $\mathcal{G}\}$.

Classiquement, nous notons $\mathcal{G} \models \varphi$ si et seulement si $\mathcal{G} \in [\varphi]$ et $\varphi \models \psi$ si et seulement si $[\varphi] \subseteq [\psi]$.

Dans tous nos exemples, nous utilisons un langage propositionnel $\mathscr{L}_\mathbf{U}$ capable d'exprimer des conditions sur :

3. Notons que l'inclusion dans $\mathcal{A}$ permet de s'assurer que l'ensemble d'arguments renvoyé par la fonction f_{acc} fait partie du graphe sur lequel elle est appliquée. Cette contrainte n'est pas nécessaire dans le cas des sémantiques classiques telles que les sémantiques préférée, basique et stable.

4. Cette fonction pourrait être paramétrée par la sémantique utilisée.

- la présence ou l'absence d'un argument dans un graphe avec le prédicat "arg" (signifiant "appartient à $\mathcal{G}$"),
- le statut accepté ou rejeté d'un argument dans un graphe avec le prédicat "acc" (signifiant "est accepté dans $\mathcal{G}$'),
- la présence ou l'absence d'une attaque entre deux arguments dans un graphe avec le prédicat "att" (signifiant "ont une interaction dans $\mathcal{G}$").

Ainsi, bien que nous nous concentrions sur l'addition et la suppression d'un argument, ce langage permet de traiter les quatre opérations élémentaires de la définition 6.3 page 115.

Définition 7.2 (**Langage propositionnel de description d'un système d'argumentation**). *Soit $\mathbf{G_U}$ l'ensemble des graphes d'argumentation qui peuvent être construits sur $\mathcal{A}_\mathbf{U}$ au moyen de la définition 6.2 page 112. Soit $\mathscr{L}_\mathbf{U}$ le langage propositionnel associé avec le vocabulaire $\{acc(x), arg(x), att(x, y) \mid x, y \in \mathcal{A}_\mathbf{U}, (x, y) \in \mathcal{R}_\mathbf{U}\}$, avec les connecteurs logiques classiques $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ et les constantes $\bot$ et $\top$. Sa sémantique est définie par rapport à $\mathbf{G_U}$ de la façon suivante : soit $\mathcal{G} = (\mathcal{A}, \mathcal{R}) \in \mathbf{G_U}$,*

- *la formule $\bot$ est toujours fausse dans $\mathcal{G}$,*
- *la formule $\top$ est toujours vraie dans $\mathcal{G}$,*
- *si $x \in \mathcal{A}_\mathbf{U}$, alors*
  - *— la formule $acc(x)$ est vraie dans $\mathcal{G}$ si et seulement si $x \in f_{acc}(\mathcal{G})$,*
  - *— la formule $arg(x)$ est vraie dans $\mathcal{G}$ si et seulement si $x \in \mathcal{A}$,*
- *si $(x, y) \in \mathcal{R}_\mathbf{U}$, alors la formule $att(x, y)$ est vraie dans $\mathcal{G}$ si et seulement si $(x, y) \in \mathcal{R}$,*
- *les formules non atomiques sont interprétées de façon classique, $\neg\varphi$ est vraie dans $\mathcal{G}$ si φ n'est pas vraie dans $\mathcal{G}$, $\varphi_1 \vee \varphi_2$ est vraie dans $\mathcal{G}$ si φ_1 ou φ_2 sont vraies dans $\mathcal{G}$, etc.*

Notons que tout argument accepté dans un graphe d'argumentation doit appartenir à ce graphe.

Proposition 7.1. *Dans $\mathscr{L}_\mathbf{U}$, $\forall \mathcal{G} \in \mathbf{G_U}, \forall x \in \mathcal{A}_\mathbf{U}, \mathcal{G} \models acc(x) \rightarrow arg(x)$.*

De plus, une attaque présente dans un graphe ne peut concerner que des arguments présents dans ce même graphe.

Proposition 7.2. *Dans $\mathscr{L}_\mathbf{U}$, $\forall \mathcal{G} \in \mathbf{G_U}, \forall x, y \in \mathcal{A}_\mathbf{U}, \mathcal{G} \models att(x, y) \rightarrow (arg(x) \wedge arg(y))$.*

Les démonstrations de ces deux propositions sont données dans l'annexe C.3 page 211.

Nous avons maintenant besoin d'une fonction permettant de décrire un système d'argumentation dans le langage $\mathscr{L}_\mathbf{U}$.

Définition 7.3 (Fonction caractéristique associée au langage propositionnel $\mathscr{L}_{\mathbf{U}}$). *La* fonction caractéristique $f_{\mathbf{U}}$ *associée à* $\mathscr{L}_{\mathbf{U}}$ *est définie par :*

$$\begin{array}{llll} f_{\mathbf{U}}: & \mathbf{G}_{\mathbf{U}} & \to & \mathscr{L}_{\mathbf{U}} \\ & \mathcal{G} & \mapsto & \bigwedge_{x \in \mathcal{A}} arg(x) \quad \wedge \quad \bigwedge_{x \in \mathcal{A}_{\mathbf{U}} \setminus \mathcal{A}} \neg arg(x) \quad \wedge \\ & & & \bigwedge_{(x,y) \in \mathcal{R}} att(x,y) \quad \wedge \quad \bigwedge_{(x,y) \in \mathcal{R}_{\mathbf{U}} \setminus \mathcal{R}} \neg att(x,y). \end{array}$$

Notons que, dans la définition 7.2 page précédente, si l'ensemble des arguments et l'ensemble des interactions appartenant à $\mathcal{G}$ sont connus, alors $\mathcal{G}$ est parfaitement connu. Plus formellement, $f_{\mathbf{U}}(\mathcal{G})$ caractérise $\mathcal{G}$ de façon unique :

Proposition 7.3. $\forall \mathcal{G} \in \mathbf{G}_{\mathbf{U}}$, $[f_{\mathbf{U}}(\mathcal{G})] = \{\mathcal{G}\}$.

Exemple 7.1. *Le procureur a réussi une manœuvre délicate en convainquant les jurés que l'argument x_0 représente seulement la présomption d'innocence de l'accusé (supprimant ainsi l'attaque de x_0 vers x_1), ce qui lui donne enfin la possibilité de faire accepter sceptiquement x_1. Mais lors de l'ajournement de l'audience, le procureur s'est aperçu que la presse avait publié dans son édition du matin des informations retentissantes à propos de l'affaire à laquelle il participe. Selon elle, l'accusé, homme vénal par excellence, avait tout intérêt à faire disparaître sa chère et tendre ; une histoire classiquement sordide d'héritage (argument x_7 de la table 3.1 page 32). Cette aide littéraire inattendue pourrait lui permettre de convaincre les jurés ! Mais l'article prouve également que le meurtre n'a pu qu'être commis sous le coup de la rage. C'est bien sa veine ! Il avait justement réussi à présenter son accusation sous l'angle du meurtre différé (et prémédité depuis longtemps), incapacitant ainsi l'argument de l'alibi (x_2) à attaquer l'argument de la culpabilité de l'accusé (x_1).*

Le procureur se demande si les jurés ont lu leurs feuilles de chou, ce qui changerait radicalement la tournure de l'audience. Il n'est maintenant plus sûr de ce qu'ils savent ; leur système d'argumentation n'est donc plus complètement connu par ce dernier. Le système est représenté dans $\mathscr{L}_{\mathbf{U}}$ (où **U** *est l'univers présenté dans la figure 6.1 page 112) par la formule $\varphi_{jurés}$:*

$$\begin{array}{llllllll} \varphi_{jurés} = & & & & & & & \\ arg(x_0) & \wedge & arg(x_1) & \wedge & arg(x_2) & \wedge & arg(x_4) & \wedge \\ \neg arg(x_3) & \wedge & \neg arg(x_5) & \wedge & \neg arg(x_6) & \wedge & & \\ att(x_1, x_0) & \wedge & att(x_4, x_1) & \wedge & & & & \\ \neg att(x_0, x_1) & \wedge & \neg att(x_3, x_2) & \wedge & \neg att(x_5, x_4) & \wedge & \neg att(x_6, x_1) & \wedge \\ ((arg(x_7) & \wedge & att(x_7, x_4) & \wedge & att(x_2, x_1)) & \vee & & \\ (\neg arg(x_7) & \wedge & \neg att(x_7, x_4) & \wedge & \neg att(x_2, x_1))). & & & \end{array}$$

Cette formule couvre les deux graphes donnés dans la figure 7.1 ; la disjonction entre $arg(x_7) \wedge att(x_7, x_4) \wedge att(x_2, x_1)$ et $\neg arg(x_7) \wedge \neg att(x_7, x_4) \wedge \neg att(x_2, x_1)$ exprime le fait que le procureur hésite entre la situation (a), où les jurés ont lu les journaux, et la situation (b), où les jurés sont restés dans l'optique présentée par le procureur.

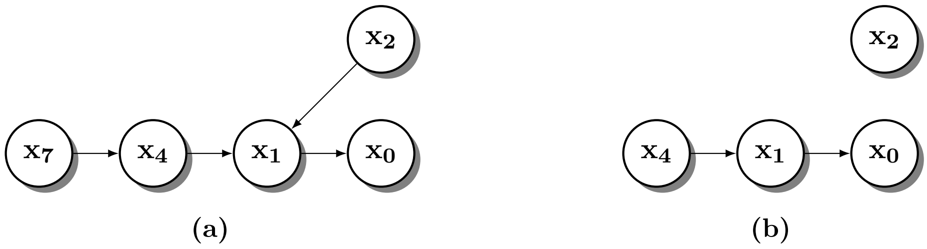


Figure 7.1 – *Illustration des deux graphes d'argumentation possibles couverts par la formule $\varphi_{jurés}$.*

De plus, x_0, x_2 et (x_4 ou x_7) sont les seuls membres de l'extension basique (voir le chapitre 2 page 11). Ainsi, $\varphi_{jurés} \models acc(x_0) \wedge acc(x_2) \wedge (acc(x_4) \vee acc(x_7))$.

7.3 Appartenance forcée généralisée

Formalisons l'appartenance forcée classique en utilisant les définitions présentées dans le chapitre 6 page 109. Soit $\mathcal{G} \in \mathbf{G_U}$ et $\mathcal{S} \subseteq \mathcal{A}_\mathbf{U}$. Forcer l'appartenance de $\mathcal{S}$ dans $\mathcal{G}$ produit un graphe $\mathcal{G}' \in \mathbf{G_U}$ obtenu à partir de $\mathcal{G}$ en appliquant des opérations de changement de telle sorte que $\mathcal{S} \subseteq f_{acc}(\mathcal{G}')$. Plusieurs appartenances forcées de $\mathcal{S}$ dans $\mathcal{G}$ peuvent être comparées en utilisant un préordre $\preccurlyeq_\mathcal{G}$. Par exemple, il semble naturel de chercher des appartenances forcées réalisant un changement minimal sur $\mathcal{G}$. La minimalité peut par exemple être basée sur une distance entre graphes (nous en avons vu certaines dans la section 6.2 page 121). Dans ce cas, étant donné deux appartenances forcées produisant $\mathcal{G}'$ et $\mathcal{G}''$ de $\mathcal{S}$ dans $\mathcal{G}$, $\mathcal{G}' \preccurlyeq_\mathcal{G} \mathcal{G}''$ peut être défini par $distance(\mathcal{G}, \mathcal{G}') \leq distance(\mathcal{G}, \mathcal{G}'')$.

Un tel préordre suggère un parallèle entre le problème de l'appartenance forcée et un problème de mise à jour. En effet, l'avons vu dans la section 7.1 page 136, la mise à jour exploite également des préordres sur les mondes. Plus précisément, un opérateur de mise à jour associe à une base de connaissances et une information une nouvelle base de connaissances, où les bases de connaissances sont exprimées grâce à des formules propositionnelles. Le pendant sémantique de ce lien est défini par des opérations sur des modèles de formules, c'est-à-dire des mondes. Ceci donne naissance à l'idée que les graphes d'argumentation sont aux mondes ce que les formules caractérisant des ensembles de graphes sont aux formules caractérisant des ensembles de mondes.

La définition 7.2 page 140 nous permet de continuer ce parallèle. Soit $\mathcal{S} \subseteq \mathcal{A}_\mathbf{U}$ et $\alpha = \bigwedge_{x \in \mathcal{S}} acc(x)$. $[\alpha]$ peut être considéré comme l'ensemble de graphes dans lesquels les éléments de $\mathcal{S}$ sont acceptés. En d'autres termes, $[\alpha]$ joue le rôle de l'ensemble des graphes qui acceptent $\mathcal{S}$.

Ceci mène à la formalisation du problème de l'appartenance forcée par un opérateur s'appliquant à des formules propositionnelles, avec un pendant sémantique fonctionnant avec des graphes d'argumentation. Ainsi, l'appartenance forcée appliquée à des formules propositionnelles φ et α signifie forcer α à être satisfaite à partir de graphes qui satisfont φ.

Cette formalisation nous permet de proposer deux généralisations de l'appartenance forcée : tout d'abord, il est désormais possible d'utiliser l'appartenance forcée pour non seulement imposer qu'un ensemble d'arguments soit accepté, mais aussi faire en sorte que l'appartenance forcée puisse concerner n'importe quel autre but pouvant être exprimé dans un langage propositionnel décrivant des graphes (présence/absence d'arguments ou d'attaques, acceptation/rejet d'arguments, etc.). Deuxièmement, il n'est pas nécessaire que le graphe initial soit complètement connu puisque une description dans un langage propositionnel permet une expressivité plus riche. Ainsi, un ensemble de graphes sera considéré comme représentant l'état initial du système d'argumentation.

Néanmoins, les buts relatifs (voir la section 6.1.3 page 119) sont difficilement exprimables dans un contexte de mise à jour ; aussi, dans la suite de ce chapitre, nous ne considèrerons que les buts absolus. Ainsi, en associant une formule propositionnelle à un but absolu, ce but est réalisé par un graphe si la formule associée est satisfaite dans ce graphe.

Exemple 7.2. *Nous savons que le procureur veut forcer l'appartenance de l'ensemble* $\{x_1\}$*. Ce but peut être exprimé dans* $\mathscr{L}_{\mathbf{U}}$ *par la formule* $acc(x_1)$*. De façon à forcer l'appartenance de l'argument* x_1 *dans le graphe des jurés, le procureur peut utiliser le programme* $\langle(\bigoplus, \{x_3\}, \{(x_3, x_2)\}), (\bigoplus, \{x_5\}, \{(x_5, x_4)\})\rangle$ *qui a l'impact montré dans la figure 7.3. Un but plus complexe aurait pu être, par exemple,* $\neg acc(x_4) \vee acc(x_0)$*.*

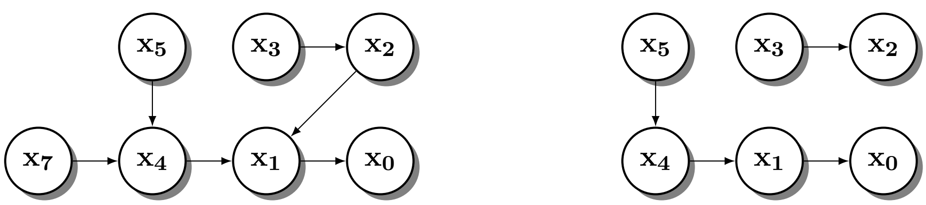


Figure 7.3 – *Impact du programme du procureur sur les possibles systèmes des jurés*

Nous sommes maintenant capables de définir formellement l'appartenance forcée généralisée. Celle-ci est basée sur un langage propositionnel $\mathscr{L}$, dit *bien formé sur* $\mathbf{G_U}$, dans lequel, pour tout graphe $\mathcal{G} \in \mathbf{G_U}$, il existe une expression de $\mathscr{L}$ qui le décrit spécifiquement (c'est-à-dire, dont l'interprétation est réduite à $\mathcal{G}$).

Définition 7.4 (**Langage bien formé sur l'ensemble des graphes d'argumentation $\mathbf{G_U}$**)**.** *Soit* $\mathbf{G_U}$ *l'ensemble des graphes d'argumentation qui peuvent être construits sur* $\mathcal{A}_\mathbf{U}$ *au moyen de la définition* *6.2 page 112**. Un langage propositionnel* $\mathscr{L}$ *est* bien formé sur $\mathbf{G_U}$ *si et seulement s'il existe une fonction* f *telle que* $\forall \mathcal{G} \in \mathbf{G_U}$*,* $f(\mathcal{G}) \in \mathscr{L}$ *et* $[f(\mathcal{G})] = \{\mathcal{G}\}$ *;* f *est appelée* fonction caractéristique associée à $\mathscr{L}$*.*

Exemple 7.3. *Le langage* $\mathscr{L}_\mathbf{U}$ *de la définition* *7.2 page 140* *est bien formé sur* $\mathbf{G_U}$*. Néanmoins,* $\mathscr{L}_\mathbf{U}$ *ne permet pas d'exprimer tous les buts envisagés dans la section* *6.1.3 page 119**, notamment les conditions à propos de la cardinalité de chaque extension après l'appartenance forcée ; en effet,* $\mathscr{L}_\mathbf{U}$ *est un simple exemple introduit dans un but illustratif.*

Les résultats que nous avançons sont valables pour tout langage propositionnel $\mathscr{L}$ bien formé sur $\mathbf{G_U}$.

De façon à capturer l'appartenance forcée classique, nous devons également être capable de restreindre les changements que les graphes peuvent subir. Cela est fait par l'introduction d'un ensemble $\mathcal{T} \subseteq \mathbf{G_U} \times \mathbf{G_U}$ de transitions permises entre graphes.

Voici trois exemples d'ensembles de transitions permises qui peuvent être utilisés :

- Si les changements permis sont des opérations exécutables par un agent k, alors $\mathcal{T}_1 = \{(\mathcal{G}, \mathcal{G}') \in \mathbf{G_U} \times \mathbf{G_U}$, $\exists o$ telle que o est une opération exécutable par k sur $\mathcal{G}$ tel que $o(\mathcal{G}) = \mathcal{G}'\}$.
- Si les changements permis sont des programmes exécutables par un agent k, alors $\mathcal{T}_2 = \{(\mathcal{G}, \mathcal{G}') \in \mathbf{G_U} \times \mathbf{G_U}$, $\exists p$ tel que p est un programme exécutable par k sur $\mathcal{G}$ tel que $p(\mathcal{G}) = \mathcal{G}'\}$.
- L'expansion normale de Baumann (Baumann et Brewka (2010); Baumann (2012b)) peut être traduite en termes de transitions permises de la façon suivante : $\mathcal{T}_B = \{(\mathcal{G}, \mathcal{G}') \in \mathbf{G_U} \times \mathbf{G_U}$, avec $\mathcal{G} = (\mathcal{A}, \mathcal{R})$ et $\mathcal{G}' = (\mathcal{A} \cup \mathcal{A}_1, \mathcal{R} \cup \mathcal{R}_1)$, où $\mathcal{A}_1 \neq \varnothing$, $\mathcal{A} \cap \mathcal{A}_1 = \varnothing$ et $\mathcal{R}_1 \subseteq (\mathcal{A} \times \mathcal{A}_1) \cup (\mathcal{A}_1 \times \mathcal{A}) \cup (\mathcal{A}_1 \times \mathcal{A}_1)\}$. Cela signifie que les transitions admises par l'expansion normale de Baumann sont restreintes à l'addition d'un nouvel ensemble d'arguments ayant éventuellement des attaques "internes" et des attaques de/vers l'ensemble initial (la seule restriction est donc que l'on ne peut pas ajouter d'attaques entre les arguments initiaux).

Nous pouvons maintenant définir formellement un opérateur d'appartenance forcée généralisée :

Définition 7.5 (**Opérateur d'appartenance forcée généralisée**)**.** *Soit* $\mathbf{G_U}$ *l'ensemble des graphes d'argumentation qui peuvent être construits sur* $\mathcal{A}_\mathbf{U}$ *au moyen de la définition* *6.2 page 112* *et* $\mathscr{L}$ *un langage propositionnel bien formé sur* $\mathbf{G_U}$*. Un* opérateur d'appartenance forcée généralisée $\diamondsuit_{\mathcal{T}}$ relatif à un ensemble de transitions permises $\mathcal{T} \subseteq \mathbf{G_U} \times \mathbf{G_U}$ *est une fonction de* $\mathscr{L} \times \mathscr{L} \to \mathscr{L}$ *qui associe*

- *à toute formule* $\varphi \in \mathscr{L}$ *donnant des informations à propos d'un système d'argumentation cible*
- *et toute formule* $\alpha \in \mathscr{L}$ *encodant un but,*

une formule $\varphi \diamondsuit_{\mathcal{T}} \alpha \in \mathscr{L}$ *caractérisant le système d'argumentation dans lequel* α *est satisfaite, et qui peut être obtenue par un changement appartenant à* $\mathcal{T}$*.*

Exemple 7.2 (cont.). *Le procureur veut faire en sorte que l'argument* x_1 *soit accepté (dans le cadre de la sémantique basique) en utilisant un programme exécutable, les arguments* x_2 *et* x_4 *étant présents. Il peut exploiter le résultat suivant :* $\varphi_{jurés} \diamondsuit_{\mathcal{T}_p^{proc}} (acc(x_1) \wedge arg(x_2) \wedge arg(x_4)) \models arg(x_3) \wedge att(x_3, x_2) \wedge arg(x_5) \wedge att(x_5, x_4)$*.*

Notation 7.2. $\forall \varphi, \psi \in \mathscr{L}$*, une transition dans* $\mathcal{T}$ *est possible entre un ensemble de graphes satisfaisant* φ *et un ensemble de graphes satisfaisant* ψ*, noté* $(\varphi, \psi) \models \mathcal{T}$*, si et seulement si* $([\varphi] \neq \varnothing$ *et* $\forall \mathcal{G} \in [\varphi]$*,* $\exists \mathcal{G}' \in [\psi]$*,* $(\mathcal{G}, \mathcal{G}') \in \mathcal{T})$*.*

En d'autres termes, une transition d'un ensemble donné de graphes vers un autre ensemble est possible si et seulement s'il existe une transition possible de *chaque graphe* du premier ensemble (qui ne doit pas être vide) vers au moins un graphe du deuxième ensemble.

7.4 Postulats de l'appartenance forcée généralisée

Dans cette partie, nous allons définir un ensemble de postulats rationnels pour $\diamondsuit_{\mathcal{T}}$. Les postulats décrits dans cette section sont définis $\forall \varphi, \psi, \alpha, \beta \in \mathscr{L}$. Ces postulats sont des contraintes qui visent à traduire l'idée de l'appartenance forcée. Certains postulats venant de la mise à jour sont appropriés, notamment **U1** puisqu'il assure que les contraintes imposées par α sont vraies après l'appartenance forcée. Le postulat **U2** est optionnel ; il impose en effet que si α est déjà satisfaite dans un graphe, alors forcer l'appartenance de α signifie qu'aucun changement n'est apporté. Ce postulat impose l'inertie en tant que changement préféré, ce qui peut ne pas être désirable dans toutes les situations. **U3**, transposé en termes de graphes, impose que si une formule est satisfaite pour certains graphes et si l'information de la mise à jour est également satisfaite pour certains graphes, alors le résultat de l'appartenance forcée doit fournir un ensemble non vide de graphes. Ici, par contre, non ne souhaitons pas imposer que toute appartenance forcée soit toujours possible, puisque certains graphes peuvent être inatteignables à partir d'autres. Aussi, nous proposons de remplacer **U3** par un postulat nommé **E3** et basé sur l'ensemble de transitions permises $\mathcal{T}$.

E3 : $[\varphi \diamondsuit_{\mathcal{T}} \alpha] \neq \varnothing$ ssi $(\varphi, \alpha) \models \mathcal{T}$.

En raison de la définition de $(\varphi, \alpha) \models \mathcal{T}$, **E3** gère deux cas d'impossibilité en appartenance forcée : pas de transition possible de φ à α et l'inexistence de graphe satisfaisant φ ou α (comme nous allons le voir dans la proposition 7.6 page ci-contre).

U4 est approprié pour notre formalisation puisque les opérateurs d'appartenance forcée sont définis sémantiquement. **U5** est également approprié pour l'appartenance forcée puisqu'il statue que les graphes où l'on force l'appartenance de α et dans lesquels β est déjà satisfaite sont des graphes où les contraintes α et β sont forcées. En raison de notre volonté de permettre les échecs d'appartenance forcée, ce postulat a été restreint aux formules "complètes"[5] (voir démonstration du théorème 7.2 dans l'annexe C.3 page 211).

E5 : si $|[\varphi]| = 1$, alors $(\varphi \diamondsuit_{\mathcal{T}} \alpha) \wedge \beta \models \varphi \diamondsuit_{\mathcal{T}} (\alpha \wedge \beta)$.

U8 capture la décomposabilité de l'appartenance forcée par rapport à un ensemble de graphes d'argumentation initiaux. Nous modifions légèrement ce postulat de façon à prendre en compte la possibilité d'échec, c'est-à-dire si le fait de forcer l'appartenance de quelque chose est impossible, alors forcer son appartenance dans un plus grand ensemble de graphes est également impossible, sinon l'appartenance forcée peut être décomposable :

E8 : si $([\varphi] \neq \varnothing$ et $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing)$ ou $([\psi] \neq \varnothing$ et $[\psi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing)$,
alors $[(\varphi \vee \psi) \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$
sinon $[(\varphi \vee \psi) \diamondsuit_{\mathcal{T}} \alpha] = [(\varphi \diamondsuit_{\mathcal{T}} \alpha) \vee (\psi \diamondsuit_{\mathcal{T}} \alpha)]$.

Le postulat **U9** est en quelque sorte la réciproque du postulat **E5** **U9** est restreint à une formule "complète" dans l'article de Katsuno et Mendelzon ; cette restriction est requise, avec celle de **E5**, pour la démonstration du théorème 7.2 page ci-contre que nous verrons plus loin.

Notons que la présence de **U1** dans l'ensemble de postulats caractérisant un opérateur d'appartenance forcée n'est pas nécessaire puisqu'il peut être dérivé à partir **E3**, **E5** et **E8**.

Proposition 7.4. ***U1*** *découle de* ***E3****,* ***E5*** *et* ***E8****.*

Ainsi, nous avons maintenant un nouvel ensemble de postulats pour l'appartenance forcée :

E3 : $[\varphi \diamondsuit_{\mathcal{T}} \alpha] \neq \varnothing$ si et seulement si $(\varphi, \alpha) \models \mathcal{T}$.

U4 : $[\varphi] = [\psi]$ et $[\alpha] = [\beta] \implies [\varphi \diamond \alpha] = [\psi \diamond \beta]$.

E5 : si $|[\varphi]| = 1$, alors $(\varphi \diamondsuit_{\mathcal{T}} \alpha) \wedge \beta \models \varphi \diamondsuit_{\mathcal{T}} (\alpha \wedge \beta)$.

E8 : si $([\varphi] \neq \varnothing$ et $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing)$ ou $([\psi] \neq \varnothing$ et $[\psi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing)$,
alors $[(\varphi \vee \psi) \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$
sinon $[(\varphi \vee \psi) \diamondsuit_{\mathcal{T}} \alpha] = [(\varphi \diamondsuit_{\mathcal{T}} \alpha) \vee (\psi \diamondsuit_{\mathcal{T}} \alpha)]$.

5. Notons que $|[\varphi]| = 1$ si et seulement si $\exists \mathcal{G} \in \mathbf{G_U}$ tel que $[\varphi] = [f(\mathcal{G})]$.

U9 : si $|[\varphi]| = 1$, alors $[(\varphi \diamond \alpha) \wedge \beta] \neq \varnothing \implies \varphi \diamond (\alpha \wedge \beta) \models (\varphi \diamond \alpha) \wedge \beta$.

Ces postulats nous permettent d'écrire le théorème de représentation suivant concernant l'appartenance forcée, à savoir un opérateur d'appartenance forcée satisfaisant ces postulats peut être défini au moyen d'une famille de préordres sur les graphes.

Définition 7.6 (**Assignation de préordres respectant un ensemble de transitions**)**.** *Étant donné un ensemble $\mathcal{T} \subseteq \mathbf{G_U} \times \mathbf{G_U}$ de transitions permises, une* assignation respectant $\mathcal{T}$ *est une fonction qui à chaque $\mathcal{G} \in \mathbf{G_U}$ associe un préordre complet $\preccurlyeq_{\mathcal{G}}$ tel que $\forall \mathcal{G}_1, \mathcal{G}_2 \in \mathbf{G_U}$, si $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et $(\mathcal{G}, \mathcal{G}_2) \notin \mathcal{T}$, alors $\mathcal{G}_2 \not\preccurlyeq_{\mathcal{G}} \mathcal{G}_1$.*

Théorème 7.2. *Étant donné un ensemble $\mathcal{T} \subseteq \mathbf{G_U} \times \mathbf{G_U}$ de transitions permises, il existe un opérateur $\diamondsuit_{\mathcal{T}} : \mathscr{L} \times \mathscr{L} \to \mathscr{L}$ satisfaisant **E3**, **U4**, **E5**, **E8**, **U9** si et seulement s'il existe une assignation respectant $\mathcal{T}$ telle que $\forall \mathcal{G} \in \mathbf{G_U}$, $\forall \varphi, \alpha \in \mathscr{L}$,*

- $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] = \{\mathcal{G}_1 \in [\alpha] \text{ tel que } (\mathcal{G}, \mathcal{G}_1) \in \mathcal{T} \text{ et } \forall \mathcal{G}_2 \in [\alpha] \text{ tel que } (\mathcal{G}, \mathcal{G}_2) \in \mathcal{T}, \mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_2\}$
- $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \left| \begin{array}{ll} \varnothing & \text{si } \exists \mathcal{G} \in [\varphi] \text{ tel que } [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] = \varnothing \\ \bigcup_{\mathcal{G} \in [\varphi]} [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] & \text{sinon.} \end{array} \right.$

Ce résultat est important car il donne un lien entre l'existence d'une assignation de préordres et le fait qu'un opérateur d'appartenance forcée satisfait les postulats décrits ci-dessus. Cependant, comme tout théorème de représentation, il ne donne aucun indice quant à la façon d'assigner ces préordres, c'est-à-dire comment concevoir précisément un opérateur d'appartenance forcée.

La proposition suivante établit le fait que cinq postulats sont nécessaires et suffisants pour définir un opérateur d'appartenance forcée, à savoir **E3**, **U4**, **E5**, **E8** et **U9** (nous avons vu dans la proposition 7.4 page précédente que **U1** peut être dérivé à partir de ces derniers).

Proposition 7.5. ***E3**, **U4**, **E5**, **E8**, **U9** constituent un ensemble minimal : aucun postulat ne peut être dérivé des autres.*

Grâce au théorème 7.2, nous pouvons déduire deux cas simples d'impossibilité : si la situation initiale ou le but sont impossibles, alors l'appartenance forcée est impossible (ce résultat constitue, en quelque sorte, la réciproque de **U3**).

Proposition 7.6. *Si $\diamondsuit_{\mathcal{T}}$ satisfait **E3**, **U4**, **E5**, **E8** et **U9**, alors ($[\varphi] = \varnothing$ ou $[\alpha] = \varnothing \implies [\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$).*

La proposition suivante assure que si une appartenance forcée est possible, alors une appartenance forcée plus générale est également possible.

Proposition 7.7. *Si $\diamondsuit_{\mathcal{T}}$ satisfait **E3**, alors ($[\varphi \diamondsuit_{\mathcal{T}} \alpha] \neq \varnothing \implies [\varphi \diamondsuit_{\mathcal{T}} (\alpha \vee \beta)] \neq \varnothing$).*

Il est important de noter qu'il existe certains cas où **U2** n'est pas compatible avec **E3**, **U4**, **E5**, **E8** et **U9**. En effet, si **U2** est imposé, alors l'opérateur d'appartenance forcée est associé avec un préordre dans lequel un graphe est toujours plus proche de lui-même que de tout autre graphe. C'est pour cette raison qu'il impose d'avoir une assignation fidèle. Dans ce cas, la relation représentée par $\mathcal{T}$ doit être réflexive.

Définition 7.7 (**Assignation fidèle de préordres**)**.** *Une* assignation fidèle *est une fonction qui associe à chaque $\mathcal{G} \in \mathbf{G_U}$ un préordre complet*[6] $\preceq_{\mathcal{G}}$ *tel que* $\forall \mathcal{G}_1 \in \mathbf{G_U}$, $\mathcal{G} \prec_{\mathcal{G}} \mathcal{G}_1$.

Proposition 7.8. *Étant donné une relation* réflexive $\mathcal{T} \subseteq \mathbf{G_U} \times \mathbf{G_U}$ *de transitions permises, il existe un opérateur* $\diamondsuit_{\mathcal{T}} : \mathscr{L} \times \mathscr{L} \to \mathscr{L}$ *satisfaisant* ***E3****,* ***U4****,* ***E5****,* ***E8****,* ***U9*** *et qui satisfait* ***U2*** *si et seulement s'il existe une assignation fidèle respectant* $\mathcal{T}$ *définie telle que dans le théorème 7.2 page précédente.*

Si nous supprimons la contrainte à propos des transitions permises, alors nous retrouvons le théorème de Katsuno et Mendelzon, à savoir :

Proposition 7.9. *Si* $\mathcal{T} = \mathbf{G_U} \times \mathbf{G_U}$*, alors* $\diamondsuit_{\mathcal{T}}$ *satisfait* ***U2****,* ***E3****,* ***U4****,* ***E5****,* ***E8****,* ***U9*** *si et seulement si* $\diamondsuit$ *satisfait* ***U1****,* ***U2****,* ***U3****,* ***U4****,* ***U5****,* ***U8*** *et* ***U9****.*

Parmi les différentes sortes de changements proposés par Baumann (2012b), l'expansion normale, c'est-à-dire l'addition d'un argument ainsi que des attaques le concernant, peut être traitée dans notre cadre de la façon suivante.

Remarque 7.1. *L'appartenance forcée, au sens de Baumann, par une expansion normale est un opérateur d'appartenance forcée particulier* $\diamondsuit_{\mathcal{T}} : \mathscr{L} \times \mathscr{L} \to \mathscr{L}$ *tel que* $\mathcal{T} = \mathcal{T}_B$*, où* $\mathcal{T}_B$ *est l'ensemble de transitions permises décrit dans la section 7.3 page 142. De plus, le langage utilisé est restreint de la manière suivante : les formules qui décrivent le système initial sont restreintes à* $\{\varphi \in \mathscr{L}_1, |[\varphi]| = 1\}$ *et les formules qui décrivent les faits dont l'appartenance doit être forcée sont seulement des conjonctions de littéraux positifs de* $\mathscr{L}_2$*, où* $\mathscr{L}_1$ *est le langage propositionnel construit sur les littéraux* $arg(x)$ *et* $att(x, y)$ *et* $\mathscr{L}_2$ *est le langage propositionnel construit uniquement sur les littéraux* $acc(x)$ *(avec* $x, y \in \mathcal{A}_\mathbf{U}$*).*

Dans le cadre de Baumann, la formule concernant le graphe initial doit être complète, c'est-à-dire doit correspondre à un seul graphe. La formule concernant le but de l'appartenance forcée doit décrire un ensemble d'arguments devant être acceptés (dans le cadre d'une sémantique donnée) après le changement. Baumann propose d'utiliser $\mathcal{G}' \preceq_{\mathcal{G}} \mathcal{G}''$ si et seulement si $dist(\mathcal{G}, \mathcal{G}') \leq dist(\mathcal{G}, \mathcal{G}'')$, où $dist(\mathcal{G}, \mathcal{G}')$ est le nombre d'attaques qui diffèrent entre $\mathcal{G}$ et $\mathcal{G}'$. Donc, grâce au théorème 7.2 page précédente, nous savons que l'opérateur de Baumann satisfait nos postulats et peut donc être vu comme une mise à jour.[7]

6. Dans la suite de ce travail, $\prec_{\mathcal{G}}$ est classiquement défini à partir de $\preceq_{\mathcal{G}}$ par : $a \prec_{\mathcal{G}} b$ si et seulement si $a \preceq_{\mathcal{G}} b$ et $b \not\preceq_{\mathcal{G}} a$ (voir annexe B page 195).

7. Notons que puisque l'appartenance forcée de Baumann est définie sur un graphe et non un ensemble de graphes, elle se trouve également être une sorte de révision de croyances. En effet, la

7.5 Discussion

Synthèse

Dans ce chapitre, nous avons examiné le parallèle entre l'appartenance forcée en argumentation et la théorie de la mise à jour de croyances. Ceci nous a permis d'adopter une approche axiomatique du changement en argumentation abstraite et de passer outre l'hypothèse du cadre formel présenté dans le chapitre 6 page 109 statuant que le système cible doit être totalement connu par l'agent voulant le modifier.

En nous basant sur les chapitres précédents, nous avons pris en compte la possibilité de supprimer un argument, ce qui peut, éventuellement, aider à forcer l'appartenance d'un ensemble d'arguments avec moins d'effort. En outre, la notion de ce qui peut être forcé a été généralisée ; il est en effet possible de non seulement forcer l'appartenance d'un ensemble d'arguments, mais aussi de forcer tout but exprimable dans la logique propositionnelle.

D'autre part, nous avons tenu compte de la possibilité de restreindre les changements possibles, ce qui se traduit par un ensemble de transitions permises entre graphes d'argumentation. Ceci nous permet de prendre en compte les notions du cadre présenté dans le chapitre 6 page 109 (par exemple, la notion d'opération exécutable ; voir section 6.1.2 page 113). De manière plus générale, cela nous permet de considérer uniquement les opérations que l'on souhaite voir se produire et de déterminer les arguments pouvant être ajoutés ou supprimés. Cette prise en compte des contraintes portant sur les transitions entre graphes fait de l'appartenance forcée présentée dans ce chapitre une généralisation de la mise à jour classique.

Travaux connexes

Les principales références dans le domaine de l'argumentation à propos de l'appartenance forcée sont Baumann et Brewka (2010); Baumann (2012b) qui se penchent sur la question suivante : *est-il possible de modifier un système d'argumentation donné, en lui appliquant des opérations de changement, de telle sorte qu'un ensemble d'arguments devienne accepté ?* Baumann a spécifié des conditions nécessaires et suffisantes sous lesquelles l'appartenance forcée est possible, dans le cas où les opérations de changement sont restreintes à l'addition de nouveaux arguments et de nouvelles attaques. Plus précisément, Baumann (2012b) introduit trois types de changement appelés “expansions” :

- l'*expansion normale*, qui ajoute de nouveaux arguments et de nouvelles attaques concernant au moins un des nouveaux arguments,
- l'*expansion faible*, qui raffine l'expansion normale par l'addition de nouveaux arguments n'attaquant aucun des anciens arguments et

révision et la mise à jour de croyances se confondent lorsque le monde initial est complètement connu ; à ce titre, ce genre de révision de croyances ne se place pas dans la pure tradition d'*AGM*, mais utilise plutôt des contraintes de transition.

- l'*expansion forte*, qui raffine l'expansion normale par l'addition de nouveaux arguments n'étant attaqués par aucun des anciens arguments.

De manière générale, il n'est pas possible de forcer l'appartenance de tout ensemble d'arguments en utilisant une expansion particulière. De plus, dans certains cas, plusieurs appartenances forcées sont possibles, certaines d'entre elles nécessitant plus d'efforts que d'autres. De manière à capturer cette idée, Baumann (2012b) introduit la notion de *caractéristique* qui dépend d'une sémantique et d'un ensemble d'expansions possibles. La caractéristique d'un ensemble d'arguments est définie comme le nombre minimal de modifications (les modifications étant les différences entre les attaques des deux graphes) qui sont nécessaires pour forcer l'appartenance de cet ensemble d'arguments. Ce nombre est égal à zéro lorsque chaque argument de l'ensemble désiré est déjà accepté. Il est par contre égal à l'infini si aucune appartenance forcée n'est possible. Baumann (2012b) fournit de plus des moyens pour calculer la caractéristique en fonction d'un type d'expansion et d'une sémantique donnés. Cette notion de caractéristique est une évaluation d'un changement ; elle est donc liée à une relation de préférence entre changements dont l'existence est attestée grâce au théorème 7.2 page 147.

Notons également que Booth *et al.* (2013), déjà évoqué dans la section 6.3 page 131, introduit ce qui peut être vu comme un opérateur d'appartenance forcée fondé sur une relation de préférence, cette dernière étant basée sur une distance entre les labels [8].

L'approche consistant à définir un opérateur de mise à jour basé sur un ensemble de transitions permises est apparue pour la première fois dans Cordier et Siegel (Cordier et Siegel (1995)). Leur proposition va au-delà de la nôtre puisqu'ils utilisent des contraintes de transition avec priorités, ce qui permet une plus grande expressivité. Néanmoins, cette proposition n'est définie qu'à un niveau sémantique (en termes de préordres entre les mondes), et ils ne fournissent donc pas de postulats ou de théorème de représentation associés à leur opérateur de mise à jour. De plus, notre idée de définir des postulats en fonction d'un ensemble $\mathcal{T}$ de transitions permises généralise Herzig (2005) puisque les contraintes d'intégrité que leur cadre permet de représenter peuvent être encodées avec $\mathcal{T}$ (l'inverse n'étant pas possible).

Signalons par ailleurs que Moguillansky *et al.* (2010) caresse également l'idée d'étudier un opérateur de révision à travers la notion de *désactivation* d'arguments, c'est-à-dire de rendre rejetés des arguments de façon à *activer* (rendre accepté) un argument particulier. Mais il n'est pas ici question de définir des postulats ou un théorème de représentation ; l'objectif des auteurs est plutôt d'étudier le changement d'un point de vue pratique tout en gardant des notions étroitement liées à la théorie du changement de croyance, en particulier le changement minimal. Le principe consiste tout d'abord à déterminer des *lignes d'arguments attaquantes* (des successions d'arguments qui s'attaquent en cascade et qui empêchent un argument

8. Le lecteur trouvera en section 2.3.1.2 page 23 quelques détails supplémentaires quant aux sémantiques basées sur les labels.

particulier d'être accepté), puis de rajouter de nouveaux arguments de sorte que les lignes attaquantes ne soient plus opérantes. La difficulté est ici que le ou les arguments ajoutés peuvent influer sur plusieurs lignes d'arguments et créer de nouvelles lignes attaquantes, ce qui impose de bien choisir les arguments à ajouter. Par ailleurs, la notion de minimalité se retrouve dans le choix de l'argument à attaquer dans la ligne d'arguments : un changement est minimal s'il modifie, en termes de statut, le moins d'arguments possibles de la ligne.

Notons enfin que le langage que nous proposons dans la section 7.2 page 139 n'est qu'un exemple et, pour une meilleure capture des notions de sémantique, le lecteur peut se référer à Villata *et al.* (2012) ; ce travail introduit un formalisme logique permettant de manipuler (représenter et raisonner sur) les notions d'extension (introduites par Dung (1995)) dans le langage. Ce dernier article s'inscrit dans la continuité du langage logique proposé par Besnard et Doutre (2004). Ce langage, ayant pour symboles les arguments du graphe, permet d'écrire des formules dont les modèles sont donc des ensembles d'arguments. Il est ainsi possible, grâce à lui, de donner l'expression logique des extensions issues des sémantiques classiques de Dung (1995). Néanmoins, notons que ce langage est destiné à caractériser les extensions et ne permet donc pas, notamment, de raisonner à la fois sur la structure du graphe et la sémantique. Un langage a l'expressivité semblable a été proposé par Coste-Marquis *et al.* (2006). Nonobstant, le but de Coste-Marquis *et al.* (2006) est complètement différent : il se propose de généraliser le cadre formel de Dung (1995) par la prise en compte de contraintes additionnelles (exprimées en logique) sur les ensembles admissibles d'arguments. Par ailleurs, un langage logique a également été proposé par Wooldridge *et al.* (2005), lequel permet de représenter et de raisonner sur l'acceptabilité, les conflits et la défense d'ensembles d'arguments ; notons néanmoins que ce formalisme est dédié à des arguments logiques (de type support-conclusion).

Au fil des chapitres précédents, après l'avoir justifiée, nous avons fait une étude formelle du changement en argumentation abstraite. Bien que les notions introduites soient intéressantes en elles-mêmes, il est néanmoins important de se demander si elles sont utilisables effectivement. Pour répondre à cette interrogation, le chapitre suivant se concentre sur l'implémentation des notions que nous avons vues jusque-là et leur étude d'un point de vue pratique.

Quatrième partie

Contributions pratiques

Chapitre 8

Implémentation

Vanité que vouloir changer le monde. Le monde change à son heure, malgré ceux qui veulent le changer.

Robert Marteau, *Mont-Royal.*

Sommaire

Tout au long de ce travail, nous avons proposé des réponses à certaines des interrogations classiques sur le changement en argumentation, "*Comment le représenter ? Quelles en sont les conséquences ?*", ainsi que de nouvelles questions ayant émergé plus récemment et concernant l'usage de tels changements, "*A qui sont destinés ces changements ? Comment les utiliser de manière effective pour arriver à un but précis ?*". Ce sont précisément ces deux questions : *le changement optimal* et *la prise en compte de l'auditoire*, qui ont guidé notre recherche et nous ont finalement amené à proposer un nouveau cadre théorique, exposé dans le chapitre 6 page 109.

Ce cadre, proche d'une certaine notion de planification, appelle naturellement une mise en œuvre de façon à pouvoir évaluer ses qualités de façon empirique. C'est pour cette raison que nous proposons dans ce chapitre un outil logiciel implémentant notre travail. Cet outil prend en entrée le système d'argumentation d'un agent, un système cible et un but. Il est alors capable de fournir en sortie la liste des opérations exécutables par l'agent sur le système cible afin de réaliser son but. Pour réaliser ce logiciel, nous avons utilisé une grande partie des notions présentées et étudiées dans ce travail, notamment, outre le cadre défini dans le chapitre 6 page 109, les propriétés caractérisant le changement établies dans le chapitre 5 page 81 et donc, par extension, la typologie du changement vue dans le chapitre 4 page 59.

Ainsi, nous présentons en section 8.1 les détails de l'implémentation de l'outil. En section 8.2 page 160, les résultats obtenus et les protocoles d'expérimentation utilisés sont donnés. Enfin, la section 8.3 page 165 conclut ce chapitre en présentant d'autres outils ayant une finalité similaire.

Notons que ce chapitre est présenté dans Bisquert *et al.* (2013a,d).

8.1 Présentation de l'outil

Cet outil s'organise autour de deux modules spécifiques : un gestionnaire de systèmes d'argumentation, et un moteur de calcul des opérations de changement. Les sorties du premier module servent d'entrée au deuxième module.

8.1.1 Le gestionnaire de systèmes d'argumentation

Nous présentons ici la facette argumentative "pure" du programme, qui concerne la création de différents systèmes d'argumentation. Nous avons implémenté la notion de système d'argumentation, et lui avons adjoint la possibilité de calculer ses propres extensions. Ce module est implémenté en utilisant un langage à objet (*Python 2.7*) pour la flexibilité et la facilité que cela procure dans l'implémentation des notions utilisées.

Ainsi, pour créer un système d'argumentation, il faut fournir :

- un ensemble d'arguments,
- un ensemble d'attaques.

Ce module gère la cohérence des données fournies en vérifiant par exemple qu'un argument n'apparaît qu'une seule fois, ou qu'une attaque n'existe que si les arguments concernés existent aussi. Il renvoie des informations concernant les systèmes d'argumentation et lance si nécessaire le calcul des extensions relativement à une sémantique pouvant être fournie en paramètre.

Conformément à notre cadre théorique, l'outil permet de créer plusieurs systèmes d'argumentation, et donc en particulier celui de l'agent et celui de sa cible. Cette création s'effectue en fournissant la liste des arguments et des attaques de ces deux systèmes. De plus, la sémantique utilisée dans le système cible est aussi précisée (elle servira à vérifier la satisfaction des buts de l'agent).

Les systèmes d'argumentation de l'agent et de la cible, ainsi que les extensions de ce dernier sont alors transmis au deuxième module.

8.1.2 Le moteur de calcul

Ce deuxième module est destiné à calculer des "opérations de changement". Plus précisément, il permet de répondre à la question *"Quelles sont les opérations exécutables par l'agent sur le système cible réalisant son but ?"*.

Le moteur de calcul prend en entrée les systèmes d'argumentation de l'agent et de la cible, ainsi que l'ensemble d'extensions de la cible pour une sémantique donnée. En plus de cela, il est nécessaire de lui fournir :

- un but correspondant à ce que l'agent cherche à satisfaire dans le système cible (la section 6.1.3 page 119 présente la syntaxe de ces buts),
- une base de caractérisations permettant de vérifier si une opération réalise le but fourni (les caractérisations existantes sont données dans le chapitre 5 page 81).

Notons que l'utilisateur peut filtrer les résultats en forçant le type d'opération attendu, il peut ne s'intéresser qu'aux additions d'arguments par exemple.

Le cœur de ce module est une règle qui :

- génère toutes les opérations exécutables par l'agent sur la cible et réalisant le but,
- et produit, pour chacune des opérations, la caractérisation justifiant ce résultat.

Cette règle comporte deux parties, l'une se chargeant de construire les opérations, et l'autre vérifiant que l'opération correspond aux *desiderata* de l'agent.

Une vision synthétique de l'outil, et donc de l'articulation entre les deux modules de l'outil, est donnée par la figure 8.1 page suivante.

Notons que nous utilisons dans ce second module un langage de programmation logique (*Prolog*) pour deux raisons primordiales : les caractérisations se traduisent naturellement en règles logiques, ce qui facilite l'implémentation, et le mécanisme d'unification nous permet de générer et de contraindre facilement des opérations à partir des systèmes d'argumentation et du but de l'agent.

Construction des opérations

La construction des opérations découle des définitions 6.3 à 6.6 pages 115–118. Ainsi, nous générons directement une opération exécutable, puis nous calculons son impact. Ceci permet d'éviter de considérer des opérations non autorisées ou non exécutables et ainsi d'optimiser le temps de calcul.

Vérification des opérations

Lorsqu'une opération est générée par les règles de construction, celle-ci est traitée grâce aux caractérisations. Ainsi, pour une opération donnée, s'il existe une caractérisation correspondant au type de l'opération et au but recherché par l'agent et

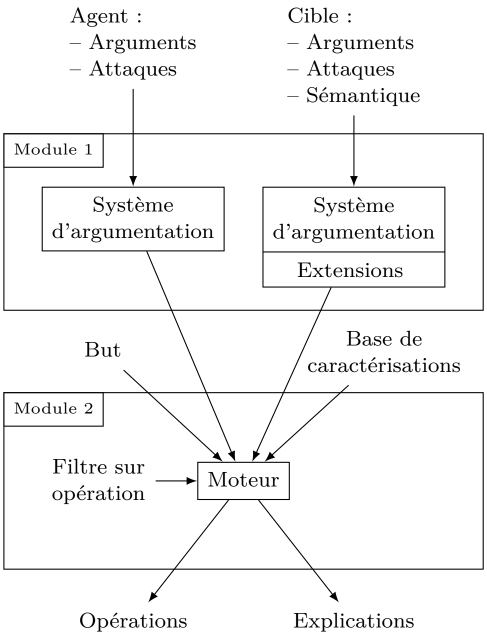


Figure 8.1 – *Architecture schématique de l'outil.*

si les conditions de cette caractérisation sont satisfaites, alors cette opération sera fournie à l'utilisateur avec la caractérisation correspondante à titre d'explication.

Il est important de noter que l'outil ne traite pour l'instant que des programmes réduits à une opération élémentaire. Par ailleurs, puisqu'il se repose sur une base de caractérisations issues de celles vues dans le chapitre 5 page 81, l'outil n'est pour le moment capable de produire que des opérations d'addition ou de suppression d'argument [1] ; néanmoins, il passera naturellement, c'est-à-dire sans modification logicielle, à l'addition ou suppression d'attaque lorsqu'on lui adjoindra des caractérisations concernant ces deux types de changement.

8.1.3 Application à l'exemple de l'audience

Nous illustrons l'utilisation de l'outil sur l'exemple de l'audience. Pour ce faire, considérons la situation suivante : le procureur doit faire en sorte que l'argument x_1 soit accepté. Il a à sa disposition les arguments présentés dans la figure 8.2a page suivante et doit modifier le système d'argumentation des jurés, représenté par la figure 8.2b page ci-contre.

Le procureur doit donc tout d'abord fournir à l'outil les données nécessaires, à savoir :

1. Plus précisément, il est capable de construire toute opération exécutable possible (addition ou suppression d'argument et addition ou suppression d'attaque). Par contre, il n'est pas capable de vérifier si une opération d'addition ou suppression d'attaque est valable car il n'a pas de caractérisation correspondante.

(a) *Procureur ($\mathcal{G}_{proc}$).* **(b)** *Jurés ($\mathcal{G}_{jurés}$).*

Figure 8.2 – *Système d'argumentation du procureur et des jurés.*

- l'ensemble des arguments et des attaques qu'il connaît : $\{x_0, x_1, x_3, x_4, x_5, x_6\}$ et $\{(x_0, x_1), (x_1, x_0), (x_4, x_1), (x_5, x_4), (x_6, x_1)\}$,
- l'ensemble des arguments et attaques connus par les jurés (le système cible) : $(\{x_0, x_1, x_4\}, \{(x_1, x_0), (x_4, x_1)\})$,
- la sémantique utilisée par les jurés, que nous supposons ici être la sémantique basique.

L'outil déduira de ces données l'ensemble des extensions pour le système des jurés ($\mathbf{E} = \{\{x_0, x_4\}\}$).

Le procureur doit aussi préciser le but qu'il souhaite voir satisfait : "x_1 *doit appartenir à l'extension basique.*"

Et enfin il doit fournir une base de caractérisations. Nous supposerons pour les besoins de cet exemple que cette base se résume à l'unique caractérisation 5.20 page 91 ; pour faciliter la lecture de cette section, nous la réécrivons ci-après.

Proposition 5.20. *Lors de l'addition d'un argument z dans le cadre de la sémantique basique, si z n'est pas attaqué par $\mathcal{G}$ et z défend indirectement x et $x \notin \mathcal{E}$, alors $x \in \mathcal{E}'$.*

L'outil génère alors les opérations exécutables par le procureur sur le système des jurés puis calcule leurs impacts. L'outil génère par exemple les opérations :

- $(\ominus, \{x_4\}, \varnothing)$, qui passe avec succès les étapes de construction et dont l'impact est le système ayant pour ensemble d'arguments $\{x_0, x_1\}$ et pour ensemble d'attaques $\{(x_1, x_0)\}$,
- $(\oplus, \{x_5\}, \{(x_5, x_4)\})$, qui passe également avec succès les étapes de construction et dont l'impact est le système $(\{x_0, x_1, x_4, x_5\}, \{(x_1, x_0), (x_4, x_1), (x_5, x_4)\})$.

Notons que les opérations concernant des arguments ou attaques inconnus de l'agent, ou non exécutables sur le système cible, ne sont pas générées par l'outil. Ainsi, l'opération $(\ominus, \{x_6\}, \varnothing)$ n'est pas générée car l'argument x_6 n'est pas présent dans le système des jurés. Ceci assure le caractère fini du processus.

Les opérations générées sont en même temps examinées au travers des caractérisations. Si une opération ne correspond à aucune des caractérisations, elle est rejetée ; si elle est concernée par une ou plusieurs caractérisations, alors l'outil renvoie tous les couples (opération, caractérisation).

Dans notre exemple, l'opération $(\ominus, \{x_4\}, \varnothing)$ est rejetée car son type (retrait d'un argument) ne correspond pas à celui spécifié dans l'unique caractérisation disponible. Par contre, le type de l'opération $(\oplus, \{x_5\}, \{(x_5, x_4)\})$ correspond ; l'outil doit donc vérifier que cette dernière satisfait les contraintes spécifiées dans la caractérisation, à savoir que z, en l'occurrence apparié à l'argument x_5, n'est pas attaqué et qu'il doit défendre indirectement x (ce dernier apparié à x_1) tel que x n'appartient pas à l'extension. [2] Ces conditions étant vérifiées, l'opération réalise effectivement le but du procureur.

8.2 Expérimentations

L'objectif de notre outil est de trouver un programme (au sens de la définition 6.7 page 119) réalisant un but. Rappelons que nous nous restreignons dans un premier temps aux *programmes ne contenant qu'une seule opération*. Le travail aurait pu être fait directement en calculant l'impact d'une opération puis l'ensemble des extensions du graphe obtenu. Mais l'outil est conçu dans le but d'éviter de recalculer des extensions, ce qui peut être très coûteux. En effet, si calculer l'impact d'une opération est aisé en termes d'arguments et d'attaques (car n'impliquant que des traitements "simples" d'ensembles), il est bien plus lourd de recalculer les ensembles d'extensions (Liao *et al.* (2011)). Notre idée est de générer des opérations exécutables et de les "tester" grâce aux caractérisations, plutôt que de calculer les extensions pour chacun des systèmes résultants puis de vérifier que le but sélectionné est satisfait. Le gain en termes de temps et d'espace reste encore à évaluer, ce que nous comptons faire dans des travaux futurs.

Avant d'analyser les résultats produits par l'outil (section 8.2.2 page 162), nous présentons, en section 8.2.1, les protocoles expérimentaux utilisés, qui seront ensuite critiqués en section 8.2.3 page 164.

8.2.1 Protocoles expérimentaux

Les deux protocoles expérimentaux mis en place visent en premier lieu à vérifier la correction de notre implémentation, et dans un deuxième temps, à mettre en lumière des lacunes au niveau des caractérisations. Notons à ce titre que l'emploi de l'outil dans ce cadre ne résume pas son utilisation finale potentielle puisqu'il ne s'agit, ici, que de protocoles permettant de le tester.

2. Les concepts nécessaires aux caractérisations ont été implémentés. Nous comptons ainsi parmi eux les concepts d'attaque indirecte par un argument, de défense directe et indirecte par un argument, d'attaque et de défense d'ensembles, etc.

8.2.1.1 Hypothèse d'inclusion

Ces deux protocoles supposent que le système cible est un sous-graphe partiel du système de l'agent. Cette hypothèse paraît naturelle dans le cas de l'exemple de l'audience : le procureur et l'avocat entendent publiquement ce qui a été dit au cours de l'audience. Ils sont donc au courant de tous les arguments présents dans le système d'argumentation des jurés (leur cible). [3]

8.2.1.2 Génération aléatoire

Ce protocole vise à générer aléatoirement deux systèmes (un pour l'agent et un pour la cible en respectant l'hypothèse d'inclusion) et un but à satisfaire :

- un ensemble d'arguments de taille n est créé, avec n ayant une valeur aléatoire entre 2 et 20,
- un ensemble d'attaques entre ces arguments de taille nb_att est créé, avec nb_att ayant une valeur aléatoire entre $n * 0.5$ et $n * 1.5$.

Ceci constitue le premier système. Il va servir de base pour créer un sous-graphe partiel représentant la cible :

- deux nombres $suppr\_arg$ et $suppr\_att$ sont générés aléatoirement, avec $suppr\_arg$ ayant une valeur entre 0 et $n-1$ et $suppr\_att$ ayant une valeur entre 0 et $nb\_att - 1$,
- $suppr\_arg$ arguments sont retirés aléatoirement de l'ensemble d'arguments, et pour chaque argument retiré nous soustrayons à $suppr\_att$ le nombre d'attaques le concernant,
- $suppr\_att$ attaques sont retirées aléatoirement.

Ceci nous permet d'obtenir le deuxième système. Et enfin, un but est choisi aléatoirement parmi $x \in \mathcal{E}'$, $x \notin \mathcal{E}'$, $z \in \mathcal{E}'$, $\mathcal{E} = \mathcal{E}'$, $\mathcal{E} \subseteq \mathcal{E}'$, $\mathcal{E} \subset \mathcal{E}'$, $\mathcal{E}' \subseteq \mathcal{E}$, $\mathcal{E}' \subset \mathcal{E}$, $\mathcal{E}' = \varnothing$ et $\mathcal{E}' \neq \varnothing$, où x appartient au système cible, z appartient au système de l'agent et $\mathcal{E}$ (resp. $\mathcal{E}'$) est l'extension basique de la cible avant (resp. après) le changement.

La génération d'opérations est alors lancée à partir de ces deux systèmes et du but (voir les résultats obtenus en section 8.2.2 page suivante).

8.2.1.3 Génération systématique

Bien que la génération aléatoire soit efficace, elle peut laisser de côté des cas intéressants. Pour pallier ce problème, nous avons mis en place un autre protocole expérimental consistant à tester exhaustivement une liste de couples de systèmes.

Ainsi, pour n arguments, nous créons tous les couples de systèmes tels que :

3. Le non-respect de cette hypothèse signifierait que l'agent pourrait ne pas être d'accord avec la validité de certains arguments ou attaques entendus. On se pose là la question complexe de la différentiation entre connaissance d'un argument ou d'une attaque et croyance en sa validité. Cette question mériterait un approfondissement qui sort du cadre de ce travail.

- le premier système a un ensemble d'arguments arg_agent (numérotés de 1 à n) et un ensemble d'attaques att_agent (dont le nombre va de 0 à $n*(n-1)$).
- le deuxième système a un ensemble d'arguments arg_cible variant parmi tous les sous-ensembles possibles de arg_agent, et un ensemble d'attaques variant parmi tous les sous-ensembles possibles de att_agent restreint aux arguments de arg_cible.

Nous procédons alors à la génération d'opérations pour un but particulier donné (ici, nous avons testé le but d'appartenance forcée pour chaque argument du système cible).

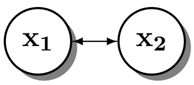
Notre protocole calcule l'extension basique du système résultant de l'opération. Ceci nous permet de constater les problèmes de "couverture" de l'outil, c'est-à-dire les cas où aucune opération n'est trouvée pour réaliser le but alors qu'il en existe une. Ces résultats de couverture sont présentés dans la section suivante.

8.2.2 Résultats

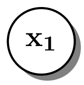
Pour la génération aléatoire :

Pour chaque couple de systèmes généré (en tout cinq millions), le premier protocole expérimental affiche des informations concernant la génération des opérations exécutables réalisant un but choisi aléatoirement. L'exemple 8.1 propose un court extrait montrant un cas de réussite de génération d'opérations.

Exemple 8.1. *Considérons les systèmes suivants et le but $x_1 \notin \mathcal{E}'$:*



(a) *Agent.*



(b) *Cible.*

L'outil génère deux opérations : $(\bigoplus, \{x_2\}, \{(x_2, x_1)\})$ et $(\bigoplus, \{x_2\}, \{(x_2, x_1), (x_1, x_2)\})$ avec les caractérisations correspondantes (ici, il s'agit deux fois de la même : celle donnée par la proposition 5.21 page 91).

Par ailleurs, le protocole propose également un récapitulatif des résultats, pour chacun des buts, en détaillant le nombre de cas où une solution est trouvée. La table 8.1 page suivante montre un exemple de résultats.

Selon le but considéré, l'outil trouve plus ou moins souvent des solutions. Plus précisément, il existe deux causes possibles amenant l'outil à ne pas trouver d'opérations exécutables :

- soit *il n'existe pas d'opération exécutable* réalisant le but cherché ; notons que ce cas inclut celui où il n'est pas possible de réaliser ce but avec une seule opération (voir l'exemple 8.2 page ci-contre).
- soit *il manque une caractérisation* à l'outil permettant de trouver au moins une opération (voir l'exemple 8.3 page 164).

But	**Nombre de cas testés**	**Ensemble de solutions**		**%**
		Non vide	**Vide**	
$x \in \mathcal{E}'$	499216	443010	56206	88.7
$x \notin \mathcal{E}'$	500009	397706	102303	79.6
$z \in \mathcal{E}'$	500196	443931	56265	88.8
$\mathcal{E} = \mathcal{E}'$	499770	389933	109837	78.0
$\mathcal{E} \subseteq \mathcal{E}'$	499116	499116	0	100.0
$\mathcal{E} \subset \mathcal{E}'$	500697	489562	11135	97.8
$\mathcal{E}' \subseteq \mathcal{E}$	499860	435600	64260	87.1
$\mathcal{E}' \subset \mathcal{E}$	499546	402207	97339	80.5
$\mathcal{E}' = \varnothing$	500728	27222	473506	5.4
$\mathcal{E}' \neq \varnothing$	500862	279162	221700	55.7
Total	5000000	3807449	1192551	76.1

Table 8.1 – *Récapitulatif de résultats pour le protocole de génération aléatoire. Pour chaque but, la deuxième colonne indique le nombre total de couples de systèmes d'argumentation testés par l'outil. La troisième colonne (resp. quatrième colonne) indique le nombre de fois où l'outil a renvoyé un ensemble de solutions non vide (resp. vide) pour un couple de systèmes. Enfin, la cinquième colonne indique le pourcentage des cas où l'outil a renvoyé un ensemble de solutions non vide par rapport au total des cas testés pour un but particulier.*

Exemple 8.2. *Considérons les systèmes suivants et le but $\mathcal{E}' = \varnothing$:*

(a) *Agent.* **(b)** *Cible.*

L'outil ne génère aucune opération car il est impossible d'avoir une opération exécutable réalisant le but recherché : d'une part, il n'existe pas d'argument attaquant en même temps x_2, x_3 et x_4 et étant attaqué par au moins l'un d'entre eux, et, d'autre part, il n'est pas possible de retirer tous les arguments en une seule opération.

Exemple 8.3. *Considérons les systèmes suivants et le but $\mathcal{E}' \neq \varnothing$:*

(a) *Agent.* **(b)** *Cible.*

L'outil ne génère aucune opération, et pourtant, ne rien faire suffirait à réaliser le but ; il manque donc à l'outil au moins une caractérisation affirmant que lorsque l'on ne fait rien, les extensions ne changent pas.

L'ensemble des solutions fourni par l'outil peut donc être vide pour des raisons très différentes mais qui ne sont pas distinguées par ce protocole. C'est pour mettre en évidence des exemples relevant du deuxième cas (montrant des lacunes dans les caractérisations) que nous avons mis en place le deuxième protocole expérimental, dont nous donnons maintenant les résultats.

Pour la génération systématique :

En considérant un but particulier, et en traitant tous les cas possibles, ce protocole nous a permis de nous concentrer sur les cas où le système ne trouve pas de solution et ainsi de pouvoir détecter les manques dans notre base de caractérisations.

Nous avons ainsi pu compléter notre base de caractérisations de telle sorte qu'actuellement la génération automatique pour $n = 3$ ou $= 4$ ne détecte plus aucun manque de caractérisation pour le but correspondant à l'appartenance forcée d'un argument à l'extension basique.

8.2.3 Critiques des protocoles

Notre protocole de génération aléatoire a permis de tester un grand nombre d'exemples (avec différents buts). Cependant, nous n'avons pas étudié la pertinence du choix de ces exemples, c'est-à-dire, les choix effectués ne sont pas forcément représentatifs de tous les cas.

Quant à la génération systématique, elle est très coûteuse : pour une génération concernant n arguments, il y a $2^{n*(n-1)}$ systèmes possibles pour l'agent et dans le pire des cas $2^{n*(n-1)} + (n * 2^{(n-1)*(n-2)}) + \cdots + (n * 2^0)$ cas possibles pour la cible. Le nombre total de couples possibles est de l'ordre de $2^{n*(n-1)} * (2^{n*(n-1)} + (n * 2^{(n-1)*(n-2)}) + \cdots + (n * 2^0))$. Ceci est tolérable pour un petit nombre d'arguments. Par exemple, pour $n = 3$, il y a $2^6 * (2^6 + 3 * 2^2 + 3) = 5056$ couples possibles. Mais cela devient rédhibitoire à partir de cinq arguments. Ce problème est en partie dû à la génération inutile de couples de graphes isomorphes.

Le deuxième inconvénient de ce protocole est qu'il dépend du choix du but. En effet, pour étudier la "couverture" de l'outil, lorsqu'il ne trouve pas d'opération permettant de satisfaire le but fixé, il est nécessaire d'effectuer des contrôles particuliers pour vérifier qu'il n'en existe effectivement aucune. Pour chaque but considéré, il est ainsi nécessaire de réfléchir et de fournir des tests appropriés pour vérifier si ce but peut être satisfait ou non.

8.3 Discussion

Synthèse

Dans ce chapitre, nous avons présenté une implémentation des notions étudiées tout au long de ce travail sous la forme d'un outil logiciel. Celui-ci permet à un utilisateur de trouver une opération de changement réalisant un but à la condition que l'utilisateur lui fournisse un système d'argumentation "source" représentant ses connaissances ainsi qu'un système d'argumentation cible à modifier. Cet outil peut être utilisé en tant qu'*assistant cognitif*[4] ("cognitive assistant") d'un utilisateur cherchant, par exemple, à analyser un dialogue.

Après avoir décrit le fonctionnement interne de l'outil, nous avons étudié son comportement au moyen de deux protocoles expérimentaux. Ces protocoles ont été critiqués, ce qui permet d'envisager des améliorations en ce qui concerne les expérimentations, notamment au niveau du choix des graphes à générer.

Travaux connexes

Plusieurs outils logiciels ont déjà été conçus dans le but d'aider un utilisateur à raisonner avec des arguments. Ils sont destinés à être des *assistants cognitifs* (voir par exemple van Gelder (2007); Tecuci *et al.* (2011)) ou se spécialisent dans un domaine particulier (par exemple, Ashley (2002); Aleven (2003) dans le domaine du raisonnement légal basé sur des cas). Notre proposition a tendance à se situer davantage à un niveau stratégique, puisqu'elle vise à aider un utilisateur à persuader un auditoire et qu'elle fournit des moyens pour le faire sous la forme d'addition ou de suppression d'arguments.

Nous avons également parlé, au sein de ce chapitre, de Liao *et al.* (2011). Le but de ce travail est pratique dans le sens où il vise à fournir un mécanisme permettant de recalculer efficacement les extensions après un changement subi par un système d'argumentation. L'approche est ici de diviser le système en trois parties :

- la partie non affectée par le changement, dans laquelle le statut des arguments reste le même qu'avant le changement,
- la partie affectée par le changement, où il est nécessaire de recalculer le statut des arguments,

4. Un assistant cognitif est un programme informatique autonome capable de résoudre des tâches diverses de façon à assister un utilisateur.

- la partie *conditionnante* ("conditioning"), c'est-à-dire certains arguments non affectés aidant au calcul du statut des arguments affectés.

Cette division est faite suivant un principe de dépendance : un argument b est affecté par un argument a s'il existe un chemin de a vers b ; dans le cas contraire, b n'est pas affecté par a. La partie affectée (respectivement, non affectée) est donc l'ensemble des arguments étant affectés (respectivement, non affectés) par l'argument ajouté ou supprimé ; la partie conditionnante est un sous-ensemble d'arguments non affectés tels qu'il existe une attaque entre ces arguments et un argument affecté.

Après cette division, il est possible de recalculer le statut des seuls arguments affectés en construisant un *système d'argumentation conditionné* ("conditioned argumentation framework"), qui est en fait un sous-graphe partiel du système d'origine constitué uniquement des parties affectée et conditionnante. Le statut des arguments conditionnants, calculer directement à partir des extensions d'origine, permet de calculer le nouveau statut des arguments affectés.

Liao *et al.* (2011) propose alors une méthode pour combiner les statuts des différentes parties et obtenir les extensions après le changement.

Enfin, notons la présence d'un travail très proche, dans l'esprit, du nôtre. Kontarinis *et al.* (2013) a pour objectif de permettre à un agent de satisfaire un but dans un système d'argumentation représentant l'état courant d'un débat grâce à un nombre minimal de modifications. Les buts sont ici restreints à l'acceptation ou au rejet d'un argument particulier. Notons que dans cet article, il est supposé que les arguments ne peuvent pas être modifiés mais il est par contre possible de faire plusieurs modifications à la fois. Les changements possibles sont donc restreints à l'addition et/ou la suppression d'une ou plusieurs attaques.

Par ailleurs, s'ils n'ont pas la notion des deux systèmes d'argumentation servant à définir les opérations exécutables, les auteurs de ce travail utilisent une définition légèrement modifiée d'un système d'argumentation. En effet, à la définition classique d'un système d'argumentation, ils adjoignent deux autres relations binaires : la première est un ensemble d'attaques n'étant pas dans le système (mais concernant les mêmes arguments) et qu'il est possible d'ajouter, et la seconde est un ensemble d'attaques étant dans le système et qu'il est possible de supprimer.

De façon à calculer les modifications permettant la satisfaction d'un but, Kontarinis *et al.* (2013) se base sur la notion d'*atome* ("atom") d'un système d'argumentation qui représente, entre autres, des actions (addition ou suppression d'une attaque) ou des buts (tel argument doit être accepté ou rejeté), etc. À partir d'ensembles d'atomes d'action sont construits des *mouvements* ("moves") (qui sont proches de notre notion de *programme*). Le principe est alors de calculer les mouvements possibles permettant d'obtenir un système d'argumentation satisfaisant un but donné ; parmi ces mouvements, ne sont retenus que ceux qui sont minimaux en termes d'inclusion. Le calcul des mouvements minimaux se fait grâce à des règles de réécriture (notamment basées sur la sémantique d'attaque ; voir la section 2.3.1.2 page 23) : un axiome représentant le but est réécrit, ce qui produit de nouveaux termes qui sont à leur tour réécrits successivement jusqu'à ce qu'il ne soit plus possible de réécrire. Chaque terme obtenu à la fin du calcul est un mouvement sur le système d'argumentation.

Enfin, les auteurs étudient les résultats fournis par l'outil qu'ils ont implémenté en considérant la correction et la complétude des mouvements obtenus.

Chapitre 9

Conclusion et perspectives

Un rien, mais ce rien est tout.

Honoré de Balzac,
Le Chef-d'œuvre inconnu.

Sommaire

La présentation des travaux de cette thèse arrive à son terme. Afin de conclure, il est temps de revenir sur ce que nous avons proposé au fil des pages de ce document et de donner les futures directions scientifiques qui nous motiveront dans les années à venir.

9.1 Résumé conclusif

Si l'on devait extraire la quintessence de cette thèse, elle serait tout entière consacrée au changement en argumentation. Plus précisément, les contributions principales en sont :

- la justification de l'intérêt de l'argumentation abstraite, et des changements s'y appliquant, pour une utilisation dans un contexte concret à travers la présentation de trois exemples applicatifs,
- l'enrichissement du cadre de base en étendant la typologie des propriétés du changement (les modifications que peut subir un système d'argumentation) de façon à permettre la prise en compte de l'opération de suppression d'un argument, et un travail détaillé de caractérisation de ces propriétés, à la fois pour l'addition et pour la suppression d'un argument,
- la spécialisation de la notion de changement à travers l'introduction d'un cadre permettant de contraindre les opérations de changement et de considérer la notion d'optimalité ainsi qu'une axiomatisation du changement,
- l'implémentation des notions étudiées sous la forme d'un outil logiciel.

Au sein du chapitre 2, nous avons rappelé les bases de notre travail. Nous nous sommes principalement reposés sur les travaux de Dung (1995) et Cayrol *et al.* (2010c).

À Dung (1995), article fondateur de l'argumentation abstraite, nous empruntons tout d'abord les notions de système d'argumentation, paire constituée d'un ensemble de symboles représentant des arguments et d'une relation binaire sur cet ensemble représentant les attaques entre les arguments. Nous empruntons ensuite la notion de sémantique qui permet de calculer des extensions (des ensembles d'arguments acceptables) desquelles il est possible de déduire le statut de chacun des arguments (accepté sceptiquement, accepté crédulement ou rejeté).

À Cayrol *et al.* (2010c), nous empruntons les définitions des opérations de changement qui sont des moyens de modifier un système d'argumentation. Ces opérations sont au nombre de quatre et ont pour fonction d'ajouter ou supprimer un argument et d'ajouter ou supprimer une attaque.

Le chapitre 3 constitue la première de nos contributions. Il comporte une réflexion à propos de la portée du domaine de l'argumentation abstraite. Loin d'être une théorie purement abstraite n'ayant aucune application pratique, nous avons montré à force d'exemples que l'argumentation abstraite se révèle emplie de ressources pour gérer des processus de la vie quotidienne. En particulier, nous avons présenté trois exemples détaillés. Le premier d'entre eux met en scène l'affrontement d'un procureur et d'un avocat dans le cadre d'une audience de tribunal. Leur but est de convaincre un auditoire, les jurés, et ils usent pour cela d'opérations de changement. L'intérêt principal de cet exemple est de donner une intuition des notions théoriques que nous utilisons, d'expliquer ce que les sémantiques et le statut des arguments abstraits peuvent signifier dans une argumentation réelle. Le deuxième exemple, proche du premier, montre comment l'argumentation abstraite et les notions de changement peuvent aider à modéliser la prise de décision entre plusieurs personnes. On y voit comment un agent peut agir sur un système d'argumentation pour satisfaire un but qui se trouve être l'objet de la soirée. Le troisième exemple fait montre de la flexibilité de l'argumentation abstraite. En effet, du fait de son caractère abstrait, elle peut s'appliquer à des contextes n'ayant pas ou peu de rapport direct avec l'argumentation au sens commun du langage (c'est-à-dire, en tant qu'outil d'échange d'arguments entre différents agents). Dans cet exemple, consacré à l'allocation de ressources, les arguments représentent des affectations de personnes à des postes. Dans ce contexte particulier, nous avons expliqué à quoi les notions théoriques, entre autres les sémantiques, statuts et opérations de changement, peuvent concrètement correspondre. Nous avons terminé ce chapitre en proposant une liste de cas concrets pouvant être capturés par chacune des opérations de changement.

Le chapitre 4 annonce le début de l'apport théorique de notre travail concernant la notion de "propriété du changement", c'est-à-dire les modifications qu'un changement induit sur un système d'argumentation. Nous y avons présenté une extension de la typologie des propriétés du changement de Cayrol *et al.* (2010c) de façon à prendre en compte une opération de changement peu traitée jusque-là, la suppression d'un argument. Cette nouvelle typologie propose une partition claire des modifications que peut subir un système d'argumentation à trois différents niveaux : le premier niveau regroupe les modifications impliquant la structure de l'ensemble des extensions (par exemple, l'augmentation ou la diminution du nombre d'extensions du système) ; le deuxième niveau regroupe les modifications ayant trait à l'acceptabilité conjointe des arguments d'une extension (par exemple, le fait que tous les arguments d'une extension restent conjointement acceptés après un changement) ; le troisième niveau regroupe les modifications touchant l'acceptabilité d'un argument particulier (par exemple, lorsqu'un argument accepté sceptiquement avant un changement se retrouve rejeté après).

Le chapitre 5 poursuit l'étude des propriétés du changement en présentant des résultats de caractérisation, c'est-à-dire des ensembles de conditions nécessaires et/ou suffisantes assurant qu'une opération de changement aura pour effet une modification particulière. Ces résultats sont obtenus de deux façons différentes : tout d'abord directement, en donnant des conditions que nous prouvons, puis grâce à une méthodologie basée sur la notion de dualité, lien unissant les opérations d'addition et de suppression d'un argument.

Le chapitre 6 entame la troisième partie de ce travail. Y est tout d'abord présenté un cadre permettant de traiter les notions de changement contraint et de but. Plus précisément, ce cadre autorise un agent ayant des connaissances (représentées par un système d'argumentation) à agir sur un système cible de façon à satisfaire son but. Néanmoins, les actions qu'il peut entreprendre (c'est-à-dire les opérations de changement qu'il peut utiliser) sont dépendantes de ses connaissances et du système cible. Nous avons modifié les opérations de changement de Cayrol *et al.* (2010c) pour s'assurer que les changements respectent les connaissances de l'agent (opération autorisée) et le système cible (opération exécutable) ; nous avons introduit la notion de programme qui représente une séquence d'opérations exécutables (satisfaisant éventuellement un but). La deuxième partie de ce chapitre est consacrée à l'étude du changement optimal dans ce cadre, c'est-à-dire comment choisir le “meilleur” changement possible au regard d'un ou plusieurs critères spécifiques (taille des programmes, nombre de modifications subies par le graphe, etc.). Nous avons montré comment les programmes peuvent être comparés par rapport à un critère, puis nous avons donné quelques pistes pour combiner plusieurs critères.

Le chapitre 7 se place au confluent de deux domaines : l'argumentation abstraite et la théorie de la mise à jour de croyances. Nous avons opéré un parallèle entre ces deux disciplines et nous avons montré que les notions issues de la mise à jour de croyances permettent un nouvel axe d'étude du changement en argumentation abstraite. Après avoir rappelé en quoi consiste la théorie du changement de croyances et expliqué le parallèle, nous avons introduit un langage propositionnel donnant la possibilité de décrire des graphes d'argumentation. Ceci nous a mené à l'étude de l'appartenance forcée généralisée (s'assurant qu'un but est satisfait, pour tout type de but exprimable dans le langage propositionnel). Le parallèle est ensuite effectivement mis en place à travers la définition de postulats de rationalité et d'un théorème de représentation statuant que lorsqu'un opérateur d'appartenance forcée satisfait les postulats, alors cet opérateur peut être défini au moyen d'une famille de préordres sur les graphes d'argumentation.

Le chapitre 8 délaisse la théorie pour la pratique en proposant une implémentation de la plupart des notions formelles que nous avons définies dans ces pages sous la forme d'un outil logiciel. Cet outil offre la possibilité à un utilisateur de calculer automatiquement une opération de changement satisfaisant un but sur un système cible à partir d'un autre système représentant les connaissances de cet utilisateur. Nous avons présenté la structuration en deux parties de cet outil, le gestionnaire de systèmes d'argumentation et le moteur de calcul des opérations, et nous avons exemplifié son utilisation dans le cadre de l'audience de tribunal. De façon à étudier les performances de cet outil, deux protocoles expérimentaux ont été proposés. Ces derniers ont pour but de montrer les lacunes de l'outil (les cas où il ne trouve pas de solution alors qu'il le devrait) et de vérifier qu'il ne fait pas d'erreur (les cas où les solutions proposées ne permettent pas de satisfaire le but donné en entrée). Le premier protocole génère aléatoirement des paires de systèmes d'argumentation (représentant le système de l'agent et le système cible) et tente de satisfaire un but généré aléatoirement lui aussi. Le second protocole génère un graphe complet (représentant le système de l'agent) puis génère exhaustivement tous ses sous-graphes partiels (chacun représentant un système cible). Pour chacune des paires formées par le graphe complet et un de ses sous-graphes partiels, ce protocole tente de satisfaire un but fixé à l'avance. Nous avons présenté les résultats obtenus grâce à ces protocoles et, pour conclure, nous avons donné quelques critiques constructives à leur propos.

9.2 Perspectives

Le travail proposé dans ces pages constitue en de nombreux points un défrichage de zones peu explorées jusque-là. Il a ainsi ouvert de nombreuses pistes prometteuses. Examinons-les chapitre par chapitre.

Le chapitre 3 a présenté plusieurs exemples montrant l'intérêt de l'argumentation abstraite dans des contextes concrets. S'il n'y a pas, à proprement parler, de perspectives en ce qui concerne ces exemples, il serait intéressant de proposer d'autres contextes et d'autres intuitions. Notons à ce titre que nos deux premiers exemples se limitent à des dialogues de persuasion et de délibération, avec des agents ayant des buts contradictoires. Il pourrait être intéressant de considérer d'autres types de dialogues mettant en jeu une coalition d'agents coopérant pour réaliser un but en commun (par exemple plusieurs avocats tentant d'étoffer une défense). De manière plus générale, approfondir cette piste implique d'essayer de délimiter les types de contextes pour lesquels l'argumentation abstraite se trouve être pertinente.

Le chapitre 4 a affiné la typologie présentée dans Cayrol *et al.* (2010c). Elle pourrait être affinée plus avant, ce qui soulève deux points à approfondir.

- D'autres propriétés du changement peuvent être définies. Il pourrait par exemple être souhaitable de représenter le gain ou la perte d'un argument spécifique au sein d'une ou plusieurs extensions particulières. Ou encore, qu'un argument, à défaut de pouvoir appartenir à toutes les extensions, appartienne à un plus grand nombre d'extensions après le changement qu'avant.
- Définir un grand nombre de propriétés du changement peut amener une certaine complexité à les utiliser toutes de concert. Il est donc important de savoir quelles propriétés sont véritablement nécessaires, ce qui dépend sans aucun doute du contexte dans lequel elles sont utilisées. Établir des sous-ensembles suffisants de propriétés suivant les contextes considérés permettrait de contrôler cette complexité ; l'outil présenté dans le chapitre 8 page 155 pourrait nous aider sur ce point.

Le chapitre 5 s'est intéressé, entre autres, à une facette inédite de l'argumentation, à savoir la dualité entre addition et suppression d'un argument. Par conséquent, de nombreux points restent à explorer ou approfondir. Deux points particuliers nous paraissent importants.

- Au sein de ce chapitre, et de manière générale dans l'ensemble des chapitres, nous n'avons considéré que deux des quatre opérations de changement de Cayrol *et al.* (2010c). Une des premières extensions de ce que nous avons proposé est d'examiner les deux opérations manquantes, à savoir l'addition et la suppression d'une interaction, et de compléter alors les résultats de caractérisation les concernant.
- Dans le cas des caractérisations obtenues par dualité, nous avons parfois été obligés de procéder à un traitement supplémentaire pour rendre utilisables certains des résultats, ce qui est problématique. Nous souhaitons trouver de nouvelles façons d'aborder la dualité pour éviter ce traitement ou, tout au moins, de trouver des critères permettant d'identifier avant l'application de la méthodologie les propositions qui auront besoin d'être transformées.

Le chapitre 6 a introduit un cadre formel permettant à un agent de choisir une opération de changement satisfaisant son but en fonction de ses connaissances et d'un système cible. Il ouvre de nombreuses perspectives ; voici une liste de ce que nous souhaitons examiner.

- Il convient tout d'abord d'approfondir la notion de programme. Plus précisément, nous nous intéressons au cas d'existence et d'unicité de programme réalisant l'ensemble des buts d'un agent dans différentes situations (une ou plusieurs opérations, prise en compte des critères d'optimalité, etc.). En d'autres termes, nous nous posons les questions "*Existe-t-il toujours un programme dans ce cas ?*" et "*En existe-t-il plusieurs ?*"
- De plus, l'étude de nouvelles notions en lien avec celle de programme semble pertinente : redondance (si une opération apparaît plusieurs fois dans un

même programme), vacuité (si deux opérations s'annulant mutuellement apparaissent dans un même programme) et ordre des opérations.

- Pour cela, il serait utile de considérer de nouvelles notions d'équivalences d'opérations portant sur les opérations en elles-mêmes (équivalence de type, d'argument et/ou d'attaques), sur leurs impacts (équivalence sur la structure des systèmes résultats) ou sur leurs effets (équivalence sur les extensions des systèmes résultats).
- Par ailleurs, l'optimalité de programme pose des questions intéressantes : "*Comment tirer parti des contraintes d'exécutabilité pour trouver des changements optimaux ?*" "*Est-il possible d'établir des classes d'équivalence de programmes dans lesquelles il serait utile de considérer la notion d'optimalité ?*"
- Nous nous intéressons également à la réécriture d'une opération élémentaire en une ou plusieurs opérations "basiques", c'est-à-dire l'addition ou la suppression d'un seul argument (sans attaque) ainsi que l'addition ou la suppression d'une seule attaque, de manière à permettre une définition plus homogène des opérations.
- L'hypothèse d'inclusion du système cible dans le système d'argumentation de l'agent pose la question de la mise à jour par l'agent du système représentant ses propres connaissances. En effet, si nous considérons l'exemple de l'audience de tribunal, il apparaît nécessaire d'étendre notre cadre de façon à permettre au procureur et à l'avocat de prendre en compte ce que dit son adversaire (notamment dans le cas où l'argument n'était pas connu), autorisant ainsi une réelle interaction entre ces derniers. De manière plus générale, ceci implique que nous étudiions les changements opérés par un agent sur son propre système lorsqu'un autre agent effectue une modification du système cible.
- Enfin, à plus long terme, nous souhaitons porter notre attention sur la notion de stratégie et son impact en termes de comparaison de programmes : approche *défensive*, où l'agent va tenter de défendre ses arguments, approche *agressive*, où l'agent va chercher à attaquer tout argument allant à l'encontre de son but, approche *pertinente*, où l'agent va n'avancer que des arguments qui appartiendront à une ou plusieurs extensions, approche *prudente*, où l'agent va révéler le moins d'informations possibles, etc.

Le chapitre 7 s'est penché sur le parallèle entre mise à jour de croyances et changement en argumentation abstraite. Là encore, plusieurs voies d'amélioration s'offrent à nous.

- Nous avons fait le choix de ne considérer que les buts absolus. Il est à noter que les buts relatifs pourraient être traités au moyen des transitions autorisées.
- Si les postulats que nous proposons sont valables, ils sont encore trop empreints de la mise à jour de croyances. Ils sont en particulier capables de caractériser les changements dans tout type de graphe pouvant être défini dans la logique propositionnelle (à condition qu'une fonction de transition

soit donnée). Ainsi, en premier lieu, il est important de trouver des postulats plus spécifiques à la dynamique en argumentation abstraite, notamment en prenant en compte les particularités des graphes représentant les systèmes d'argumentation (les notions de sémantiques pourraient, par exemple, être introduites dans les postulats).

- De plus, nous nous sommes concentrés dans ce chapitre sur un théorème de représentation basé sur des préordres *complets* entre des paires de graphes d'argumentation. Une autre étude est nécessaire pour un théorème basé sur des préordres *partiels*.
- Enfin, étudier ce qui pourrait être l'équivalent de l'appartenance forcée dans le cas de la révision de croyances plutôt que dans celui de la mise à jour nous semble valoir la peine. La révision de croyances consisterait en effet à considérer de nouvelles informations sur le système cible (par exemple, le fait de savoir que, finalement, l'argument que l'on croyait accepté dans le système cible ne l'est pas) ; il serait alors important de savoir de quelle façon prendre en compte ces nouvelles informations.

Finalement, le chapitre 8 a présenté l'implémentation de l'outil. De nombreux points sont à approfondir ; nous les déclinons selon deux axes.

Le premier axe concerne l'outil lui-même.

- Nous nous pencherons tout d'abord sur la complexité. Puisqu'un calcul des extensions après changement est très coûteux (malgré les progrès faits dans Liao *et al.* (2011)), notre objectif est de montrer que notre approche l'est moins. Cela nécessitera la détermination de la complexité de l'algorithme du moteur de calcul d'opérations (écrit en Prolog), une des difficultés étant d'intégrer à ce calcul la complexité liée à la vérification de l'applicabilité de certaines caractérisations. [1]
- Par ailleurs, nous prévoyons de lever la restriction de ne considérer qu'une seule opération élémentaire afin de pouvoir considérer les programmes (au sens du chapitre 6). Une étude des possibilités de planification permettra d'envisager la génération automatique de ces programmes. Ces derniers, en autorisant des modifications plus complexes, donneront lieu à plus de solutions.
- Nous avons évoqué l'utilisation de l'outil pour repérer des lacunes de caractérisation ; il pourrait être intéressant de développer un analyseur plus fin qui détecterait toutes les opérations exécutables que l'outil ne trouve pas, permettant ainsi de nous concentrer sur les caractérisations manquantes.

Le second axe concerne les expérimentations réalisées grâce à l'outil et ses applications :

- La poursuite d'expérimentations, telles celles présentées dans ce chapitre, est importante puisqu'elle permet de créer des "benchmarks" dans le domaine de l'argumentation.

1. En effet, certaines caractérisations utilisent des notions difficiles à calculer : par exemple la défense indirecte d'un argument par un ensemble d'arguments.

- Il pourrait être intéressant de considérer des cas réels d'argumentation pour établir d'autres "benchmarks", plus concrets. En particulier, les débats en ligne (sur des réseaux sociaux par exemple) semblent bien se prêter à cet exercice.
- Pour finir, le point précédent constitue également une piste d'applications, permettant éventuellement à un utilisateur d'être guidé dans son choix d'arguments à avancer lors de débats en ligne. Ceci nous permettrait sans doute d'obtenir quelques justifications supplémentaires à ajouter à celles du chapitre 3. La boucle serait alors bouclée.

Si tous les points précédents nous motivent à continuer nos recherches, il paraît important d'en extraire une direction principale pour nos futurs travaux. D'un point de vue plus général, nous pensons que la perspective primordiale est de continuer notre démarche de rapprochement entre argumentation abstraite et applications concrètes. En d'autres termes, de *donner du sens* à l'argumentation abstraite. À ce titre, nous avons vu dans le chapitre 3 page 29 deux contextes profitant pleinement de l'argumentation abstraite, à savoir le dialogue et l'allocation de ressources. Il nous semble ainsi essentiel de déterminer d'autres contextes et situations où l'argumentation abstraite est appropriée et, pour chacun d'entre eux, d'expliciter l'intérêt des notions théoriques (les sémantiques, les statuts, etc.). Ceci nécessitera une étude plus globale de ces contextes pour tenter d'extraire des caractéristiques particulières (multi-agents, relation de dominance entre arguments, système cible plus ou moins bien connu, etc.) ; il sera alors possible de choisir certaines notions théoriques adéquates selon les caractéristiques des situations considérées. Une telle approche réclamera sans aucun doute l'approfondissement de ces notions théoriques pour capturer les détails propres à chaque situation.

Bibliographie

ALCHOURRÓN, C., GÄRDENFORS, P. et MAKINSON, D. (1985). On the logic of theory change : partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530. (Cité en page 136.)

ALEVEN, V. (2003). Using background knowledge in case-based legal reasoning : A computational model and an intelligent learning environment. *Artificial Intelligence*, 150(1-2):183–237. (Cité en page 165.)

AMGOUD, L., BODENSTAFF, L., CAMINADA, M., MCBURNEY, P., PARSONS, S., PRAKKEN, H., van VEENEN, J. et VREESWIJK, G. (2006). Final review and report on formal argumentation system. Rapport technique, ASPIC Poject. Deliverable D2.6, ASPIC IST-FP6-002307. (Cité en pages 22 et 23.)

AMGOUD, L. et CAYROL, C. (2002a). Inferring from inconsistency in preference-based argumentation frameworks. *International Journal of Automated Reasoning*, 29(2):125–169. (Cité en pages 22 et 24.)

AMGOUD, L. et CAYROL, C. (2002b). A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1-3):197–215. (Cité en page 24.)

AMGOUD, L., CAYROL, C., LAGASQUIE-SCHIEX, M.-C. et LIVET, P. (2008). On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems, Bipolar Representations of Information and Preference (Part 2 : reasoning and learning)*, 23(10):1062–1093. (Cité en pages 12, 24 et 25.)

AMGOUD, L., CAYROL, C. et LE BERRE, D. (1996). Comparing arguments using preference orderings for argument-based reasoning. *In Tools with Artificial Intelligence, 1996., Proceedings Eighth IEEE International Conference on*, pages 400–403. (Cité en page 24.)

AMGOUD, L., MAUDET, N. et PARSONS, S. (2000a). Modelling dialogues using argumentation. *In Proceedings of the 4th International Conference on Multi-Agent Systems(ICMAS-2000)*, pages 31–38, Boston, MA. (Cité en page 23.)

AMGOUD, L., PARSONS, S. et MAUDET, N. (2000b). Arguments, dialogue, and negotiation. *In* HORN, W. (éditeur) : *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*, pages 338–342. IOS Press. (Cité en page 23.)

Amgoud, L. et Prade, H. (2004). Reaching agreement through argumentation : A possibilistic approach. *In Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 175–182. (Cité en page 23.)

Amgoud, L. et Prade, H. (2009). Using arguments for making and explaining decisions. *Artificial Intelligence*, 173(3-4):413–436. (Cité en page 23.)

Amgoud, L. et Vesic, S. (2012). A formal analysis of the role of argumentation in negotiation dialogues. *Journal of Logic and Computation*, 22:957–978. (Cité en page 23.)

Ashley, K. (2002). An AI model of case-based legal argument from a jurisprudential viewpoint. *Artificial Intelligence Law*, 10:163–218. (Cité en page 165.)

Audi, R. (1999). *The Cambridge Dictionary of Philosophy, Second Edition*. Cambridge University Press. (Cité en page 55.)

Austin, J. (1962). *How to Do Things with Words*. Oxford University Press, London. (Cité en page 21.)

Baroni, P., Giacomin, M. et Guida, G. (2005). Scc-recursiveness : a general schema for argumentation semantics. *Artificial Intelligence*, 168(1):162–210. (Cité en page 26.)

Baumann, R. (2012a). Normal and strong expansion equivalence for argumentation frameworks. *Artificial Intelligence*, 193:18–44. (Cité en page 27.)

Baumann, R. (2012b). What does it take to enforce an argument ? minimal change in abstract argumentation. *In* Raedt, L. D., Bessière, C., Dubois, D., Doherty, P., Frasconi, P., Heintz, F. et Lucas, P. J. F. (éditeurs) : *20th European Conference on Artificial Intelligence (ECAI 2012)*, volume 242 de *Frontiers in Artificial Intelligence and Applications*, pages 127–132. IOS Press. (Cité en pages 27, 144, 148, 149 et 150.)

Baumann, R. et Brewka, G. (2010). Expanding argumentation frameworks : Enforcing and monotonicity results. *In Proceeding of the 2010 conference on Computational Models of Argument : Proceedings of COMMA 2010*, pages 75–86, Amsterdam, The Netherlands, The Netherlands. IOS Press. (Cité en pages 27, 59, 135, 144 et 149.)

Besnard, P. et Doutre, S. (2004). Checking the acceptability of a set of arguments. *In* Delgrande, J. P. et Schaub, T. (éditeurs) : *10th International Workshop on Non-Monotonic Reasoning (NMR 2004), Whistler, Canada, June 6-8, 2004, Proceedings*, pages 59–64. (Cité en page 151.)

Besnard, P. et Hunter, A. (2001). A logic-based theory of deductive arguments. *Artificial Intelligence*, 128(1-2):203–235. (Cité en page 22.)

Besnard, P. et Hunter, A. (2008). *Elements of Argumentation*. MIT Press. (Cité en page 22.)

Bisquert, P. (2011). Changement dans un système d'argumentation : suppression d'un argument. *In 10ème Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA)*. (Cité en pages 8, 60 et 82.)

Bisquert, P., Cayrol, C., de Saint-Cyr, F. D. et Lagasquie-Schiex, M.-C. (2012a). Duality between addition and removal. *In* Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B. et Yager, R. R. (éditeurs) : *Advances on Computational Intelligence*, volume 297 de *Communications in Computer and Information Science*, pages 219–229. Springer. (Cité en pages 8 et 82.)

Bisquert, P., Cayrol, C., Dupin de Saint Cyr - Bannay, F. et Lagasquie-Schiex, M.-C. (2011). Change in argumentation systems : exploring the interest of removing an argument (regular paper). *In* Benferhat, S. et Grant, J. (éditeurs) : *International Conference on Scalable Uncertainty Management (SUM), Dayton, Ohio, 10/10/2011-12/10/2011*, pages 275–288, http ://www.springerlink.com/. Springer-Verlag. (Cité en pages 8 et 60.)

Bisquert, P., Cayrol, C., Dupin de Saint Cyr Bannay, F. et Lagasquie-Schiex, M.-C. (2012b). Changement dans un système d'argumentation : suppression d'un argument. *Revue d'Intelligence Artificielle*, 26(3/2012):225–254. (Cité en pages 8, 60 et 82.)

Bisquert, P., Cayrol, C., Dupin de Saint Cyr Bannay, F. et Lagasquie-Schiex, M.-C. (2013a). Changements guidés par les buts en argumentation : Cadre théorique et outil (regular paper). *In* Bonzon, E. et Cholvy, L. (éditeurs) : *Journées Francophones MODÈLES FORMELS de l'INTERACTION (MFI), Lille, 01/07/2013-02/07/2013*, page (en ligne), http ://afia-france.org/. AFIA. (Cité en pages 8 et 156.)

Bisquert, P., Cayrol, C., Dupin de Saint Cyr Bannay, F. et Lagasquie-Schiex, M.-C. (2013b). Characterizing change in abstract argumentation systems. *In* Simari, G. et Fermé, E. (éditeurs) : *Trends in Belief Revision and Argumentation Dynamics*, pages 1–30. College Publications, http ://www.collegepublications.co.uk/. (Cité en pages 8 et 82.)

Bisquert, P., Cayrol, C., Dupin de Saint Cyr Bannay, F. et Lagasquie-Schiex, M.-C. (2013c). Enforcement in argumentation is a kind of update (regular paper). *In* Liu, W., Subrahmanian, V. et Wijsen, J. (éditeurs) : *International Conference on Scalable Uncertainty Management (SUM), Washington DC, USA, 16/09/2013-18/09/2013*, numéro 8078 de LNAI, pages 30–43, http ://www.springerlink.com/. Springer-Verlag. (Cité en pages 8, 110 et 136.)

Bisquert, P., Cayrol, C., Dupin de Saint Cyr Bannay, F. et Lagasquie-Schiex, M.-C. (2013d). Goal-driven changes in argumentation : A theoretical framework and a tool. (regular paper). *In International Conference on Tools with Artificial Intelligence (ICTAI), Washington, USA, 04/11/2013-06/11/2013*, http ://ieeexplore.ieee.org/. IEEExplore digital library. (Cité en pages 8, 110 et 156.)

Black, E. et Atkinson, K. (2011). Agreeing what to do. *In Proceedings of the 7th international conference on Argumentation in Multi-Agent Systems (ArgMAS 2010)*, pages 12–30, Berlin, Heidelberg. Springer-Verlag. (Cité en page 23.)

Black, E. et Hunter, A. (2007). A generative inquiry dialogue system. *In* Durfee, E. H., Yokoo, M., Huhns, M. N. et Shehory, O. (éditeurs) : *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, pages 241 :1–241 :8. IFAAMAS. (Cité en page 23.)

Boella, G., Gabbay, D. M., van der Torre, L. et Villata, S. (2010). Support in abstract argumentation. *In Proceedings of the 2010 conference on Computational Models of Argument (COMMA 2010)*, pages 111–122, Amsterdam, The Netherlands, The Netherlands. IOS Press. (Cité en page 24.)

Boella, G., Kaci, S. et van der Torre, L. (2009a). Dynamics in argumentation with single extensions : Abstraction principles and the grounded extension. *In* Sossai, C. et Chemello, G. (éditeurs) : *10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2009)*, volume 5590 de *Lecture Notes in Computer Science*, pages 107–118. Springer. (Cité en pages 27, 59 et 105.)

Boella, G., Kaci, S. et van der Torre, L. (2009b). Dynamics in argumentation with single extensions : Attack refinement and the grounded extension. *In* Sierra, C., Castelfranchi, C., Decker, K. S. et Sichman, J. S. (éditeurs) : *8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 1213–1214. IFAAMAS. (Cité en pages 27, 59 et 105.)

Boella, G., Kaci, S. et van der Torre, L. (2009c). Dynamics in argumentation with single extensions : Attack refinement and the grounded extension (extended version). *In* McBurney, P., Rahwan, I., Parsons, S. et Maudet, N. (éditeurs) : *6th International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2009)*, volume 6057 de *Lecture Notes in Computer Science*, pages 150–159. Springer. (Cité en pages 27 et 105.)

Bonet, B. et Geffner, H. (1996). Arguing for decisions : A qualitative model of decision making. *In* Horvitz, E. et Jensen, F. V. (éditeurs) : *In Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, pages 98–105. Morgan Kaufmann. (Cité en page 23.)

BONZON, E. et MAUDET, N. (2011). On the outcomes of multiparty persuasion. *In Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'11)*, pages 47–54. (Cité en page 132.)

BOOTH, R., KACI, S., RIENSTRA, T. et TORRE, L. (2013). A logical theory about dynamics in abstract argumentation. *In* LIU, W., SUBRAHMANIAN, V. et WIJSEN, J. (éditeurs) : *Scalable Uncertainty Management*, volume 8078 de *Lecture Notes in Computer Science*, pages 148–161. Springer Berlin Heidelberg. (Cité en pages 27, 133 et 150.)

BOOTH, R., KACI, S., RIENSTRA, T. et van der TORRE, L. W. N. (2012). Conditional acceptance functions. *In* VERHEIJ, B., SZEIDER, S. et WOLTRAN, S. (éditeurs) : *Proceeding of the 2012 conference on Computational Models of Argument (COMMA 2012)*, volume 245 de *Frontiers in Artificial Intelligence and Applications*, pages 470–477. IOS Press. (Cité en page 27.)

BREWKA, G. (2001). Dynamic argument systems : A formal model of argumentation processes based on situation calculus. *Journal of Logic and Computation*, 11(2): 257–282. (Cité en page 30.)

BREWKA, G., DUNNE, P. E. et WOLTRAN, S. (2011). Relating the semantics of abstract dialectical frameworks and standard AFs. *In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 780–785. AAAI Press. (Cité en page 25.)

BREWKA, G., ELLMAUTHALER, S., STRASS, H., WALLNER, J. P. et WOLTRAN, S. (2013). Abstract dialectical frameworks revisited. *In* ROSSI, F. (éditeur) : *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 2013)*, pages 803–809. AAAI Press / IJCAI. (Cité en page 25.)

BREWKA, G. et WOLTRAN, S. (2010). Abstract dialectical frameworks. *In* LIN, F., SATTLER, U. et TRUSZCZYNSKI, M. (éditeurs) : *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010)*. AAAI Press. (Cité en pages 25 et 27.)

CAMINADA, M. (2006). Semi-stable semantics. *In Proceedings of the 2006 conference on Computational Models of Argument (COMMA 2006)*, pages 121–130, Amsterdam, The Netherlands, The Netherlands. IOS Press. (Cité en page 26.)

CAMINADA, M. et AMGOUD, L. (2007). On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5-6):286–310. (Cité en page 22.)

CAPOBIANCO, M., CHESÑEVAR, C. et SIMARI, G. (2005). Argumentation and the dynamics of warranted beliefs in changing environments. *Autonomous Agents and Multi-Agent Systems*, 11(2):127–151. (Cité en page 228.)

CAYROL, C., DEVRED, C. et LAGASQUIE-SCHIEX, M.-C. (2010a). Acceptability semantics accounting for strength of attacks in argumentation (short paper). *In*

European Conference on Artificial Intelligence (ECAI 2010), pages 995–996. IOS Press. (Cité en pages 24, 27 et 55.)

Cayrol, C., Devred, C. et Lagasquie-Schiex, M.-C. (2010b). Dialectical proofs accounting for strength of attacks in argumentation systems (regular paper). *In* Grégoire, E. (éditeur) : *International Conference on Tools with Artificial Intelligence (ICTAI 2010)*, pages 207–214. IEEE Computer Society - Conference Publishing Services. (Cité en page 24.)

Cayrol, C., Dupin de Saint Cyr, F. et Lagasquie-Schiex, M.-C. (2010c). Change in abstract argumentation frameworks : Adding an argument. *Journal of Artificial Intelligence Research*, 38:49–84. (Cité en pages 11, 16, 17, 18, 19, 20, 21, 28, 55, 59, 69, 70, 74, 82, 89, 90, 105, 113, 170, 171, 172, 173, 202, 203, 204 et 205.)

Cayrol, C. et Lagasquie-Schiex, M.-C. (2009). Bipolar abstract argumentation systems. *In* Rahwan, I. et Simari, G. (éditeurs) : *Argumentation in Artificial Intelligence*, chapitre 4, pages 65–84. Springer, http ://www.springerlink.com. (Cité en pages 12 et 24.)

Cayrol, C. et Lagasquie-Schiex, M.-C. (2010). Coalitions of arguments : a tool for handling bipolar argumentation frameworks. *International Journal of Intelligent Systems*, 25:83–109. (Cité en page 24.)

Cayrol, C. et Lagasquie-Schiex, M.-C. (2011). Bipolarity in argumentation graphs : towards a better understanding (regular paper). *In* Benferhat, S. et Grant, J. (éditeurs) : *International Conference on Scalable Uncertainty Management (SUM 2011)*, numéro 6929 de LNAI, pages 137–148, http ://www.springerlink.com. Springer. (Cité en page 24.)

Cayrol, C. et Lagasquie-Schiex, M.-C. (2013). Bipolarity in argumentation graphs : towards a better understanding. *International Journal of Approximate Reasoning*, xx:(on line). (Cité en pages 12 et 24.)

Cordier, M.-O. et Siegel, P. (1995). Prioritized transitions for updates. *In ECSQARU*, pages 142–150. (Cité en page 150.)

Coste-Marquis, S., Devred, C. et Marquis, P. (2005). Prudent semantics for argumentation frameworks. *In Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '05)*, pages 568–572, Washington, DC, USA. IEEE Computer Society. (Cité en page 26.)

Coste-Marquis, S., Devred, C. et Marquis, P. (2006). Constrained argumentation frameworks. *In Proc. of KR*, pages 112–122, Lake District. (Cité en pages 25 et 151.)

Coste-Marquis, S., Konieczny, S., Mailly, J.-G. et Marquis, P. (2013). On the revision of argumentation systems : Minimal change of arguments status. *In*

2nd International Workshop on Theory and Applications of Formal Argumentation(TAFA'13), Beijing. workshop at IJCAI'13. (Cité en pages 27 et 133.)

Coste-Marquis, S., Konieczny, S., Marquis, P. et Ouali, M. A. (2012). Weighted attacks in argumentation frameworks. *In* Brewka, G., Eiter, T. et McIlraith, S. A. (éditeurs) : *Proceedings of the Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning, (KR 2012)*. AAAI Press. (Cité en pages 24 et 27.)

Devred, C., Doutre, S., Lefevre, C. et Nicolas, P. (2010). Dialectical proofs for constrained argumentation. *In Proceeding of the 2010 conference on Computational Models of Argument (COMMA 2010)*, pages 159–170. (Cité en page 25.)

Dubois, D., Dupin de Saint-Cyr, F. et Prade, H. (1995). Update postulates without inertia (regular paper). *In Symbolic and Quantitative Approaches to Reasoning and Uncertainty (Selected papers of the Europ. Conf. ECSQARU'95), Fribourg, Switzerland,* , numéro 946 de LNAI, pages 162–170, http ://www.springerlink.com/. Springer-Verlag. (Cité en page 138.)

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358. (Cité en pages 11, 12, 13, 14, 21, 24, 26, 27, 47, 54, 111, 112, 151, 170, 202, 205 et 206.)

Dung, P. M., Mancarella, P. et Toni, F. (2007). Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(10-15):642–674. (Cité en page 26.)

Dunne, P. E. et Bench-Capon, T. J. (2001). Complexity and combinatorial properties of argument systems. Technical report, University of Liverpool, Department of Computer Science (U.L.C.S.). (Cité en page 15.)

Dunne, P. E. et Bench-Capon, T. J. (2002). Coherence in finite argument system. *Artificial Intelligence*, 141(1-2):187–203. (Cité en page 15.)

Dunne, P. E., Hunter, A., McBurney, P., Parsons, S. et Wooldridge, M. (2009). Inconsistency tolerance in weighted argument systems. *In Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 851–858, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. (Cité en page 24.)

Fellows, M. R. et Parberry, I. (1993). Getting children excited about computer science. *Computing Research News*, 5(1). (Cité en page 1.)

Fluet, C. (2011). L'économie de la preuve judiciaire. Rapport technique, CIRPÉ. Cahier de recherche/Working Paper 11-02. (Cité en page 25.)

Fox, J. et McBurney, P. (2002). Decision making by intelligent agents : logical argument, probabilistic inference and the maintenance of beliefs and acts. *In*

Benferhat, S. et Giunchiglia, E. (éditeurs) : *9th International Workshop on Non-Monotonic Reasoning (NMR 2002), April 19-21, Toulouse, France, Proceedings*, pages 293–301. (Cité en page 23.)

Fox, J. et Parsons, S. (1997). On using arguments for reasoning about actions and values. *In Proceedings of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, Stanford. (Cité en page 23.)

García, A. J. et Simari, G. R. (2004). Defeasible logic programming : an argumentative approach. *Theory and Practice of Logic Programming*, 4(2):95–138. (Cité en page 22.)

Gordon, T. F. et Karacapilidis, N. (1997). The zeno argumentation framework. *In Proceedings of the 6th international conference on Artificial intelligence and law*, ICAIL '97, pages 10–18, New York, NY, USA. ACM. (Cité en page 23.)

Governatori, G., Maher, M. J., Antoniou, G. et Billington, D. (2004). Argumentation semantics for defeasible logic. *Journal of Logic and Computation*, 14(5):675–702. (Cité en page 22.)

Grabisch, M. et Perny, P. (2003). Agrégation multicritère. *In* Bouchon-Meunier, B. et Marsala, C. (éditeurs) : *Logique floue, principes, aide à la décision*, pages 81–120. Hermès. (Cité en pages 128 et 129.)

Hadjinikolis, C., Modgil, S., Black, E., McBurney, P. et Siantos, Y. (2013). Opponent modelling in persuasion dialogues. *In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 2013)*. (Cité en page 23.)

Herzig, A. (2005). On updates with integrity constraints. *In Belief Change in Rational Agents*. (Cité en pages 138 et 150.)

Herzig, A. et Rifi, O. (1999). Propositional belief base update and minimal change. *Artificial Intelligence*, 115:107–138. (Cité en page 138.)

Hitchcock, D., Mcburney, P. et Parsons, S. (2001). A framework for deliberation dialogues. *In* Hansen, H. V., Tindale, C. W., Blair, J. A. et Johnson, R. H. (éditeurs) : *Proceedings of the 4th biennial conference of the Ontario Society for the Study of argumentation (OSSA 2001)*, Ontario, Canada. (Cité en page 23.)

Homère (1995). *L'Iliade. L'Odyssée.* Bouquins. Robert Laffont. (Cité en page 2.)

Jakobovits, H. et Vermeir, D. (1999). Robust semantics for argumentation frameworks. *Journal of Logic and Computation*, 9(2):215–261. (Cité en page 27.)

Kakas, A. et Moraitis, P. (2006). Adaptive agent negotiation via argumentation. *In Proceedings of the fifth international joint conference on Autonomous agents* *and multiagent systems (AAMAS '06)*, pages 384–391, New York, NY, USA. ACM. (Cité en page 23.)

KARACAPILIDIS, N. et PAPADIAS, D. (2001). Computer supported argumentation and collaborative decision making : the HERMES system. *Information Systems*, 26(4):259–277. (Cité en page 24.)

KATSUNO, H. et MENDELZON, A. (1991). On the difference between updating a knowledge base and revising it. *In* ALLEN, J. et AL. (éditeurs) : *Proc. of the 2nd Inter. Conf. on Principles of Knowledge Representation and Reasoning*, pages 387–394, Cambridge, MA. (Cité en pages 136, 137 et 138.)

KOK, E. M., MEYER, J.-J. C., PRAKKEN, H. et VREESWIJK, G. A. W. (2011). A formal argumentation framework for deliberation dialogues. *In Proceedings of the 7th international conference on Argumentation in Multi-Agent Systems (ArgMAS'10)*, pages 31–48, Berlin, Heidelberg. Springer-Verlag. (Cité en page 23.)

KONTARINIS, D., BONZON, E., MAUDET, N., PEROTTI, A., van der TORRE, L. et VILLATA, S. (2013). Rewriting rules for the computation of goal-oriented changes in an argumentation system. *In 14th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA'13)*. (Cité en pages 28 et 166.)

KRAUS, S., SYCARA, K. et EVENCHIK, A. (1998). Reaching agreements through argumentation : a logical model and implementation. *Artificial Intelligence*, 104(1-2):1–69. (Cité en page 23.)

LÉA SOMBÉ (1994). A glance at revision and updating in knowledge bases. *International Journal of Intelligent Systems*, 9(1):1–27. (Cité en page 136.)

LIAO, B. (2014). *Efficient Computation of Argumentation Semantics*. Intelligent Systems. Academic Press, Oxford, UK. (Cité en page 59.)

LIAO, B., JIN, L. et KOONS, R. C. (2011). Dynamics of argumentation systems : A division-based method. *Artificial Intelligence*, 175(11):1790 – 1814. (Cité en pages 28, 59, 160, 165, 166, 175 et 228.)

LINSBICHLER, T. (2013). On the limits of expressiveness in abstract argumentation semantics : Realizability and signatures. Mémoire de D.E.A., Technische Universität Wien. (Cité en page 78.)

MARTÍNEZ, D. C., GARCÍA, A. J. et SIMARI, G. R. (2007). On defense strength of blocking defeaters in admissible sets. *In Proceedings of the 2nd international conference on Knowledge science, engineering and management (KSEM'07)*, pages 140–152, Berlin, Heidelberg. Springer-Verlag. (Cité en page 24.)

MODGIL, S. (2009). Reasoning about preferences in argumentation frameworks. *Artif. Intell.*, 173(9-10):901–934. (Cité en page 24.)

Moguillansky, M. O., Rotstein, N. D., Falappa, M. A., García, A. J. et Simari, G. R. (2010). Argument theory change through defeater activation. *In Proceeding of the 2010 conference on Computational Models of Argument (COMMA 2010)*, pages 359–366, Amsterdam, The Netherlands, The Netherlands. IOS Press. (Cité en pages 27, 59 et 150.)

Parsons, S. et Jennings, N. R. (1996). Negotiation through argumentation – a preliminary report. *In Proceedings of the 2nd International Conference on Multi Agent Systems*, pages 267–274. (Cité en page 23.)

Parsons, S., Wooldridge, M. et Amgoud, L. (2003). Properties and Complexity of Some Formal Inter-agent Dialogues. *Journal of Logic and Computation*, 13(3): 347–376. (Cité en page 23.)

Perelman, C. et Olbrechts-Tyteca, L. (1958). *Traité de l'argumentation : La nouvelle rhétorique*. PUF, Paris, France. (Cité en page 3.)

Prakken, H. (2006). Formal systems for persuasion dialogue. *Knowledge Engineering Review*, 21(2):163–188. (Cité en page 23.)

Russell, S. J. et Norvig, P. (2002). *Artificial Intelligence : A Modern Approach (2nd Edition)*. Prentice Hall. (Cité en page 2.)

Simari, G. R. et Loui, R. P. (1992). A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53(2-3):125–157. (Cité en page 22.)

Tecuci, G., Marcu, D., Boicu, M., Schum, D. et K., R. (2011). Computational theory and cognitive assistant for intelligence analysis. *In Proceedings of STIDS*, pages 68–75. (Cité en page 165.)

van Gelder, T. (2007). The Rationale for Rationale. *Law, Probability and Risk*, 6:23–42. (Cité en page 165.)

Verheij, B. (1996). Two approaches to dialectical argumentation : admissible sets and argumentation stages. *In Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC'96)*, pages 357–368, Utrecht, NL. (Cité en pages 26 et 27.)

Verheij, B. (2003). Deflog : On the logical interpretation of prima facie justified assumptions. *Journal of Logic and Computation*, 13(3):319–346. (Cité en page 24.)

Villata, S., Boella, G., Gabbay, D. M., van der Torre, L. et Hulstijn, J. (2012). A logic of argumentation for specification and verification of abstract argumentation frameworks. *Annals of Mathematics and Artificial Intelligence*, 66(1-4):199–230. (Cité en page 151.)

VILLATA, S., BOELLA, G. et VAN DER TORRE, L. (2011). Attack semantics for abstract argumentation. *In Proceedings of the Twenty-Second international joint conference on Artificial Intelligence (IJCAI'11)*, pages 406–413. AAAI Press. (Cité en page 27.)

WALTON, D. et KRABBE, E. (1995). *Commitment in Dialogue : Basic Concepts of Interpersonal Reasoning*. SUNY series in Logic and Language. State University of New York Press. (Cité en page 23.)

WALTON, D. N. (2003). *Relevance in argumentation*. Lawrence Erlbaum. (Cité en page 39.)

WINSLETT, M. (1988). Reasoning about action using a possible models approach. *In Proc. of the 7^{th} National Conference on Artificial Intelligence*, pages 89–93, St. Paul. (Cité en pages 136 et 137.)

WOLTRAN, S. (2013). On the limits of expressiveness in abstract argumentation semantics. Séminaire “Belief Change and Argumentation in Multi-Agent Scenarios”, Schloss Dagstuhl - Leibniz Center for Informatics, Dagstuhl, Allemagne. (Cité en page 78.)

WOOLDRIDGE, M., MCBURNEY, P. et PARSONS, S. (2005). On the meta-logic of arguments. *In 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 560–567. ACM. (Cité en page 151.)

ZHUANG, Z. (2013). Argument rejection and acceptance through attack abstractions. Séminaire “Belief Change and Argumentation in Multi-Agent Scenarios”, Schloss Dagstuhl - Leibniz Center for Informatics, Dagstuhl, Allemagne. (Cité en page 133.)

Cinquième partie

Annexes

Annexe A

Notations

Notion	**Notation**
Élément de base	
Argument	u, v, w, x, y, z $(x_0, x_1, \cdots)$
Attaque	$x\mathcal{R}y$, (x, y)
Agent	$a_1, a_2, \cdots$
Opération	$o_1, o_2, \cdots$
Programme	$p_1, p_2, \cdots$
But	$b_1, b_2, \cdots$
Ensemble d'éléments de base	
Ensemble d'arguments	$\mathcal{A}$, $\mathcal{S}$
Extension	$\mathcal{E}$ $(\mathcal{E}_1, \mathcal{E}_2, \cdots)$
Ensemble d'arguments défendus par $\mathcal{S}$	$\mathcal{F}(\mathcal{S})$
Ensemble des arguments non attaqués dans $\mathcal{G} \setminus \{z\}$	$\mathcal{N}_z$
Ensemble de l'argument z et des arguments défendus uniquement par z	$\mathcal{D}_z$
Ensemble d'attaques (concernant l'argument x)	$\mathcal{R}$ $(\mathcal{R}_x)$
Ensemble des agents	$Agents$
Graphe d'argumentation	$\mathcal{G}$

Ensemble des opérations élémentaires	$\mathcal{O}$
Ensemble des opérations autorisées pour l'agent a_i	$\mathcal{O}_i$
Ensemble des opérations exécutables par l'agent a_i sur $\mathcal{G}$	$\mathcal{O}_i^{\mathcal{G}}$
Ensemble de buts (de l'agent a_i)	$\mathcal{B}$ ($\mathcal{B}_i$)
Ensemble de tous les programmes exécutables par l'agent a_i sur $\mathcal{G}$	$\mathcal{P}_i^{\mathcal{G}}$
Ensemble de tous les programmes de l'agent a_i sur $\mathcal{G}$ réalisant l'ensemble de buts $\mathcal{B}_i$	$\mathcal{P}_{\mathcal{B}_i}^{\mathcal{G}}$
Paire ou ensemble d'ensembles d'éléments de base	
Système d'argumentation	$(\mathcal{A}, \mathcal{R})$
Abstraction d'univers de référence	$\mathbf{U} = (\mathcal{A}_{\mathbf{U}}, \mathcal{R}_{\mathbf{U}})$
Système d'argumentation de l'agent a_i	$\mathbf{U}_i = (\mathcal{A}_i, \mathcal{R}_i)$
Ensemble d'extensions	$\mathbf{E}$
Ensemble d'extensions contenant l'argument x	$\mathbf{E}_x$
Ensemble des graphes d'argumentation (issus de $\mathbf{U}$)	$\mathbf{G}$ ($\mathbf{G}_{\mathbf{U}}$)
Autres	
Opération d'addition d'argument avec des interactions le concernant	$\circledcirc\!\!\!\!\oplus$
Opération de suppression d'argument avec ses interactions	$\circledcirc\!\!\!\!\ominus$
Opération d'addition d'attaque	$\oplus$

Opération suppression d'attaque	$\ominus$
Propriété du changement	$\pi_1, \pi_2, \cdots$
Ensemble des propriétés du changement	Π, Π_{aut}, Π_{exe}, Π_{rea}
Critère	γ
Ensemble de critères	Γ
Fonction associée à un critère	f_γ
Relation de préférence	$\preccurlyeq^\gamma$, $\approx^\gamma$, $\prec^\gamma$
Formule	ϕ, ψ
Langage	$\mathscr{L}$
Modèle	ω
Ensemble des modèles	Ω
Ensemble des modèles d'une formule ϕ	$[\phi]$
Ensemble des transitions	$\mathcal{T}$
Opérateur de mise à jour	$\diamond$
Operateur d'appartenance forcée (en fonction de $\mathcal{T}$)	$\diamondsuit_{\mathcal{T}}$
Symboles classiques	
Inclusion (stricte)	$\subseteq$ $(\subset)$
Appartenance	$\in$
Séquence	$\langle\cdots\rangle$
N-uplet	$(\cdots)$
Cardinal	$\|\cdots\|$
Symboles de la logique classique	$\wedge$, $\vee$, $\neg$, $\longrightarrow$, $\longleftrightarrow$, $\models$, $\vdash$, $\forall$, $\exists$

Table A.1 – *Liste des notations utilisées.*

Annexe B

Notions de base

Dans cette section, nous rappelons quelques notions utiles à notre travail. Plus précisément, nous nous penchons sur des notions d'ordre et de préordre qui sont des relations binaires particulières.

Mais avant de les présenter, rappelons qu'une *relation binaire* sur un ensemble A est un ensemble de couples (a, b) tels que a et b appartiennent à A. On dit alors que a et b sont en relation par R, ce que l'on note $(a, b) \in R$ (ou aRb).

Définition B.1 (**Ordre**). *Une relation binaire R sur un ensemble A est un* ordre *si et seulement si elle est :*

- réflexive *($\forall a \in A,\ (a, a) \in R$),*
- antisymétrique *($\forall a, b \in A,\ (a, b) \in R \wedge (b, a) \in R \longrightarrow a = b$),*
- transitive *($\forall a, b, c \in A,\ (a, b) \in R \wedge (b, c) \in R \longrightarrow (a, c) \in R$).*

Exemple B.1. *La relation binaire "$\leq$" portant sur l'ensemble des naturels est un ordre. La relation binaire "$<$" portant sur l'ensemble des naturels n'est pas un ordre car elle n'est pas réflexive.*

Définition B.2 (**Ordre strict**). *Une relation binaire R sur un ensemble A est un* ordre strict *si et seulement si elle est :*

- asymétrique *($\forall a, b \in A,\ (a, b) \in R \longrightarrow (b, a) \notin R$),*
- transitive.

Un ordre strict est par conséquent irréflexif *($\forall a \in A,\ (a, a) \notin R$),*

Exemple B.2. *La relation binaire "$<$" portant sur l'ensemble des naturels est un ordre strict.*

Définition B.3 (**Relation d'équivalence**). *Une relation binaire est une* relation d'équivalence *si et seulement si elle est :*

- *réflexive,*

- *symétrique* ($\forall a, b \in A,\ (a, b) \in R \longrightarrow (b, a) \in R$),
- *transitive.*

Exemple B.3. *La relation binaire "$=$" portant sur l'ensemble des naturels est une relation d'équivalence.*

Définition B.4 (Préordre). *Une relation binaire est un* préordre *si et seulement si elle est :*

- *réflexive,*
- *transitive.*

Exemple B.4. *Une relation binaire de type "est accessible en voiture" portant sur un ensemble de villes est un préordre. La relation binaire "est la mère de" portant sur un ensemble de personnes n'est par contre pas un préordre car elle n'est ni transitive ni réflexive.*

Définition B.5 (Relations déterminées par un préordre). *Étant donné R un préordre portant sur un ensemble A,*

- *l'*ordre strict déterminé par *R : $\{(a, b) \in A \times A \mid (a, b) \in R$ et $(b, a) \notin R\}$.*
- *la* relation d'équivalence déterminée par *R : $\{(a, b) \in A \times A \mid (a, b) \in R$ et $(b, a) \in R\}$.*
- *la* relation d'incomparabilité déterminée par *R : $\{(a, b) \in A \times A \mid (a, b) \notin R$ et $(b, a) \notin R\}$.*

Pour finir, nous considérons des *relations de préférence*. Une relation de préférence est une relation binaire que l'on note généralement $\preccurlyeq$; dans notre travail, elle sera toujours un préordre.

Étant donné $\preccurlyeq$ une relation de préférence portant sur un ensemble A et a, b deux éléments de A, nous dirons que *a est préféré à b* si et seulement si $a \preccurlyeq b$. À partir de cette première relation, il est possible d'en définir trois autres :

- $a \prec b$ si et seulement si $\prec$ est l'ordre strict déterminé par $\preccurlyeq$; nous dirons que *a est strictement préféré à b*,
- $a \approx b$ si et seulement si $\approx$ est la relation d'équivalence déterminée par $\preccurlyeq$; nous dirons que *a et b sont indifférents.*
- $a \sim b$ si et seulement si $\sim$ est la relation d'incomparabilité déterminée par $\preccurlyeq$; nous dirons que *a et b sont incomparables.*

Annexe C

Démonstrations des propositions

Sommaire

Nous trouvons ici l'ensemble des démonstrations de nos propositions et lemmes disséminés dans les chapitres 4 page 59 (section C.1), 5 page 81 (section C.2 page 199) et 7 page 135 (section C.3 page 211).

C.1 Typologie (chapitre 4)

Démonstration de la proposition 4.1 page 67.

- Dans le cas de la sémantique basique, il y a une seule extension. Etant donné la condition 1, les conditions 2 et 3 sont trivialement équivalentes.
- Dans le cas de la sémantique préférée ou stable, supposons que les extensions de $\mathcal{G}$ et $\mathcal{G}'$ satisfont les conditions 1 et 2.
  - — Nous allons tout d'abord établir qu'il n'est pas possible d'avoir deux extensions différentes $\mathcal{E}_i$ et $\mathcal{E}_k$ de $\mathcal{G}$ contenues dans une extension $\mathcal{E}'_j$ de $\mathcal{G}'$. Raisonnons par l'absurde. Supposons que $\mathcal{E}_i \subset \mathcal{E}'_j$, $\mathcal{E}_k \subset \mathcal{E}'_j$ et $\mathcal{E}_i \neq \mathcal{E}_k$.

L'ensemble $\mathcal{E}_i \cup \mathcal{E}_k$ est inclus dans $\mathcal{E}'_j$. Il est donc sans-conflit dans $\mathcal{G}'$, mais aussi dans $\mathcal{G}$ puisqu'il est contenu dans $\mathcal{G}$. $\mathcal{E}_i$ et $\mathcal{E}_k$ sont des extensions préférées de $\mathcal{G}$ (toute extension stable est aussi préférée), donc des ensembles qui défendent leurs éléments dans $\mathcal{G}$. L'ensemble $\mathcal{E}_i \cup \mathcal{E}_k$ défend donc lui aussi ses éléments dans $\mathcal{G}$. Donc cet ensemble est admissible dans $\mathcal{G}$. Enfin, $\mathcal{E}_i \neq \varnothing$ ce qui implique que l'ensemble $\mathcal{E}_i \cup \mathcal{E}_k$ contient strictement $\mathcal{E}_k$. Ce qui contredit le fait que $\mathcal{E}_k$ est maximal admissible.

— Le résultat précédent permet d'associer à chaque extension $\mathcal{E}_i$ une extension $\mathcal{E}'_{j(i)}$ de telle sorte que $\mathcal{E}_i \subset \mathcal{E}'_{j(i)}$ et que si $i \neq k$ alors $j(i) \neq j(k)$. $\mathbf{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_n\}$ donc l'ensemble des extensions de la forme $\mathcal{E}'_{j(i)}$ (qui est inclus dans $\mathbf{E}'$) contient n extensions. Par la condition 1, nous savons que $\mathbf{E}'$ contient aussi n extensions. Donc, $\mathbf{E}'$ est exactement égal à l'ensemble des extensions de la forme $\mathcal{E}'_{j(i)}$. En conséquence, toute extension de $\mathbf{E}'$ est de la forme $\mathcal{E}'_{j(i)}$ et contient donc strictement l'extension $\mathcal{E}_i$.

□

Démonstration de la proposition 4.2 page 68.

- Dans le cas de la sémantique basique, il y a une seule extension. Etant donné la condition 1, les conditions 2 et 3 sont trivialement équivalentes.
- Dans le cas de la sémantique préférée ou stable, supposons que les extensions de $\mathcal{G}$ et $\mathcal{G}'$ satisfont les conditions 1 et 3. La preuve est analogue à celle de la proposition concernant le changement *c-expansif*. On commence par établir par l'absurde qu'il n'est pas possible d'avoir deux extensions différentes $\mathcal{E}'_j$ et $\mathcal{E}'_k$ de $\mathcal{G}'$ contenues dans une extension $\mathcal{E}_i$ de $\mathcal{G}$.
On en déduit ensuite que l'on peut associer à chaque extension $\mathcal{E}'_j$ une extension $\mathcal{E}_{i(j)}$ de telle sorte que $\mathcal{E}'_j \subset \mathcal{E}_{i(j)}$ et que si $j \neq k$ alors $i(j) \neq i(k)$. L'ensemble des extensions de la forme $\mathcal{E}_{i(j)}$ (qui est inclus dans $\mathbf{E}$) contient $|\mathbf{E}'| = n$ extensions. Par la condition 1, on en déduit que l'ensemble des extensions de la forme $\mathcal{E}_{i(j)}$ est exactement égal à $\mathbf{E}$. En conséquence, toute extension de $\mathbf{E}$ est de la forme $\mathcal{E}_{i(j)}$ et contient donc strictement l'extension $\mathcal{E}'_j$.

□

Démonstration de la proposition 4.3 page 71.

- Le premier point découle directement des définitions 4.21 page 70 et 4.23 page 70.
- Le deuxième point découle directement des définitions 4.22 page 70 et 4.24 page 71.

□

C.2 Caractérisations (chapitre 5)

C.2.1 Démonstrations concernant les résultats préliminaires (section 5.1)

Démonstration du lemme 5.1 page 84.
Le fait que z n'attaque pas $\mathcal{E}'$ est équivalent au fait que si z attaque un argument $x \in \mathcal{G}'$ alors $x \notin \mathcal{E}'$. Or, $\mathcal{N}_z = \{x \in \mathcal{G} \setminus \{z\}$ tel que x n'est pas attaqué dans $\mathcal{G} \setminus \{z\}\}$. Nous avons $\mathcal{E}' = \mathcal{N}_z \cup \bigcup_{i \geq 1} \mathcal{F}'^i(\mathcal{N}_z)$. Donc, nous avons $x \notin \mathcal{E}'$ si et seulement si $x \notin \mathcal{N}_z$ et $x \notin \bigcup_{i \geq 1} \mathcal{F}'^i(\mathcal{N}_z)$, c'est-à-dire si et seulement si x est attaqué par $\mathcal{G} \setminus \{z\}$ et x n'est pas défendu indirectement par $\mathcal{N}_z$ dans $\mathcal{G} \setminus \{z\}$. □

La démonstration du lemme 5.2 page 84 nécessite des résultats intermédiaires qui sont donnés dans les lemmes C.1 à C.5 pages 199–200.

Lemme C.1. *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si z n'attaque pas $\mathcal{N}_z$, alors $\mathcal{N}_z \subseteq \mathcal{F}(\mathcal{N}_z)$ et donc pour tout $i \geq 1$, $\mathcal{F}^i(\mathcal{N}_z) \subseteq \mathcal{F}^{i+1}(\mathcal{N}_z)$.*

Démonstration du lemme C.1 de la présente page.
Par définition, $\mathcal{N}_z$ est l'ensemble des arguments non attaqués dans $\mathcal{G} \setminus \{z\}$. Donc, si z n'attaque pas $\mathcal{N}_z$, aucun argument de $\mathcal{N}_z$ n'est attaqué par $\mathcal{G}$. Donc, $\mathcal{N}_z \subseteq \mathcal{F}(\mathcal{N}_z)$. De plus, puisque $\mathcal{F}$ est monotone, pour tout $i \geq 1$, $\mathcal{F}^i(\mathcal{N}_z) \subseteq \mathcal{F}^{i+1}(\mathcal{N}_z)$. □

Lemme C.2. *Soit $\mathcal{S} \subseteq \mathcal{G} \setminus \{z\}$. Lors de la suppression d'un argument z dans le cadre de la sémantique basique, $\mathcal{F}(\mathcal{S}) \setminus \{z\} \subseteq \mathcal{F}'(\mathcal{S})$.*

Démonstration du lemme C.2 de la présente page.
Soit $y \in \mathcal{F}(\mathcal{S}) \setminus \{z\}$. Deux cas sont possibles. Premièrement, si y n'est pas attaqué par $\mathcal{G}'$, y appartient trivialement à $\mathcal{F}'(\mathcal{S})$. Deuxièmement, supposons que y est attaqué par $\mathcal{G}'$. $\forall w \in \mathcal{G}'$ tel que w attaque y dans $\mathcal{G}'$, puisque $y \in \mathcal{F}(\mathcal{S})$, $\mathcal{S}$ attaque w dans $\mathcal{G}$. Puisque $\mathcal{S} \subseteq \mathcal{G} \setminus \{z\}$, $\mathcal{S}$ défend y dans $\mathcal{G}'$ et donc $y \in \mathcal{F}'(\mathcal{S})$. □

Lemme C.3. *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si z n'attaque pas $\mathcal{N}_z$, alors $\forall i \geq 1$, si $z \notin \mathcal{F}^i(\mathcal{N}_z)$, alors $\mathcal{F}^i(\mathcal{N}_z) \subseteq \mathcal{F}'^i(\mathcal{N}_z)$.*

Démonstration du lemme C.3 de la présente page.
Prouvons par induction sur $i \geq 1$ que si z n'attaque pas $\mathcal{N}_z$ et $z \notin \mathcal{F}^i(\mathcal{N}_z)$, alors $\mathcal{F}^i(\mathcal{N}_z) \subseteq \mathcal{F}'^i(\mathcal{N}_z)$.

- Cas basique ($i = 1$) : $z \notin \mathcal{F}(\mathcal{N}_z)$. En utilisant le lemme C.2 avec $\mathcal{S} = \mathcal{N}_z$, nous obtenons $\mathcal{F}(\mathcal{N}_z) \subseteq \mathcal{F}'(\mathcal{N}_z)$.

- Hypothèse d'induction (pour $1 \leq i \leq p$, la proposition est vraie) : soit $z \notin \mathcal{F}^{p+1}(\mathcal{N}_z)$, grâce au lemme C.1 page précédente, $z \notin \mathcal{F}^p(\mathcal{N}_z)$. De plus, $\mathcal{F}^{p+1}(\mathcal{N}_z) = \mathcal{F}(\mathcal{F}^p(\mathcal{N}_z))$. En utilisant le lemme C.2 page précédente avec $\mathcal{S} = \mathcal{F}^p(\mathcal{N}_z)$, nous obtenons $\mathcal{F}^{p+1}(\mathcal{N}_z) \subseteq \mathcal{F}'(\mathcal{F}^p(\mathcal{N}_z))$. En utilisant l'hypothèse d'induction (puisque $z \notin \mathcal{F}^p(\mathcal{N}_z)$), nous avons $\mathcal{F}'(\mathcal{F}^p(\mathcal{N}_z)) \subseteq \mathcal{F}'(\mathcal{F}'^p(\mathcal{N}_z))$, donc par transitivité $\mathcal{F}^{p+1}(\mathcal{N}_z) \subseteq \mathcal{F}'^{p+1}(\mathcal{N}_z)$.

□

Lemme C.4. *Soit $\mathcal{S} \subseteq \mathcal{G}$. Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si z n'attaque pas $\mathcal{F}'(\mathcal{S})$, alors $\mathcal{F}'(\mathcal{S}) \subseteq \mathcal{F}(\mathcal{S})$.*

Démonstration du lemme C.4 de la présente page.
Soit $x \in \mathcal{F}'(\mathcal{S})$. Puisque nous sommes dans le contexte de la suppression de l'argument z, nous savons que $x \neq z$. Nous devons prouver que $x \in \mathcal{F}(\mathcal{S})$. Supposons $y \in \mathcal{G}$ un argument qui attaque x dans $\mathcal{G}$. Puisque z n'attaque pas $\mathcal{F}'(\mathcal{S})$, nous savons que $y \neq z$, et donc y attaque x dans $\mathcal{G}'$. Donc il existe un argument $w \in \mathcal{S}$ tel que w attaque y dans $\mathcal{G}'$, donc $w \neq z$ et w attaque également y dans $\mathcal{G}$. Donc $x \in \mathcal{F}(\mathcal{S})$. □

Lemme C.5. *Lors de la suppression d'un argument z dans le cadre de la sémantique basique, si z n'attaque pas $\mathcal{E}'$, alors $\forall k \geq 1,\ \mathcal{F}'^k(\mathcal{N}_z) \subseteq \mathcal{F}^k(\mathcal{N}_z)$.*

Démonstration du lemme C.5 de la présente page.
Prouvons par induction sur $k \geq 1$ que si z n'attaque pas $\mathcal{E}'$, alors $\mathcal{F}'^k(\mathcal{N}_z) \subseteq \mathcal{F}^k(\mathcal{N}_z)$.

- Cas basique ($k = 1$) : puisque z n'attaque pas $\mathcal{E}'$, alors z n'attaque pas $\mathcal{F}'(\mathcal{N}_z)$ (puisque $\mathcal{N}_z \subseteq \mathcal{E}'$ et $\mathcal{F}'(\mathcal{N}_z) \subseteq \mathcal{F}'(\mathcal{E}') = \mathcal{E}'$). Donc, en utilisant le lemme C.4, nous avons $\mathcal{F}'(\mathcal{N}_z) \subseteq \mathcal{F}(\mathcal{N}_z)$.
- Hypothèse d'induction (pour $1 \leq k \leq p$, la proposition est vraie) : nous devons prouver que $\mathcal{F}'^{p+1}(\mathcal{N}_z) \subseteq \mathcal{F}^{p+1}(\mathcal{N}_z)$. Par définition, $\mathcal{F}'^{p+1}(\mathcal{N}_z) = \mathcal{F}'(\mathcal{F}'^p(\mathcal{N}_z))$. En utilisant le lemme C.4, nous avons $\mathcal{F}'(\mathcal{F}'^p(\mathcal{N}_z)) \subseteq \mathcal{F}(\mathcal{F}'^p(\mathcal{N}_z))$ (avec $\mathcal{S} = \mathcal{F}'^p(\mathcal{N}_z)$). En utilisant l'hypothèse d'induction, nous avons $\mathcal{F}'^p(\mathcal{N}_z) \subseteq \mathcal{F}^p(\mathcal{N}_z)$. $\mathcal{F}$ étant monotone, $\mathcal{F}(\mathcal{F}'^p(\mathcal{N}_z)) \subseteq \mathcal{F}(\mathcal{F}^p(\mathcal{N}_z)) = \mathcal{F}^{p+1}(\mathcal{N}_z)$ Ainsi, $\mathcal{F}'^{p+1}(\mathcal{N}_z) \subseteq \mathcal{F}^{p+1}(\mathcal{N}_z)$.

□

Démonstration du lemme 5.2 page 84.

($\Rightarrow$) Montrons que si z n'attaque pas $\mathcal{E}'$ et $z \in \bigcup_{i \geq 1} \mathcal{F}^i(\mathcal{N}_z)$, alors $\mathcal{E}'$ défend z dans $\mathcal{G}$: soit $y \in \mathcal{G}$ un argument qui attaque z dans $\mathcal{G}$. Rappelons que, par la définition 2.6 page 16, un argument ne peut pas s'attaquer lui-même, donc $y \neq z$. En utilisant le lemme C.1 page précédente, puisque z n'attaque pas $\mathcal{E}'$ et donc z n'attaque pas $\mathcal{N}_z$, nous avons $\mathcal{N}_z \subseteq \mathcal{F}(\mathcal{N}_z)$ et donc les $\mathcal{F}^i(\mathcal{N}_z)$ sont emboîtés. Soit i le plus petit indice ≥ 1 tel que $z \in \mathcal{F}^i(\mathcal{N}_z)$, donc $z \notin \mathcal{F}^{i-1}(\mathcal{N}_z)$.

  - $i = 1$: $z \in \mathcal{F}(\mathcal{N}_z)$. Par définition, $\mathcal{N}_z$ défend z, et $\mathcal{N}_z \subseteq \mathcal{E}'$ donc $\mathcal{E}'$ défend z dans $\mathcal{G}$.

- $i > 1$: $z \in \mathcal{F}^i(\mathcal{N}_z) = \mathcal{F}(\mathcal{F}^{i-1}(\mathcal{N}_z))$ et $z \notin \mathcal{F}^{i-1}(\mathcal{N}_z)$. En utilisant le lemme C.3 page 199, nous avons $\mathcal{F}^{i-1}(\mathcal{N}_z) \subseteq \mathcal{F}'^{i-1}(\mathcal{N}_z) = \mathcal{F}'^{i-1}(\mathcal{F}'(\varnothing)) \subseteq \mathcal{E}'$. Ainsi, $z \in \mathcal{F}(\mathcal{E}')$, donc $\mathcal{E}'$ défend z dans $\mathcal{G}$.

($\Leftarrow$) Montrons que si z n'attaque pas $\mathcal{E}'$ et $\mathcal{E}'$ défend z dans $\mathcal{G}$, alors $z \in \bigcup_{i\geq 1} \mathcal{F}^i(\mathcal{N}_z)$:

- Si z n'est pas attaqué par $\mathcal{G}$, alors nous avons trivialement $z \in \bigcup_{i\geq 1} \mathcal{F}^i(\mathcal{N}_z)$.
- Supposons que z est attaqué par $\mathcal{G}$. Par définition, nous avons $\mathcal{E}' = \mathcal{N}_z \cup \bigcup_{i\geq 1} \mathcal{F}'^i(\mathcal{N}_z)$. Ainsi, $\exists i \geq 0$ tel que $\mathcal{F}'^i(\mathcal{N}_z)$ défend z dans $\mathcal{G}$.
  - $i = 0$: $\mathcal{N}_z$ défend z dans $\mathcal{G}$ et donc $z \in \bigcup_{i\geq 1} \mathcal{F}^i(\mathcal{N}_z)$.
  - $i \geq 1$: nous savons que z n'attaque pas $\mathcal{E}'$. En utilisant le lemme C.5 page ci-contre, nous déduisons que $\mathcal{F}'^i(\mathcal{N}_z) \subseteq \mathcal{F}^i(\mathcal{N}_z)$. Puisque $\mathcal{F}'^i(\mathcal{N}_z)$ défend z, $\mathcal{F}^i(\mathcal{N}_z)$ défend z. Donc, $z \in \bigcup_{j\geq 1} \mathcal{F}^j(\mathcal{N}_z)$.

□

Démonstration du lemme 5.3 page 84. Découle du fait que z est le seul argument ajouté ou supprimé ; de plus, par hypothèse, z ne s'attaque pas lui-même. □

Démonstration du lemme 5.4 page 84.

- Lors de l'addition de z, si $x \neq z$ est attaqué par un argument y dans $\mathcal{G}$, alors $y \neq z$ attaque x dans $\mathcal{G}'$. Par ailleurs, si x est attaqué par une argument y dans $\mathcal{G}'$, alors soit $y = z$ donc x n'est pas attaqué dans $\mathcal{G}$ et donc x est seulement attaqué par z, soit $y \neq z$ donc y attaque déjà x dans $\mathcal{G}$.
- La preuve est similaire lors de la suppression de z.

□

C.2.2 Démonstrations concernant la dualité (section 5.2)

Démonstration de la proposition 5.6 page 85.
Découle directement des définitions 4.1 page 60, 5.1 page 85 et 5.2 page 85. □

Démonstration de la proposition 5.7.1 page 85.
D'après la proposition 5.6 page 85, un changement o satisfait la propriété *extensive* si et seulement si son changement inverse o' satisfait le symétrique de la propriété *extensive*. Or, d'après les définitions 5.2 page 85, 4.2 page 62 et 4.8 page 64, le symétrique de la propriété *extensive* est la propriété *restrictive*. Donc un changement est *extensif* si et seulement si le changement inverse est un changement *restrictif*.
□

La démonstration des propositions 5.7.2 à 5.8.4 pages 85–86 se fait de façon analogue à la proposition 5.7.1 page 85.

Démonstration de la proposition 5.8.5 page 86.
Sachant que les différentes propriétés du changement que nous avons définies forment une partition de l'ensemble des changements possibles, et en nous appuyant sur les propositions précédentes, si un changement est *c-modifiant*, le changement inverse l'est aussi. □

Par ailleurs, la démonstration de chacun des items des propositions 5.9 à 5.12 pages 86–87 se fait également de façon analogue à la proposition 5.7.1 page 85.

Démonstration de la proposition 5.13 page 88.
Pour prouver cette proposition, nous allons nous concentrer sur sa contraposée : lors de l'addition d'un argument z dans le cadre de la sémantique basique, $\forall x \in \mathcal{G}$, si $x \in \mathcal{E}'$ et z ne défend pas indirectement x, alors $x \in \mathcal{E}$. Notons en premier lieu que $x \neq z$ puisque $x \in \mathcal{G}$ et que z est ensuite ajouté à $\mathcal{G}$.

Nous savons, grâce à la définition 2.1 page 12, que $\mathcal{R}$ est fini et donc, selon Dung (1995), nous avons $\mathcal{E} = \bigcup_{i\geq 1} \mathcal{F}^i(\varnothing)$ et $\mathcal{E}' = \bigcup_{i\geq 1} \mathcal{F}'^i(\varnothing)$. Prouvons par induction sur $i \geq 1$ que si $x \in \mathcal{F}'^i(\varnothing)$ et $x \neq z$ et z ne défend pas indirectement x, alors $x \in \mathcal{F}^i(\varnothing)$.

- Cas basique ($i = 1$) : si $x \in \mathcal{F}'(\varnothing)$, alors x n'est pas attaqué par $\mathcal{G}'$ et si $x \neq z$, alors $x \in \mathcal{G}$. Donc, d'après le lemme 5.4 page 84, x n'est pas attaqué par $\mathcal{G}$ et donc $x \in \mathcal{F}(\varnothing)$.
- Hypothèse d'induction (pour $1 \leq i \leq p$, la proposition est vraie) : soit $x \in \mathcal{F}'^{p+1}(\varnothing) = \mathcal{F}'(\mathcal{F}'^p(\varnothing))$. Afin de prouver que $x \in \mathcal{F}^{p+1}(\varnothing) = \mathcal{F}(\mathcal{F}^p(\varnothing))$, nous devons prouver que $\mathcal{F}^p(\varnothing)$ défend x dans $\mathcal{G}$. Supposons que x est attaqué par y dans $\mathcal{G}$. Puisque nous sommes dans le contexte de l'addition de l'argument z, d'après le lemme 5.4 page 84, x est attaqué par y dans $\mathcal{G}'$. Puisque $x \in \mathcal{F}'(\mathcal{F}'^p(\varnothing))$, $\mathcal{F}'^p(\varnothing)$ défend x contre y. Donc, il existe un argument $w \in \mathcal{F}'^p(\varnothing)$ tel que w attaque y. Nous savons que z ne défend pas indirectement x, donc $w \neq z$ et z ne défend pas indirectement w. En utilisant l'hypothèse d'induction pour w, nous avons $w \in \mathcal{F}^p(\varnothing)$, donc $\mathcal{F}^p(\varnothing)$ défend x contre y, et donc $x \in \mathcal{F}^{p+1}(\varnothing)$.

□

C.2.3 Démonstrations concernant la caractérisation dans le cas de l'addition d'argument (section 5.3.1)

C.2.3.1 Résultats pour la sémantique basique

Démonstration de la proposition 5.14 page 90.
Voir la démonstration de la proposition 7 de Cayrol *et al.* (2010c). □

Démonstration de la proposition 5.15 page 90.
Voir la démonstration de la proposition 8 de Cayrol *et al.* (2010c). □

Démonstration de la proposition 5.16 page 90.
Voir la démonstration de la proposition 9 de Cayrol *et al.* (2010c). □

Démonstration des propositions 5.17 à 5.19 pages 90–91.
Les propositions 5.17, 5.18.1, 5.18.2, 5.18.3 et 5.19 pages 90–91 ont déjà été prouvées dans Cayrol *et al.* (2010c) sous la condition $\mathcal{E} \neq \varnothing$. Il reste donc seulement à prouver ces propositions sous la condition $\mathcal{E} = \varnothing$:

- (5.17) Si $\mathcal{E} = \varnothing$ alors nous avons trivialement $\mathcal{E} \subseteq \mathcal{E}'$.
- (5.18.1) Si $\mathcal{E}$ ne défend pas z, alors z est attaqué par $\mathcal{G}$. De plus, grâce à la proposition 5.16.1 page 90, nous savons que si $\mathcal{E} = \varnothing$ et z est attaqué par $\mathcal{G}$, alors $\mathcal{E}' = \varnothing$ et donc $\mathcal{E} = \mathcal{E}'$.
- (5.18.2 et 5.18.3) Si $\mathcal{E} = \varnothing$, alors le fait que z n'attaque pas $\mathcal{E}$ est toujours vérifié. De plus, si $\mathcal{E} = \varnothing$ défend z, alors z n'est pas attaqué par $\mathcal{G}$. Ainsi, lorsque l'on ajoute z, nous avons trivialement $\mathcal{E}' = \mathcal{E} \cup \{z\} \cup \bigcup_{i\geq 1} \mathcal{F}'^i(\{z\})$. En outre, si z n'attaque pas $\mathcal{G}$, alors z ne défend aucun argument, et nous avons donc $\mathcal{E}' = \mathcal{E} \cup \{z\}$.
- (5.19) Si $\mathcal{E} = \varnothing$ alors il n'existe aucun argument non attaqué dans $\mathcal{G}$. Ainsi, si nous ajoutons un argument z tel que z est attaqué par $\mathcal{G}$, nous pouvons conclure qu'il n'y a également aucun argument non attaqué dans $\mathcal{G}'$ et $\mathcal{E}' = \varnothing$. De plus, si $\mathcal{E} = \varnothing$ et $\mathcal{E}' = \varnothing$, alors l'argument ajouté z est inévitablement attaqué par $\mathcal{G}$. Et puisqu'il n'y a aucun argument non attaqué dans $\mathcal{G}$, z attaque trivialement tout argument non attaqué de $\mathcal{G}$.

Nous pouvons donc généraliser les précédentes propositions en supprimant la condition $\mathcal{E} \neq \varnothing$. □

Démonstration de la proposition 5.20 page 91.
Si z n'est pas attaqué par $\mathcal{G}$, alors $z \in \mathcal{E}'$. Donc $\bigcup_{i\geq 1} \mathcal{F}'^i(\{z\}) \subseteq \mathcal{E}'$ puisque la fonction $\mathcal{F}'$ est monotone. Puisque z défend indirectement x, $x \in \bigcup_{i\geq 1} \mathcal{F}'^i(\{z\})$, alors $x \in \mathcal{E}'$. □

La démonstration de la proposition 5.21 page 91 nécessite des résultats intermédiaires qui sont donnés dans les lemmes C.6 à C.7 de la présente page.

Lemme C.6. *Lors de l'addition d'un argument, si $\mathcal{S} \subseteq \mathcal{G}$ et $\mathcal{F}'(\mathcal{S}) \subseteq \mathcal{G}$, alors $\mathcal{F}'(\mathcal{S}) \subseteq \mathcal{F}(\mathcal{S})$.*

Démonstration du lemme C.6 de la présente page.
Soit $x \in \mathcal{F}'(\mathcal{S})$. Par hypothèse, $\mathcal{F}'(\mathcal{S}) \subseteq \mathcal{G}$ donc $x \in \mathcal{G}$, et par définition $\mathcal{S}$ défend x dans $\mathcal{G}'$. Supposons que x est attaqué par y dans $\mathcal{G}$. Alors y attaque aussi x dans $\mathcal{G}'$. Donc, $\mathcal{S}$ attaque y dans $\mathcal{G}'$. Mais, $\mathcal{S} \subseteq \mathcal{G}$ et $y \in \mathcal{G}$, donc $\mathcal{S}$ attaque aussi y dans $\mathcal{G}$. Donc, $\mathcal{S}$ défend x dans $\mathcal{G}$, *i.e.* $x \in \mathcal{F}(\mathcal{S})$. □

Lemme C.7. *Lors de l'addition d'un argument z, si $z \notin \mathcal{E}'$, alors $\forall i \geq 1, \mathcal{F}'^i(\varnothing) \subseteq \mathcal{F}^i(\varnothing)$.*

Démonstration du lemme C.7 de la présente page.
Par induction sur $i \geq 1$ et utilisant le lemme C.6. Tout d'abord, notons que si $z \notin \mathcal{E}'$, alors $\mathcal{E}' \subseteq \mathcal{G}$ et donc $\forall i \geq 1, \mathcal{F}'^i(\varnothing) \subseteq \mathcal{G}$.

- Cas basique ($i = 1$) : puisque $z \notin \mathcal{E}'$, z est attaqué par $\mathcal{G}$ dans $\mathcal{G}'$. Les éléments de $\mathcal{F}'(\varnothing)$ ne sont pas égaux à z et par définition ils sont non attaqués dans $\mathcal{G}'$. Puisque qu'aucune attaque n'est enlevée, ils sont aussi non attaqués dans $\mathcal{G}$. Donc $\mathcal{F}'(\varnothing) \subseteq \mathcal{F}(\varnothing)$.
- Hypothèse d'induction (pour $1 \leq i \leq p$, la proposition est vraie) : supposons que $\mathcal{F}'^k(\varnothing) \subseteq \mathcal{F}^k(\varnothing)$ et considérons $\mathcal{F}'^{(k+1)}(\varnothing) = \mathcal{F}'(\mathcal{F}'^k(\varnothing))$. Soit $\mathcal{S} = \mathcal{F}'^k(\varnothing)$. $\mathcal{S} \subseteq \mathcal{G}$, car $z \notin \mathcal{E}'$ et $\mathcal{F}'^k(\varnothing) \subseteq \mathcal{E}'$ donc $z \notin \mathcal{S}$ et $\mathcal{F}'(\mathcal{S}) \subseteq \mathcal{G}$, car $\mathcal{F}'(\mathcal{S}) \subseteq \mathcal{E}'$ et $z \notin \mathcal{E}'$. En utilisant le lemme C.6 page précédente, on obtient $\mathcal{F}'^{(k+1)}(\varnothing) \subseteq \mathcal{F}(\mathcal{F}'^k(\varnothing))$. En utilisant l'hypothèse d'induction, on a $\mathcal{F}'^k(\varnothing) \subseteq \mathcal{F}^k(\varnothing)$. Et, en utilisant la monotonie de $\mathcal{F}$, et la transitivité de l'inclusion d'ensemble, on obtient $\mathcal{F}'^{(k+1)}(\varnothing) \subseteq \mathcal{F}(\mathcal{F}^k(\varnothing))$, *i.e.* $\mathcal{F}'^{(k+1)}(\varnothing) \subseteq \mathcal{F}^{(k+1)}(\varnothing)$.

$\square$

Démonstration de la proposition 5.21 page 91.
Soit $x \in \mathcal{E}$ tel que z attaque x et $\mathcal{E} \setminus \{x\}$ n'attaque pas z. Supposons que $x \in \mathcal{E}' = \bigcup_{i \geq 1} \mathcal{F}'^i(\varnothing)$. Puisque z attaque x, $x \notin \mathcal{F}'(\varnothing)$. Soit i le plus petit indice ≥ 2 tel que $x \in \mathcal{F}'^i(\varnothing)$. Donc $x \in \mathcal{F}'(\mathcal{F}'^{(i-1)}(\varnothing))$ et $x \notin \mathcal{F}'^{(i-1)}(\varnothing)$. z attaque x et donc il existe $y \in \mathcal{F}'^{(i-1)}(\varnothing)$ tel que y attaque z dans $\mathcal{G}'$. Puisque $x \notin \mathcal{F}'^{(i-1)}(\varnothing)$, nous savons que $y \neq x$. De plus, z attaque x et nous supposons que $x \in \mathcal{E}'$, donc $z \notin \mathcal{E}'$ et donc $\mathcal{E}' \subseteq \mathcal{G}$. En utilisant le lemme C.7 page précédente, nous pouvons déduire que $\mathcal{F}'^{(i-1)}(\varnothing) \subseteq \mathcal{F}^{(i-1)}(\varnothing)$. Donc, $y \in \mathcal{E}$. Ainsi, nous avons montré que $y \in \mathcal{E} \setminus \{x\}$ attaque z, ce qui est contradictoire avec l'hypothèse. $\square$

C.2.3.2 Résultats pour la sémantique préférée

Démonstration de la proposition 5.22 page 91.
Voir la démonstration de la proposition 14 de Cayrol *et al.* (2010c) $\square$

Démonstration de la proposition 5.23 page 91.
Voir la démonstration de la proposition 15 de Cayrol *et al.* (2010c) $\square$

Démonstration de la proposition 5.24 page 91.
Voir la démonstration de la proposition 16 de Cayrol *et al.* (2010c) $\square$

Démonstration de la proposition 5.25 page 91.
Voir la démonstration de la proposition 17 de Cayrol *et al.* (2010c) $\square$

Démonstration de la proposition 5.26 page 91.
Voir la démonstration de la proposition 18 de Cayrol *et al.* (2010c) $\square$

Démonstration de la proposition 5.27 page 92.
Voir la démonstration de la proposition 19 de Cayrol *et al.* (2010c) $\square$

Démonstration de la proposition 5.28 page 92.
Voir la démonstration de la proposition 20 de Cayrol *et al.* (2010c) $\square$

Démonstration de la proposition 5.29 page 92.
Voir la démonstration de la proposition 21 de Cayrol *et al.* (2010c) □

C.2.4 Démonstrations concernant la caractérisation dans le cas de la suppression d'argument (section 5.3.2)

C.2.4.1 Résultats pour les trois sémantiques

Démonstration de la proposition 5.30 page 92.

- **Sémantique préférée** : Il suffit de montrer que toute extension $\mathcal{E}$ de $\mathcal{G}$ est admissible dans $\mathcal{G}'$. Soit $\mathcal{E} \in \mathbf{E}$:
  - $\mathcal{E}$ est sans conflit dans $\mathcal{G}$ et donc toujours sans conflit dans $\mathcal{G}'$.
  - Montrons que $\mathcal{E}$ défend ses éléments dans $\mathcal{G}'$.
    Si $x \in \mathcal{E}$ tel que x est attaqué par y dans $\mathcal{G}'$, alors x est également attaqué par y dans $\mathcal{G}$. Or $x \in \mathcal{E}$, donc il est défendu par un argument $t \in \mathcal{E}$ qui attaque y dans $\mathcal{G}$. Comme par hypothèse $z \notin \mathcal{E}$, on sait que $t \neq z$, donc $t \in \mathcal{A}'$ et donc t attaque aussi y dans $\mathcal{G}'$. Ainsi, $\mathcal{E}$ défend x dans $\mathcal{G}'$.

  $\mathcal{E}$ est donc admissible. En conclusion, puisque $\mathcal{E}$ est admissible dans $\mathcal{G}'$, elle est incluse dans une des extensions préférées de $\mathcal{G}'$.

- **Sémantique stable** : Il suffit de montrer que toute extension stable $\mathcal{E}$ de $\mathcal{G}$ est aussi une extension stable dans $\mathcal{G}'$. Soit $\mathcal{E} \in \mathbf{E}$:
  - $\mathcal{E}$ est sans conflit dans $\mathcal{G}$ et donc toujours sans conflit dans $\mathcal{G}'$.
  - Si $y \in \mathcal{A}'$ et $y \notin \mathcal{E}$, alors $y \in \mathcal{A}$ et $y \notin \mathcal{E}$.
    Comme l'extension $\mathcal{E}$ est stable dans $\mathcal{G}$, $\mathcal{E}$ attaque y dans $\mathcal{G}$. Donc, il existe $t \in \mathcal{E}$ tel que t attaque y. Comme par hypothèse, $z \notin \mathcal{E}$, on sait que $z \neq t$ et donc t attaque y dans $\mathcal{G}'$.

  $\mathcal{E}$ est donc stable dans $\mathcal{G}'$.

- **Sémantique basique** :
  Cas où $\mathbf{E} = \{\{\}\}$: On sait que $\mathbf{E}'$ est non vide (puisque nous sommes dans le cadre de la sémantique basique). Donc il existe $\mathcal{E}' \in \mathbf{E}'$. Comme $\mathcal{E} = \varnothing \subseteq \mathcal{E}'$, la proposition est vraie.

  Cas où $\mathbf{E} \neq \{\{\}\}$: Il suffit de montrer que l'extension $\mathcal{E}$ de $\mathcal{G}$ est incluse dans l'extension basique $\mathcal{E}'$ de $\mathcal{G}'$.
  Nous savons, grâce à la définition 2.1 page 12, que la relation binaire $\mathcal{R}$ est finie. Or, d'après Dung (1995), si $\mathcal{R}$ est finie, alors $\mathcal{E} = \bigcup_{i \geq 1} \mathcal{F}^i(\varnothing)$ et $\mathcal{E}' = \bigcup_{i \geq 1} \mathcal{F}'^i(\varnothing)$. Prouvons par induction sur $i \geq 1$ que $\mathcal{F}^i(\varnothing) \subseteq \mathcal{F}'^i(\varnothing)$.
  - $i = 1$: pour tout argument y, si $y \in \mathcal{F}(\varnothing)$ alors y n'est pas attaqué dans $\mathcal{G}$. La suppression de z ne changeant rien à cela, y n'est donc pas attaqué dans $\mathcal{G}'$, et donc $y \in \mathcal{F}'(\varnothing)$.

— Hypothèse d'induction (pour $1 \leq i \leq p, \mathcal{F}^i(\varnothing) \subseteq \mathcal{F}'^i(\varnothing)$) : soit $\mathcal{S} = \mathcal{F}^p(\varnothing)$ et $\mathcal{S}' = \mathcal{F}'^p(\varnothing)$. Tout d'abord, prouvons que $\mathcal{F}(\mathcal{S}) \subseteq \mathcal{F}'(\mathcal{S})$. Soit $y \in \mathcal{F}(\mathcal{S})$. Par définition, $\mathcal{F}(\mathcal{S}) \subseteq \mathcal{E}$, donc $y \in \mathcal{E}$. Si y est attaqué par x dans $\mathcal{G}'$, alors y est attaqué par x dans $\mathcal{G}$. Mais puisque $y \in \mathcal{F}(\mathcal{S})$, $\mathcal{S}$ défend y dans $\mathcal{G}$, donc $\exists t \in \mathcal{S}$ tel que t attaque x dans $\mathcal{G}$.

Par hypothèse, $z \notin \mathcal{E}$, donc $z \notin \mathcal{S}$, donc $t \neq z$ et donc $t \in \mathcal{A}'$. Ainsi, $\mathcal{S}$ défend y dans $\mathcal{G}'$. Donc $y \in \mathcal{F}'(\mathcal{S})$.

Nous venons de montrer que $\mathcal{F}(\mathcal{S}) \subseteq \mathcal{F}'(\mathcal{S})$ et nous avons aussi, par l'hypothèse d'induction, $\mathcal{S} \subseteq \mathcal{S}'$. Sachant que, par définition, $\mathcal{F}'$ est monotone, nous avons $\mathcal{F}(\mathcal{S}) = \mathcal{F}^{p+1}(\varnothing) \subseteq \mathcal{F}'(\mathcal{S}) \subseteq \mathcal{F}'(\mathcal{S}') = \mathcal{F}'^{p+1}(\varnothing)$. Donc, $\mathcal{E} \subseteq \mathcal{E}'$.

□

Démonstration de la proposition 5.31 page 92.

Condition nécessaire : Quelle que soit la sémantique, c'est une conséquence directe de la proposition 5.30 page 92.

Condition suffisante : On prouve la contraposée. Quelle que soit la sémantique, s'il existe une extension $\mathcal{E} \in \mathbf{E}$ telle que $z \in \mathcal{E}$ alors $\forall \mathcal{E}' \in \mathbf{E}'$, $\mathcal{E} \not\subseteq \mathcal{E}'$ car, du fait de la suppression, z n'appartient à aucune des extensions de $\mathcal{G}'$. Donc, $\exists \mathcal{E} \in \mathbf{E}$ telle que $\forall \mathcal{E}' \in \mathbf{E}'$ $\mathcal{E} \not\subseteq \mathcal{E}'$.

□

Démonstration de la proposition 5.32 page 93.

Sémantique préférée : $\mathcal{E}$ est sans conflit dans $\mathcal{G}$ et donc $\mathcal{E} \setminus \{z\}$ est sans conflit dans $\mathcal{G}'$.

Montrons que $\mathcal{E} \setminus \{z\}$ défend ses éléments dans $\mathcal{G}'$. Si $x \in \mathcal{E} \setminus \{z\}$ tel que x est attaqué par y dans $\mathcal{G}'$, alors x est également attaqué par y dans $\mathcal{G}$, or $x \in \mathcal{E}$ donc il est défendu par un argument $t \in \mathcal{E}$ qui attaque y dans $\mathcal{G}$. Comme, par hypothèse, z n'attaque aucun argument de $\mathcal{G}$, on sait que $t \neq z$, donc $t \in \mathcal{E} \setminus \{z\}$ et t attaque y dans $\mathcal{G}'$. Ainsi, $\mathcal{E} \setminus \{z\}$ défend x dans $\mathcal{G}'$.

Sémantique stable : soit un argument y tel que $y \notin \mathcal{E} \setminus \{z\}$, et $y \in \mathcal{G}'$. Alors, $y \neq z$ et donc $y \notin \mathcal{E}$. Or, $\mathcal{E}$ est une extension stable de $\mathcal{G}$, donc $\mathcal{E}$ attaque y dans $\mathcal{G}$. Comme z n'attaque aucun argument de $\mathcal{G}$, $\mathcal{E} \setminus \{z\}$ attaque également y dans $\mathcal{G}$. Or, étant dans le cadre de la suppression de z, $\mathcal{E} \setminus \{z\}$ attaque aussi y dans $\mathcal{G}'$ et donc $\mathcal{E} \setminus \{z\}$ est stable dans $\mathcal{G}'$.

Sémantique basique : nous savons, grâce à la définition 2.1 page 12, que la relation binaire $\mathcal{R}$ est finie. Or, d'après Dung (1995), si $\mathcal{R}$ est finie, alors $\mathcal{E} = \bigcup_{i \geq 1} \mathcal{F}^i(\varnothing)$ et $\mathcal{E}' = \bigcup_{i \geq 1} \mathcal{F}'^i(\varnothing)$. Nous allons tout d'abord montrer, par induction sur i, que $\mathcal{E} \setminus \{z\} \subseteq \mathcal{E}'$, puis que $\mathcal{E}' \subseteq \mathcal{E} \setminus \{z\}$ en montrant que $\mathcal{E} \setminus \{z\}$ est un point fixe de $\mathcal{F}'$.

- Prouvons par induction sur $i \geq 1$ que $\mathcal{F}^i(\varnothing) \setminus \{z\} \subseteq \mathcal{F}'^i(\varnothing)$.
  - $i = 1$: si $y \in \mathcal{F}(\varnothing) \setminus \{z\}$ alors $y \in \mathcal{G}'$ et y n'est pas attaqué dans $\mathcal{G}$, donc y n'est pas attaqué dans $\mathcal{G}'$ et $y \in \mathcal{F}'(\varnothing)$.
  - Hypothèse d'induction (pour $1 \leq i \leq p, \mathcal{F}^i(\varnothing) \setminus \{z\} \subseteq \mathcal{F}'^i(\varnothing)$) : soit $\mathcal{S} = \mathcal{F}^p(\varnothing)$ et $\mathcal{S}' = \mathcal{F}'^p(\varnothing)$. Tout d'abord, prouvons que $\mathcal{F}(\mathcal{S}) \setminus \{z\} \subseteq \mathcal{F}'(\mathcal{S} \setminus \{z\})$.

    Soit $y \in \mathcal{F}(\mathcal{S}) \setminus \{z\}$. Si y est attaqué par x dans $\mathcal{G}'$, alors y est attaqué par x dans $\mathcal{G}$. $y \in \mathcal{F}(\mathcal{S})$, donc $\mathcal{S}$ défend y dans $\mathcal{G}$ et il existe un argument $t \in \mathcal{S}$ tel que t attaque x dans $\mathcal{G}$. Comme z n'attaque aucun argument de $\mathcal{G}$, $t \neq z$ donc $t \in \mathcal{S} \setminus \{z\}$ et donc $\mathcal{S} \setminus \{z\}$ défend y dans $\mathcal{G}'$. Donc $y \in \mathcal{F}'(\mathcal{S} \setminus \{z\})$.

    Nous avons donc $\mathcal{F}(\mathcal{S}) \setminus \{z\} \subseteq \mathcal{F}'(\mathcal{S} \setminus \{z\})$. Par l'hypothèse d'induction, $\mathcal{S} \setminus \{z\} \subseteq \mathcal{S}'$. Sachant que $\mathcal{F}'$ est monotone, nous avons $\mathcal{F}'(\mathcal{S} \setminus \{z\}) \subseteq \mathcal{F}'(\mathcal{S}')$, d'où $\mathcal{F}^{p+1}(\varnothing) \setminus \{z\} = \mathcal{F}(\mathcal{S}) \setminus \{z\} \subseteq \mathcal{F}'(\mathcal{S}') = \mathcal{F}'^{p+1}(\varnothing)$. Donc $\mathcal{E} \setminus \{z\} \subseteq \mathcal{E}'$.

- Prouvons maintenant que $\mathcal{E} \setminus \{z\}$ est un point fixe de $\mathcal{F}'$, c'est-à-dire que $\mathcal{F}'(\mathcal{E} \setminus \{z\}) = \mathcal{E} \setminus \{z\}$.
  - Montrons tout d'abord que $\mathcal{F}'(\mathcal{E} \setminus \{z\}) \subseteq \mathcal{E} \setminus \{z\}$: soit $y \in \mathcal{F}'(\mathcal{E} \setminus \{z\})$. Alors $y \neq z$. Si un argument x attaque y dans $\mathcal{G}$, puisque z n'attaque aucun argument de $\mathcal{G}$, $x \neq z$, donc y est attaqué par x dans $\mathcal{G}'$. Donc il existe un argument $t \in \mathcal{E} \setminus \{z\}$ tel que t attaque x dans $\mathcal{G}'$. Donc t attaque x dans $\mathcal{G}$ et $\mathcal{E}$ défend y dans $\mathcal{G}$. Donc $y \in \mathcal{F}(\mathcal{E}) = \mathcal{E}$ et comme $y \neq z$, $y \in \mathcal{E} \setminus \{z\}$. Donc $\mathcal{F}'(\mathcal{E} \setminus \{z\}) \subseteq \mathcal{E} \setminus \{z\}$
  - Montrons ensuite que $\mathcal{E} \setminus \{z\} \subseteq \mathcal{F}'(\mathcal{E} \setminus \{z\})$: soit $y \in \mathcal{E} \setminus \{z\}$. Si un argument x attaque y dans $\mathcal{G}'$, alors x attaque y dans $\mathcal{G}$. Comme $y \in \mathcal{E} = \mathcal{F}(\mathcal{E})$, $\mathcal{E}$ défend y dans $\mathcal{G}$. Donc il existe un argument $t \in \mathcal{E}$ tel que t attaque x dans $\mathcal{G}$. $t \neq z$ donc $\mathcal{E} \setminus \{z\}$ attaque x dans $\mathcal{G}$ et donc $\mathcal{E} \setminus \{z\}$ défend y dans $\mathcal{G}'$. Ainsi, $y \in \mathcal{F}'(\mathcal{E} \setminus \{z\})$ et donc $\mathcal{E} \setminus \{z\} \subseteq \mathcal{F}'(\mathcal{E} \setminus \{z\})$.

$\mathcal{E}'$ étant par définition le plus petit point fixe de $\mathcal{F}'$, il en découle que $\mathcal{E}' \subseteq \mathcal{E} \setminus \{z\}$. Ainsi, $\mathcal{E}' = \mathcal{E} \setminus \{z\}$.

□

La démonstration de la proposition 5.33 page 93 nécessite l'utilisation du lemme C.8.

Lemme C.8. *Dans le cadre de la suppression d'un argument z en sémantique préférée, si $\mathcal{E}'_j$ est admissible dans $\mathcal{G}'$ et z n'attaque pas $\mathcal{E}'_j$ dans $\mathcal{G}$, alors $\mathcal{E}'_j$ est admissible dans $\mathcal{G}$.*

Démonstration du lemme C.8 de la présente page.
Soit $\mathcal{E}'_j$ un ensemble admissible dans $\mathcal{G}'$. $\mathcal{E}'_j$ reste sans conflit dans $\mathcal{G}$. Soit un argument $y \in \mathcal{E}'_j$ tel qu'il est attaqué par un argument x dans $\mathcal{G}$. Par hypothèse, z

n'attaque pas $\mathcal{E}'_j$ donc $x \neq z$. Donc y est attaqué par x dans $\mathcal{G}'$. $\mathcal{E}'_j$ est admissible donc défend y dans $\mathcal{G}'$. Il défend donc aussi y dans $\mathcal{G}$. □

Démonstration de la proposition 5.33 page 93.

- **Sémantique préférée** : soit $\mathcal{E}$ une extension préférée de $\mathcal{G}$. D'après la proposition 5.32 page 93, il existe une extension $\mathcal{E}'$ de $\mathcal{G}'$ telle que $\mathcal{E} \setminus \{z\} \subseteq \mathcal{E}'$. Or, $z \notin \mathcal{E}$, donc $\mathcal{E} = \mathcal{E} \setminus \{z\} \subseteq \mathcal{E}'$. Par ailleurs, d'après le lemme C.8 page précédente, puisque z n'attaque aucun argument de $\mathcal{G}$, toute extension de $\mathcal{G}'$ est admissible dans $\mathcal{G}$, donc il existe une extension $\mathcal{E}_i$ de $\mathcal{G}$ telle que $\mathcal{E}' \subseteq \mathcal{E}_i$. Donc $\mathcal{E} \subseteq \mathcal{E}' \subseteq \mathcal{E}_i$. Or, $\mathcal{E}$ est un ensemble admissible maximal pour l'inclusion dans $\mathcal{G}$. Donc $\mathcal{E} = \mathcal{E}' = \mathcal{E}_i$. Ainsi, $\mathcal{E}$ est une extension de $\mathcal{G}'$.
- **Sémantiques stable et basique** : découle directement de la proposition 5.32 page 93 et du fait que $\mathcal{E} \setminus \{z\} = \mathcal{E}$.

□

Démonstration de la proposition 5.34 page 93.

- **Sémantique basique** : raisonnons par l'absurde, supposons que z n'appartienne pas à l'extension basique de $\mathcal{G}$. D'après la proposition 5.31 page 92, nous avons $\mathcal{E} \subseteq \mathcal{E}'$, où $\mathcal{E}$ (resp. $\mathcal{E}'$) est l'unique extension basique de $\mathcal{G}$ (resp. $\mathcal{G}'$), ce qui est contradictoire avec la définition du changement *c-limitatif*.
- **Sémantiques préférée et stable** : raisonnons par l'absurde, supposons que z n'appartienne à aucune extension de $\mathcal{G}$. D'après la proposition 5.31 page 92, $\forall \mathcal{E} \in \mathbf{E}$, $\exists \mathcal{E}' \in \mathbf{E}'$, $\mathcal{E} \subseteq \mathcal{E}'$. Or, le changement étant *c-limitatif*, $\mathbf{E} \neq \varnothing$ et $\mathbf{E}' \neq \varnothing$. Soit une extension $\mathcal{E}_i \in \mathbf{E}$, il existe donc une extension $\mathcal{E}'_j \in \mathbf{E}'$ telle que $\mathcal{E}_i \subseteq \mathcal{E}'_j$. Par ailleurs, toujours d'après la définition du changement *c-limitatif*, il existe une extension $\mathcal{E}_k \in \mathbf{E}$ telle que $\mathcal{E}'_j \subset \mathcal{E}_k$. On a donc $\mathcal{E}_i \subset \mathcal{E}_k$.
  — (*Sémantique préférée*) Donc $\mathcal{E}_i$ n'est pas un ensemble admissible maximal et, par là même, n'est pas une extension de $\mathcal{G}$, ce qui contredit notre hypothèse.
  — (*Sémantique stable*) Chaque extension stable étant également préférée, ceci est également impossible en sémantique stable.

□

C.2.4.2 Résultats ne concernant que certaines sémantiques

Démonstration de la proposition 5.35 page 93.
Nous allons tout d'abord montrer que $\forall \mathcal{E}_i \in \mathbf{E}$, si $z \notin \mathcal{E}_i$ (resp. si $z \in \mathcal{E}_i$) alors $\exists \mathcal{E}'_j \in \mathbf{E}'$, $\mathcal{E}_i \subseteq \mathcal{E}'_j$ (resp. $\mathcal{E}_i \setminus \{z\} \subseteq \mathcal{E}'_j$), puis que $\forall \mathcal{E}'_j \in \mathbf{E}'$, si $z \notin \mathcal{E}_i$ (resp. si $z \in \mathcal{E}_i$) alors $\exists \mathcal{E}_i \in \mathbf{E}$, $\mathcal{E}'_j \subseteq \mathcal{E}_i$ (resp. $\mathcal{E}'_j \subseteq \mathcal{E}_i \setminus \{z\}$).

- Soit $\mathcal{E}_i$ une extension préférée de $\mathcal{G}$.

— Si $z \notin \mathcal{E}_i$, d'après la proposition 5.33 page 93, $\mathcal{E}_i$ est une extension de $\mathcal{G}'$.

— Si $z \in \mathcal{E}_i$, d'après la proposition 5.32 page 93, il existe une extension préférée $\mathcal{E}'_j$ de $\mathcal{G}'$ telle que $\mathcal{E}_i \setminus \{z\} \subseteq \mathcal{E}'_j$. $z \in \mathcal{E}_i$, donc $\mathcal{E}_i$ défend z dans $\mathcal{G}$. Mais z n'attaque aucun argument de $\mathcal{G}$ donc $\mathcal{E}_i \setminus \{z\}$ défend z dans $\mathcal{G}$. Donc, en tant que sur-ensemble, $\mathcal{E}'_j$ défend aussi z dans $\mathcal{G}$. D'après le lemme C.8 page 207, $\mathcal{E}'_j$ est admissible dans $\mathcal{G}$. Or, $\mathcal{E}_i \setminus \{z\}$ défend z dans $\mathcal{G}$, donc $\mathcal{E}'_j$ n'attaque pas z dans $\mathcal{G}$. Donc, $\mathcal{E}'_j \cup \{z\}$ est sans conflit et défend ses éléments. Donc $\mathcal{E}'_j \cup \{z\}$ est un ensemble admissible dans $\mathcal{G}$. Alors, il existe une extension préférée $\mathcal{E}_k$ de $\mathcal{G}$ telle que $\mathcal{E}'_j \cup \{z\} \subseteq \mathcal{E}_k$. Ainsi, nous avons $\mathcal{E}_i \setminus \{z\} \subseteq \mathcal{E}'_j$, $z \in \mathcal{E}_i$ et $\mathcal{E}'_j \cup \{z\} \subseteq \mathcal{E}_k$. Donc, $\mathcal{E}_i \subseteq \mathcal{E}'_j \cup \{z\} \subseteq \mathcal{E}_k$. $\mathcal{E}_i$ et $\mathcal{E}_k$ étant des ensembles admissibles maximaux pour l'inclusion, $\mathcal{E}_i = \mathcal{E}_k = \mathcal{E}'_j \cup \{z\}$ et donc $\mathcal{E}'_j = \mathcal{E}_i \setminus \{z\}$.

Cette première partie prouve que $|\mathbf{E}'| \geq |\mathbf{E}|$.

- Soit $\mathcal{E}'_j$ une extension préférée de $\mathcal{G}'$. D'après le lemme C.8 page 207, $\mathcal{E}'_j$ est admissible dans $\mathcal{G}$ donc il existe une extension préférée $\mathcal{E}_i$ de $\mathcal{G}$ telle que $\mathcal{E}'_j \subseteq \mathcal{E}_i$.

  — Si $z \notin \mathcal{E}_i$, d'après la proposition 5.33 page 93, $\mathcal{E}_i$ est une extension de $\mathcal{G}'$, donc $\mathcal{E}'_j = \mathcal{E}_i$ et $\mathcal{E}'_j$ est une extension de $\mathcal{G}$.

  — Si $z \in \mathcal{E}_i$, d'après la première partie de la preuve, $\mathcal{E}_i \setminus \{z\}$ est une extension préférée de $\mathcal{G}'$. On a ainsi $\mathcal{E}'_j \subseteq \mathcal{E}_i$ et $z \notin \mathcal{E}'_j$, donc $\mathcal{E}'_j \subseteq \mathcal{E}_i \setminus \{z\}$. Une extension étant un ensemble admissible maximal pour l'inclusion, on a $\mathcal{E}'_j = \mathcal{E}_i \setminus \{z\}$ et donc $\mathcal{E}_i = \mathcal{E}'_j \cup \{z\}$. $\mathcal{E}'_j \cup \{z\}$ est donc une extension préférée de $\mathcal{G}$.

Cette deuxième partie prouve que $|\mathbf{E}| \geq |\mathbf{E}'|$. Ainsi, $|\mathbf{E}| = |\mathbf{E}'|$.

□

Démonstration de la proposition 5.36 page 93.
Raisonnons par l'absurde : supposons qu'il existe une suppression expansive. On suppose donc que $\mathbf{E} \neq \varnothing$, $|\mathbf{E}| = |\mathbf{E}'|$, et pour toute extension $\mathcal{E}'$ de $\mathcal{G}'$, il existe une extension $\mathcal{E}$ de $\mathcal{G}$ telle que $\mathcal{E} \subset \mathcal{E}'$. Intéressons-nous à une extension quelconque de $\mathcal{G}'$ que nous nommons $\mathcal{E}'_j$; il existe donc une extension $\mathcal{E}_i$ de $\mathcal{G}$ telle que $\mathcal{E}_i \subset \mathcal{E}'_j$. Donc il existe un argument $y \in \mathcal{E}'_j$ tel que $y \notin \mathcal{E}_i$. Notons que $y \in \mathcal{G}$ puisque nous sommes dans le cadre de la suppression d'un argument. $\mathcal{E}_i$ étant stable dans $\mathcal{G}$, il existe un argument $t \in \mathcal{E}_i$ tel que t attaque y dans $\mathcal{G}$. Or, $\mathcal{E}_i \subset \mathcal{E}'_j$, donc $t \in \mathcal{E}'_j$, et, par hypothèse, $y \in \mathcal{E}'_j$. Donc t attaque y dans $\mathcal{G}'$ et donc $\mathcal{E}'_j$ n'est pas sans conflit, ce qui contredit notre hypothèse de départ. □

Démonstration de la proposition 5.37 page 93.

- Le premier point découle directement de la définition du changement expansif et de la proposition 5.31 page 92, aussi bien pour la sémantique préférée que pour la sémantique basique.
- Pour le second point, nous avons deux sémantiques à traiter :
  — **Sémantique préférée** : raisonnons par l'absurde, supposons qu'il existe un changement expansif et que z n'attaque aucun argument de $\mathcal{G}$. Par le premier point de cette preuve, on sait que z n'appartient à aucune extension de $\mathcal{G}$. D'après la proposition 5.33 page 93, on a alors $\forall \mathcal{E}_i$ une extension préférée de $\mathcal{G}$, $\mathcal{E}_i$ est une extension préférée de $\mathcal{G}'$, ce qui contredit la définition du changement expansif.
  — **Sémantique basique** : raisonnons par l'absurde, supposons qu'il existe un changement expansif et que z n'attaque aucun argument de $\mathcal{G}$. D'après le point 1 de la proposition 5.37 page 93, z n'appartient pas à l'extension basique de $\mathcal{G}$ et, par la proposition 5.33 page 93, on a alors $\mathcal{E} = \mathcal{E}'$, où $\mathcal{E}'$ est l'extension basique de $\mathcal{G}'$, ce qui contredit le changement expansif.

□

Démonstration de la proposition 5.38 page 93.

- **Sémantique préférée** : d'après la proposition 5.35 page 93, si z appartient à toute extention préférée de $\mathcal{G}$, les extensions préférées de $\mathcal{G}'$ sont toutes de la forme $\mathcal{E}_i \setminus \{z\}$, où $\mathcal{E}_i$ est une extension préférée de $\mathcal{G}$. D'autre part, $|\mathbf{E}| = |\mathbf{E}'|$, le changement est donc *c-limitatif*.
- **Sémantique basique** : découle directement de la proposition 5.32 page 93.

□

Démonstration de la proposition 5.39 page 93.
L'argument x n'est jamais attaqué dans $\mathcal{G}'$. □

La démonstration de la proposition 5.40 page 94 nécessite l'utilisation du lemme C.9.

Lemme C.9. *Soit $\mathcal{G}$ un graphe d'argumentation et $\mathcal{F}$ sa fonction caractéristique associée. Si $x \in \bigcup_{i \geq 1} \mathcal{F}^i(\varnothing)$, alors pour tout t attaquant de x, il existe un chemin de longueur impaire de $\mathcal{F}(\varnothing)$ à t.*

Démonstration du lemme C.9 de la présente page.
Par induction sur $i \geq 1$, on prouve que, si $x \in \mathcal{F}^i(\varnothing)$, alors pour tout t attaquant de x, il existe un chemin de longueur impaire de $\mathcal{F}(\varnothing)$ à t.

- Pour le cas $i = 1$, le résultat est immédiat car $x \in \mathcal{F}(\varnothing)$ implique que x n'est pas attaqué.

- Pour les autres cas, supposons que la propriété est vraie au rang $p \geq 1$ et que $x \in \mathcal{F}^{p+1}(\varnothing)$. Soit t un attaquant de x. Donc t est attaqué par un argument $w \in \mathcal{F}^p(\varnothing)$. Si w n'est pas attaqué dans $\mathcal{G}$, alors $w \in \mathcal{F}(\varnothing)$, donc il existe un chemin dont la longueur est égale à 1 de $\mathcal{F}(\varnothing)$ à t. Autrement, il existe u attaquant de w. En utilisant l'hypothèse d'induction sur w, il existe un chemin de longueur impaire de $\mathcal{F}(\varnothing)$ à u. Alors, ce chemin est prolongé de deux avec l'attaque de u à w et l'attaque de w à t. Nous obtenons donc un nouveau chemin de longueur impaire de $\mathcal{F}(\varnothing)$ à t.

□

Démonstration de la proposition 5.40 page 94.
Premièrement, notons que $y \neq z$. En effet, par hypothèse, $x \in \mathcal{E}$ et $z \in \mathcal{E}$, donc z ne peut pas attaquer x. Donc $y \in \mathcal{G}'$. De plus, $x \neq z$ est aussi une hypothèse de la proposition. En utilisant un raisonnement par l'absurde, supposons que $x \in \mathcal{E}'$, *i.e.* $x \in \bigcup_{i\geq 1} \mathcal{F'}^i(\varnothing)$. Suivant l'hypothèse sur x, il existe y un attaquant de x dans $\mathcal{G}$ tel que tout chemin de longueur impaire de $\mathcal{F}(\varnothing)$ à y dans $\mathcal{G}$ contient z à une position paire. Notons que $x \in \mathcal{G}'$ et $y \in \mathcal{G}'$. Donc, le lemme C.9 page ci-contre peut être appliqué sur $\mathcal{G}'$ et $\mathcal{F}'$. L'argument y attaquant x dans $\mathcal{G}'$, il existe un chemin de longueur impaire de $\mathcal{F}'(\varnothing)$ à y. Soit $t \in \mathcal{F}'(\varnothing)$ l'argument racine de ce chemin vers y. Ce chemin contient uniquement des arguments de $\mathcal{G}'$, donc il ne peut pas contenir z donc t ne peut pas appartenir à $\mathcal{F}(\varnothing)$. Cela signifie que t est attaqué dans $\mathcal{G}$ et non attaqué dans $\mathcal{G}'$, donc t est uniquement attaqué par z. Ainsi, il existe un chemin de longueur paire de z à y (via t) et donc il existe un chemin de longueur impaire de z à x, ce qui contredit la dernière hypothèse de la proposition. □

Démonstration de la proposition 5.41 page 94.
Tout d'abord, notons que $z \in \mathcal{E}$. En effet, $\mathcal{E} = \mathcal{F}(\mathcal{E})$ donc $x \in \mathcal{F}(\mathcal{E})$ et donc, par hypothèse, $z \in \mathcal{E}$. Donc il n'existe aucune attaque entre z et x. Ainsi, les attaquants de x dans $\mathcal{G}$ sont aussi les attaquants de x dans $\mathcal{G}'$. En utilisant un raisonnement par l'absurde, supposons que $x \in \mathcal{E}'$. Nous allons montrer que $\mathcal{E}'$ défend x dans $\mathcal{G}$. Soit y un attaquant de x dans $\mathcal{G}$. Donc y attaque x dans $\mathcal{G}'$. Puisque $x \in \mathcal{E}'$ et $\mathcal{E}' = \mathcal{F}'(\mathcal{E}')$, alors $\mathcal{E}'$ défend x dans $\mathcal{G}'$. Donc $\mathcal{E}'$ attaque y dans $\mathcal{G}'$ et également dans $\mathcal{G}$. Donc $x \in \mathcal{F}(\mathcal{E}')$. En utilisant l'hypothèse donnée dans la proposition, on obtient $z \in \mathcal{E}'$, ce qui est impossible puisque z a été supprimé. □

C.3 Approche axiomatique (chapitre 7)

Démonstration de la proposition 7.1 page 140.
Grâce à la définition 7.2 page 140, nous savons que $\forall x \in \mathcal{A}_{\mathbf{U}}$, $acc(x)$ est vraie dans $\mathcal{G}$ implique $x \in f_{acc}(\mathcal{G})$. Or, grâce à la définition 7.1 page 139, nous savons que $f_{acc}(\mathcal{G}) \subseteq \mathcal{A}$, donc $x \in \mathcal{A}$. Grâce à la définition 7.2 page 140, nous savons que $arg(x)$ est vraie dans $\mathcal{G}$. □

Démonstration de la proposition 7.2 page 140.
Grâce à la définition 7.2 page 140, nous savons que $\forall x, y \in \mathcal{A}_{\mathbf{U}}, att(x)$ est vraie dans $\mathcal{G}$ implique $(x, y) \in \mathcal{R}$. Or, grâce à la définition 6.2 page 112, nous savons que $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$, donc $x, y \in \mathcal{A}$. Grâce à la définition 7.2 page 140, nous savons que $arg(x)$ et $arg(y)$ sont vraies dans $\mathcal{G}$. $\square$

Démonstration de la proposition 7.3 page 141.
Soit $\mathcal{G} = (\mathcal{A}, \mathcal{R})$. Soit $\mathcal{G}_1 \in \mathbf{G_U}$, $\mathcal{G}_1 = (\mathcal{A}_1, \mathcal{R}_1)$, tel que $\mathcal{G}_1 \in [f_{\mathbf{U}}(\mathcal{G})]$. Prouvons que $\forall x \in \mathcal{A}_{\mathbf{U}}$, $x \in \mathcal{A}$ ssi $x \in \mathcal{A}_1$:

($\Rightarrow$) puisque $x \in \mathcal{A}$, nous avons, d'après la définition 7.3 page 141, $f_{\mathbf{U}}(\mathcal{G}) \models arg(x)$. Or, d'après la notation 7.1 page 139 et puisque par hypothèse nous avons $\mathcal{G}_1 \in [f_{\mathbf{U}}(\mathcal{G})]$, nous savons que $\mathcal{G}_1 \in [arg(x)]$ et donc que $arg(x)$ est vraie dans $\mathcal{G}_1$. Ainsi, d'après la définition 7.2 page 140, $x \in \mathcal{A}_1$.

($\Leftarrow$) prouvons la contraposée ($x \notin \mathcal{A} \implies x \notin \mathcal{A}_1$) : puisque $x \notin \mathcal{A}$, nous avons, d'après la définition 7.3 page 141, $f_{\mathbf{U}}(\mathcal{G}) \models \neg arg(x)$. Or, puisque par hypothèse nous avons $\mathcal{G}_1 \in [f_{\mathbf{U}}(\mathcal{G})]$, nous savons que $\mathcal{G}_1 \in [\neg arg(x)]$ et donc que $\neg arg(x)$ est vraie dans $\mathcal{G}_1$. Ainsi, d'après la définition 7.2 page 140, $x \notin \mathcal{A}_1$.

En procédant de la même manière, nous obtenons $\forall (x, y) \in \mathcal{R}_{\mathbf{U}}$, $(x, y) \in \mathcal{R}$ ssi $(x, y) \in \mathcal{R}_1$. Ainsi, $\mathcal{G}_1$ a les mêmes sommets que $\mathcal{G}$, c'est-à-dire $\mathcal{A}_1 = \mathcal{A}$, ainsi que les mêmes interactions, c'est-à-dire $\mathcal{R}_1 = \mathcal{R}$. Donc $\mathcal{G}_1 = \mathcal{G}$. $\square$

Démonstration de la proposition 7.4 page 146.
$\forall \varphi, \alpha \in \mathscr{L}$, si $\exists \mathcal{G}' \in [\varphi \diamondsuit_{\mathcal{T}} \alpha]$, alors, en raison de **E3**, $(\varphi, \alpha) \models \mathcal{T}$, c'est-à-dire $[\varphi] \neq \varnothing$ et $\forall \mathcal{G} \in [\varphi], \exists \mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$. Ainsi, en utilisant **E8**, $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \bigcup_{\mathcal{G} \in [\varphi]} [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha]$. Donc, il existe un $\mathcal{G} \in [\varphi]$ tel que $\mathcal{G}' \in [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha]$. Par **E5**, $(f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha) \wedge f(\mathcal{G}') \models f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (\alpha \wedge f(\mathcal{G}'))$. Donc $\mathcal{G}' \in [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (\alpha \wedge f(\mathcal{G}'))]$. Par **E3**, $(f(\mathcal{G}), (\alpha \wedge f(\mathcal{G}'))) \models \mathcal{T}$, donc $[\alpha \wedge f(\mathcal{G}')] \neq \varnothing$. Ainsi $\mathcal{G}' \in [\alpha]$. Et donc, **U1** est satisfait. $\square$

Démonstration du théorème 7.2 page 147.
Cette preuve est similaire à celle faite par Katsuno et Mendelzon à l'exception du fait qu'elle ne dépend pas du postulat **U2** et utilise les versions restreintes de **E3**, **E5** et **E8**. De plus, elle prend en compte l'ensemble $\mathcal{T}$. Notons que cette preuve utilise la proposition 7.7 page 147, laquelle est définie plus loin (mais repose uniquement sur **E3**).

1. Soit $\diamondsuit_{\mathcal{T}} : \mathscr{L} \times \mathscr{L} \to \mathscr{L}$ un opérateur satisfaisant **E3**, **U4**, **E5**, **E8** et **U9**. Pour tout $\mathcal{G} \in \mathbf{G_U}$, définissons $\preccurlyeq_{\mathcal{G}}$ tel que $\mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_2$ ssi $\mathcal{G}_1 \in [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (f(\mathcal{G}_1) \vee f(\mathcal{G}_2))]$ ou $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (f(\mathcal{G}_1) \vee f(\mathcal{G}_2))] = \varnothing$.
   Nous pouvons montrer que $\preccurlyeq_{\mathcal{G}}$ est un préordre complet et que cette assignation respecte $\mathcal{T}$. Dans la suite de cette preuve, nous abrégeons $f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (f(\mathcal{G}_x) \vee f(\mathcal{G}_y))$ par $\diamondsuit(x, y)$ et $f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (f(\mathcal{G}_1) \vee f(\mathcal{G}_2) \vee f(\mathcal{G}_3))$ par $\diamondsuit(1, 2, 3)$. D'après la proposition 7.4 page 146, $\diamondsuit_{\mathcal{T}}$ satisfait **U1**.
   - **Transitivité** : considérons $\mathcal{G}, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3 \in \mathbf{G_U}$ tels que $\mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_2$ et $\mathcal{G}_2 \preccurlyeq_{\mathcal{G}} \mathcal{G}_3$; il y a quatre cas :

— $\mathcal{G}_1 \in [\Diamond(1,2)]$ et $\mathcal{G}_2 \in [\Diamond(2,3)]$: en raison de la proposition 7.7 page 147, nous avons $\Diamond(1,2,3) \neq \varnothing$. Ainsi, par **U1**, $[\Diamond(1,2,3)] \subseteq \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3\}$.

  — si $[\Diamond(1,2,3) \wedge (f(\mathcal{G}_1) \vee f(\mathcal{G}_2))] = \varnothing$, alors $[\Diamond(1,2,3)] = \{\mathcal{G}_3\}$, ce qui signifie que $[\Diamond(1,2,3) \wedge (f(\mathcal{G}_2) \vee f(\mathcal{G}_3))] \neq \varnothing$, donc en utilisant à la fois **U9** et **E5**, $[\Diamond(1,2,3) \wedge (f(\mathcal{G}_2) \vee f(\mathcal{G}_3))] = [\Diamond(2,3)]$ donc $[\Diamond(2,3)] = \{\mathcal{G}_3\}$, ce qui est en contradiction avec l'hypothèse statuant que $\mathcal{G}_2 \in [\Diamond(2,3)]$; ce cas ne peut donc pas se produire.

  — si $[\Diamond(1,2,3) \wedge (f(\mathcal{G}_1) \vee f(\mathcal{G}_2))] \neq \varnothing$, alors en utilisant **E5** et **U9**, $[\Diamond(1,2)] = [\Diamond(1,2,3) \wedge (f(\mathcal{G}_1) \vee f(\mathcal{G}_2))]$. Ainsi, $\mathcal{G}_1 \in [\Diamond(1,2,3)]$. En raison de **E5** et **U9** encore une fois, $[\Diamond(1,3)] = [\Diamond(1,2,3) \wedge (f(\mathcal{G}_1) \vee f(\mathcal{G}_3))]$, et donc $\mathcal{G}_1 \in [\Diamond(1,3)]$, donc $\mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_3$.

— $[\Diamond(1,2)] = \varnothing$ et $\mathcal{G}_2 \in [\Diamond(2,3)]$: en raison de la proposition 7.7 page 147 (plus précisément, en raison de la contraposée de sa conclusion), nous avons $[f(\mathcal{G})\Diamond_{\mathcal{T}} f(\mathcal{G}_2)] = \varnothing$. Et, grâce à **E5**, $\Diamond(2,3) \wedge f(\mathcal{G}_2) \models f(\mathcal{G})\Diamond_{\mathcal{T}} f(\mathcal{G}_2)$. Ainsi, $\mathcal{G}_2 \notin [\Diamond(2,3)]$ ce qui cause une contradiction ; ce cas ne se produit donc jamais.

— $\mathcal{G}_1 \in [\Diamond(1,2)]$ et $[\Diamond(2,3)] = \varnothing$: en utilisant la proposition 7.7 page 147, nous avons $[\Diamond(1,2,3)] \neq \varnothing$, et grâce à **U1**, $[\Diamond(1,2,3)] \subseteq \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3\}$. En utilisant **E5** et **U9**, nous avons comme précédemment $\mathcal{G}_2 \notin [\Diamond(1,2,3)]$ et $\mathcal{G}_3 \notin [\Diamond(1,2,3)]$. Ainsi, $[\Diamond(1,2,3) \wedge (f(\mathcal{G}_1) \vee f(\mathcal{G}_2))]$ doit contenir $\mathcal{G}_1$. Grâce à **E5**, nous avons $\mathcal{G}_1 \in [\Diamond(1,3)]$, donc $\mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_3$.

— $[\Diamond(1,2)] = \varnothing$ et $[\Diamond(2,3)] = \varnothing$: grâce à la proposition 7.7 page 147, nous avons $[f(\mathcal{G})\Diamond_{\mathcal{T}} f(\mathcal{G}_1)] = \varnothing$ et $[f(\mathcal{G})\Diamond_{\mathcal{T}} f(\mathcal{G}_3)] = \varnothing$. En utilisant **E5**, nous avons $[\Diamond(1,3)] = \varnothing$, donc $\mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_3$.

- **Réflexivité** : $\forall \mathcal{G}, \mathcal{G}_1 \in \mathbf{G_U}$, en utilisant **U1**, $f(\mathcal{G})\Diamond_{\mathcal{T}} f(\mathcal{G}_1) \models f(\mathcal{G}_1)$, donc soit $\mathcal{G}_1 \in [f(\mathcal{G})\Diamond_{\mathcal{T}} f(\mathcal{G}_1)]$, soit $[f(\mathcal{G})\Diamond_{\mathcal{T}} f(\mathcal{G}_1)] = \varnothing$. Ainsi $\mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_1$.
- **Complétude** : $\forall \mathcal{G}, \mathcal{G}_1, \mathcal{G}_2 \in \mathbf{G_U}$, si $\mathcal{G}_1 \not\preccurlyeq \mathcal{G}_2$, alors $[f(\mathcal{G})\Diamond_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2))] \neq \varnothing$ et $\mathcal{G}_1 \notin [f(\mathcal{G})\Diamond_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2))]$. En raison de **U1**, cela signifie que $\mathcal{G}_2 \in [f(\mathcal{G})\Diamond_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2))]$. Ainsi, $\mathcal{G}_2 \preccurlyeq_{\mathcal{G}} \mathcal{G}_1$.
- L'assignation **respecte** $\mathcal{T}$: si $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et $(\mathcal{G}, \mathcal{G}_2) \notin \mathcal{T}$ et $\mathcal{G}_2 \preccurlyeq_{\mathcal{G}} \mathcal{G}_1$, cela signifie que soit $[f(\mathcal{G})\Diamond_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2))] = \varnothing$, soit $\mathcal{G}_2 \in [f(\mathcal{G})\Diamond_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2))]$. Grâce à **E3**, puisque $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$, nous avons $[f(\mathcal{G})\Diamond_{\mathcal{T}} f(\mathcal{G}_1)] \neq \varnothing$. Grâce à **U1**, $\mathcal{G}_1 \in [f(\mathcal{G})\Diamond_{\mathcal{T}} f(\mathcal{G}_1)]$. Donc, en utilisant **E5** et **U9**, $[f(\mathcal{G})\Diamond_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2)) \wedge f(\mathcal{G}_1)] = [f(\mathcal{G})\Diamond_{\mathcal{T}} f(\mathcal{G}_1)] = \{\mathcal{G}_1\}$. Contradiction, donc $\mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_2$.

Montrons maintenant que $\forall \mathcal{G} \in \mathbf{G_U}, \forall \alpha \in \mathscr{L}, [f(\mathcal{G})\Diamond_{\mathcal{T}} \alpha] = \{\mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et $\forall \mathcal{G}_2 \in [\alpha]$, tel que $(\mathcal{G}, \mathcal{G}_2) \in \mathcal{T}$, $\mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_2\}$.

- Si $\mathcal{G}_1 \in [f(\mathcal{G})\Diamond_{\mathcal{T}} \alpha]$, grâce à **U1**, $\mathcal{G}_1 \in [\alpha]$. Si $\exists \mathcal{G}_2 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_2) \in \mathcal{T}$ et $\mathcal{G}_2 \prec_{\mathcal{G}} \mathcal{G}_1$, cela signifie que $\mathcal{G}_1 \notin [f(\mathcal{G})\Diamond_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2))]$. En raison de **E5**, $(f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha) \wedge (f(\mathcal{G}_1) \vee f(\mathcal{G}_2)) \models f(\mathcal{G})\diamondsuit_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2))$, ce qui implique $\mathcal{G}_1 \in [f(\mathcal{G})\diamondsuit_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2))]$, donc $\mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_2$. De plus, grâce à **E5**, $[(f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha) \wedge f(\mathcal{G}_1)] \subseteq [f(\mathcal{G})\diamondsuit_{\mathcal{T}} f(\mathcal{G}_1)]$, donc $\mathcal{G}_1 \in [f(\mathcal{G})\diamondsuit_{\mathcal{T}} f(\mathcal{G}_1)]$, et donc $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$. Ainsi, $[f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha] \subseteq \{\mathcal{G}_1 \in [\alpha], (\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et $\forall \mathcal{G}_2 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_2) \in \mathcal{T}, \mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_2\}$.

- Réciproquement, soit $\mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et $\forall \mathcal{G}_2 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_2) \in \mathcal{T}, \mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_2$. Considérons premièrement $\mathcal{G}_2 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_2) \notin \mathcal{T}$. En raison de **E5**, $[(f(\mathcal{G})\diamondsuit_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2))) \wedge f(\mathcal{G}_2)] \subseteq [f(\mathcal{G})\diamondsuit_{\mathcal{T}} f(\mathcal{G}_2)]$ et, puisque $(\mathcal{G}, \mathcal{G}_2) \notin \mathcal{T}$, en raison de **E3**, $[f(\mathcal{G})\diamondsuit_{\mathcal{T}} f(\mathcal{G}_2)] = \varnothing$. De plus, puisque $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$, en utilisant **E3** nous avons $[f(\mathcal{G})\diamondsuit_{\mathcal{T}} (f(\mathcal{G}_1) \vee f(\mathcal{G}_2))] \neq \varnothing$. Grâce à **U1**, $[f(\mathcal{G}) \diamond (f(\mathcal{G}_1) \vee f(\mathcal{G}_2))] \subseteq \{\mathcal{G}_1, \mathcal{G}_2\}$. Ainsi, $[f(\mathcal{G})\diamondsuit_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2))] = \{\mathcal{G}_1\}$; ce résultat, que nous notons (**a**), est satisfait pour tout $\mathcal{G}_2 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_2) \notin \mathcal{T}$.

  En outre, en raison de notre hypothèse à propos de $\mathcal{G}_1$, $\forall \mathcal{G}_2 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_2) \in \mathcal{T}$, si $[f(\mathcal{G})\diamondsuit_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2))] \neq \varnothing$, alors $\mathcal{G}_1 \in [f(\mathcal{G})\diamondsuit_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_2))]$. Appelons ce résultat (**b**).

  Notons, sans perte de généralité, $\{\mathcal{G}_1, \ldots, \mathcal{G}_n\} = [\alpha]$ l'ensemble des graphes dans lesquels α est satisfaite (cet ensemble est fini puisque $\mathcal{A}_{\mathbf{U}}$ et $\mathcal{R}_{\mathbf{U}}$ sont finis). Grâce à **E3** et **U1**, puisque $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et $\mathcal{G}_1 \in [\alpha]$, nous avons $[(f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha)] \neq \varnothing$ et $[(f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha)] \subseteq [\alpha] = \{\mathcal{G}_1, \ldots, \mathcal{G}_n\}$. Ainsi, pour un certain $k \in [1, n]$, on a $[(f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha) \wedge (f(\mathcal{G}_1) \vee f(\mathcal{G}_k))] \neq \varnothing$. Remarquons que $\forall i \in [2, n]$, $[\alpha \wedge (f(\mathcal{G}_1) \vee f(\mathcal{G}_i))] = [f(\mathcal{G}_1) \vee f(\mathcal{G}_i)]$. Par **U4**, nous déduisons que $[f(\mathcal{G})\diamondsuit_{\mathcal{T}}(\alpha \wedge (f(\mathcal{G}_1) \vee f(\mathcal{G}_i)))] = [f(\mathcal{G})\diamondsuit_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_i))]$.
  Par définition de k, nous pouvons appliquer à la fois **E5** et **U9**, ce qui nous donne $[f(\mathcal{G})\diamondsuit_{\mathcal{T}}(f(\mathcal{G}_1) \vee f(\mathcal{G}_k))] = [(f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha) \wedge (f(\mathcal{G}_1) \vee f(\mathcal{G}_k))] \neq \varnothing$.
  Si $(\mathcal{G}, \mathcal{G}_k) \notin \mathcal{T}$, alors, grâce à (**a**), nous avons $[(f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha) \wedge (f(\mathcal{G}_1) \vee f(\mathcal{G}_k))] = \{\mathcal{G}_1\}$. Si $(\mathcal{G}, \mathcal{G}_k) \in \mathcal{T}$, alors, grâce à (**b**), nous avons $\mathcal{G}_1 \in [(f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha) \wedge (f(\mathcal{G}_1) \vee f(\mathcal{G}_k))]$. Dans les deux cas, $\mathcal{G}_1 \in [f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha]$.

Ainsi, si $[\varphi] = \varnothing$, alors $\bigcup_{\mathcal{G} \in [\varphi]}[f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha] = \varnothing$. Sinon, si $\exists \mathcal{G} \in [\varphi]$ tel que $[f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha] = \varnothing$, alors $[\varphi\diamondsuit_{\mathcal{T}}\alpha] = \varnothing$, sinon nous avons $[\varphi\diamondsuit_{\mathcal{T}}\alpha] = \bigcup_{\mathcal{G} \in [\varphi]}[f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha]$. D'où le résultat.

2. Soit une assignation $\preccurlyeq_{\mathcal{G}}$ respectant $\mathcal{T}$ qui à tout $\mathcal{G}$ associe un préordre complet sur $\mathbf{G_U} \times \mathbf{G_U}$, soit $\diamondsuit_{\mathcal{T}}$ défini par $\forall \mathcal{G} \in \mathbf{G_U}$, $[f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha] = \{\mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et $\forall \mathcal{G}_2 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_2) \in \mathcal{T}, \mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_2\}$ et $[\varphi\diamondsuit_{\mathcal{T}}\alpha] = \varnothing$ si $\exists \mathcal{G} \in [\varphi]$ tel que $[f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha] = \varnothing$. Sinon $[\varphi\diamondsuit_{\mathcal{T}}\alpha] = \bigcup_{\mathcal{G} \in [\varphi]}[f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha]$. Montrons que $\diamondsuit_{\mathcal{T}}$ satisfait **E3**, **U4**, **E5**, **E8** et **U9** :
   - **E3** :
     (⇒) $[\varphi\diamondsuit_{\mathcal{T}}\alpha] \neq \varnothing$ ssi $[\varphi] \neq \varnothing$ et $\forall \mathcal{G} \in [\varphi]$, $[f(\mathcal{G})\diamondsuit_{\mathcal{T}}\alpha] \neq \varnothing$. Ce qui implique que $(\varphi, \alpha) \models \mathcal{T}$ puisque cela est équivalent à $[\varphi] \neq \varnothing$ et $\forall \mathcal{G} \in [\varphi]$, $\exists \mathcal{G}' \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}') \in \mathcal{T}$.

(⇐) si $(\varphi, \alpha) \models \mathcal{T}$, alors $[\varphi] \neq \varnothing$ et $\forall \mathcal{G} \in [\varphi], \exists \mathcal{G}' \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}') \in \mathcal{T}$. La complétude de $\preccurlyeq_{\mathcal{G}}$ signifie que $\exists \mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et $\forall \mathcal{G}' \in [\alpha], \mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}'$. Par définition, cela signifie que $[\varphi \diamondsuit_{\mathcal{T}} \alpha] \neq \varnothing$.

- **U4** : puisque la définition de $\diamondsuit_{\mathcal{T}}$ est uniquement basée sur les modèles de φ et α, il est aisé de vérifier que **U4** est satisfait.
- **E5** : si $[(f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha) \wedge \beta] \neq \varnothing$, alors $\exists \mathcal{G}_a \in [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] \cap [\beta]$. Donc, $\mathcal{G}_a \in [\beta]$ et $\mathcal{G}_a \in [\alpha]$ et $(\mathcal{G}, \mathcal{G}_a) \in \mathcal{T}$ et $\forall \mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$, $\mathcal{G}_a \preccurlyeq_{\mathcal{G}} \mathcal{G}_1$. Ainsi, $\forall \mathcal{G}_2 \in [\alpha \wedge \beta]$ tel que $(\mathcal{G}, \mathcal{G}_2) \in \mathcal{T}$, $\mathcal{G}_2$ étant dans $[\alpha]$, cela signifie que $\mathcal{G}_a \preccurlyeq_{\mathcal{G}} \mathcal{G}_2$. Ainsi, $\mathcal{G}_a \in [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (\alpha \wedge \beta)]$.
- **E8** : par définition.
- **U9** : si $[(f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha) \wedge \beta] \neq \varnothing$, alors soit $\mathcal{G}_b \in [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (\alpha \wedge \beta)]$, alors $\mathcal{G}_b \in [\alpha \wedge \beta]$ et $(\mathcal{G}, \mathcal{G}_b) \in \mathcal{T}$ et $\forall \mathcal{G}_1 \in [\alpha \wedge \beta]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$, $\mathcal{G}_b \preccurlyeq_{\mathcal{G}} \mathcal{G}_1$. Soit $\mathcal{G}_a \in [(f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha) \wedge \beta]$, ce qui signifie que $\mathcal{G}_a \in [\alpha \wedge \beta]$ et $(\mathcal{G}, \mathcal{G}_a) \in \mathcal{T}$, donc $\mathcal{G}_b \preccurlyeq_{\mathcal{G}} \mathcal{G}_a$. De plus, par définition de $\mathcal{G}_a$, $\forall \mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$, $\mathcal{G}_a \preccurlyeq_{\mathcal{G}} \mathcal{G}_1$. Ainsi, par transitivité, $\forall \mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$, $\mathcal{G}_b \preccurlyeq_{\mathcal{G}} \mathcal{G}_1$. Ainsi, $\mathcal{G}_b \in [(f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha) \wedge \beta]$.

□

Démonstration de la proposition 7.5 page 147.
Rappelons l'ensemble de postulats : $\forall \varphi, \psi, \alpha, \beta \in \mathscr{L}$,

E3 : $[\varphi \diamondsuit_{\mathcal{T}} \alpha] \neq \varnothing$ ssi $(\varphi, \alpha) \models \mathcal{T}$.

U4 : $[\varphi] = [\psi]$ et $[\alpha] = [\beta] \implies [\varphi \diamond \alpha] = [\psi \diamond \beta]$.

E5 : si $|[\varphi]| = 1$, alors $(\varphi \diamondsuit_{\mathcal{T}} \alpha) \wedge \beta \models \varphi \diamondsuit_{\mathcal{T}} (\alpha \wedge \beta)$.

E8 : si $([\varphi] \neq \varnothing$ et $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing)$ ou $([\psi] \neq \varnothing$ et $[\psi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing)$,
alors $[(\varphi \vee \psi) \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$
sinon $[(\varphi \vee \psi) \diamondsuit_{\mathcal{T}} \alpha] = [(\varphi \diamondsuit_{\mathcal{T}} \alpha) \vee (\psi \diamondsuit_{\mathcal{T}} \alpha)]$

U9 : si $|[\varphi]| = 1$, alors $[(\varphi \diamondsuit_{\mathcal{T}} \alpha) \wedge \beta] \neq \varnothing \implies \varphi \diamondsuit_{\mathcal{T}} (\alpha \wedge \beta) \models (\varphi \diamondsuit_{\mathcal{T}} \alpha) \wedge \beta$.

De façon à prouver ce résultat, nous devons fournir pour chaque postulat un opérateur satisfaisant tous les autres postulats sauf lui-même.

Indépendance de E3 : soit $\diamondsuit_{\mathcal{T}}$ tel que si $(f(\mathcal{G}), \alpha) \not\models \mathcal{T}$ et $[\alpha] \neq \varnothing$, alors $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] = [\alpha]$, sinon $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$, et si $([\varphi] \neq \varnothing$ et $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing)$ ou $([\psi] \neq \varnothing$ et $[\psi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing)$, alors $[(\varphi \vee \psi) \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$, sinon $[(\varphi \vee \psi) \diamondsuit_{\mathcal{T}} \alpha] = [(\varphi \diamondsuit_{\mathcal{T}} \alpha) \vee (\psi \diamondsuit_{\mathcal{T}} \alpha)]$.

- **E3** n'est pas satisfait.
- **U4** est satisfait : puisque nous définissons $\diamondsuit_{\mathcal{T}}$ à partir des modèles de φ et α uniquement.
- **E5** est satisfait : puisque si $(f(\mathcal{G}), \alpha) \models \mathcal{T}$, alors $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$ (et donc **U5** est satisfait), sinon $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] = [\alpha]$ et $(f(\mathcal{G}), \alpha) \not\models \mathcal{T}$, ce qui implique $(f(\mathcal{G}), \alpha \wedge \beta) \not\models \mathcal{T}$, donc $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (\alpha \wedge \beta)] = [\alpha \wedge \beta]$. Ainsi, **E5** est satisfait.

- **E8** est satisfait : par construction.
- **U9** est satisfait : si $[(\varphi\Diamond_{\mathcal{T}}\alpha) \wedge \beta] \neq \varnothing$, alors cela signifie que $[(\varphi\Diamond_{\mathcal{T}}\alpha)] \neq \varnothing$, donc $(\varphi, \alpha) \not\models \mathcal{T}$, c'est-à-dire $[(\varphi\Diamond_{\mathcal{T}}\alpha)] = [\alpha]$, donc $[(\varphi\Diamond_{\mathcal{T}}\alpha) \wedge \beta] = [\alpha \wedge \beta]$. De plus, $(\varphi, \alpha) \not\models \mathcal{T}$ implique $(\varphi, \alpha \wedge \beta) \not\models \mathcal{T}$. Cela signifie que $[\varphi\Diamond_{\mathcal{T}}(\alpha \wedge \beta)] = [\alpha \wedge \beta]$. D'où le résultat.

Indépendance de U4 : étant donné une formule $\alpha \in \mathscr{L}$ et un graphe $\mathcal{G} \in \mathbf{G_U}$, notons par $r_{\mathcal{G}}(\alpha)$ le rang du premier littéral dans α qui est satisfait dans $\mathcal{G}$. Soit $\Diamond_{\mathcal{T}}$ tel que $\forall \mathcal{G} \in \mathbf{G_U}$, $[f(\mathcal{G})\Diamond_{\mathcal{T}}\alpha] = \{\mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et pour tout autre graphe $\mathcal{G}_2 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_2) \in \mathcal{T}$, $r_{\mathcal{G}_2}(\alpha) \geq r_{\mathcal{G}_1}(\alpha)\}$. Et $[\varphi\Diamond_{\mathcal{T}}\alpha] = \varnothing$ si $\exists \mathcal{G} \in [\varphi]$ tel que $[f(\mathcal{G})\Diamond_{\mathcal{T}}\alpha] = \varnothing$, sinon $[\varphi\Diamond_{\mathcal{T}}\alpha] = \bigcup_{\mathcal{G}\in[\varphi]}[f(\mathcal{G})\Diamond_{\mathcal{T}}\alpha]$.

- **E3** est satisfait : $(\varphi, \alpha) \models \mathcal{T} \iff [\varphi] \neq \varnothing$ et $\forall \mathcal{G} \in [\varphi], \exists \mathcal{G}' \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}') \in \mathcal{T} \iff [\varphi\Diamond_{\mathcal{T}}\alpha] \neq \varnothing$.
- **U4** n'est pas satisfait : considérons $\mathscr{L}$ basé sur $\mathbf{U} = (\{a, b\}, \{(a, b), (b, a)\})$ et trois graphes $\mathcal{G}, \mathcal{G}_1, \mathcal{G}_2$ tels que $f(\mathcal{G}_1) = arg(a) \wedge \neg arg(b) \wedge \neg att(a, b) \wedge \neg att(b, a)$ et $f(\mathcal{G}_2) = \neg arg(a) \wedge arg(b) \wedge \neg att(a, b) \wedge \neg att(b, a)$; supposons que $\mathcal{T}$ contient au moins les transitions $(\mathcal{G}, \mathcal{G}_1)$ et $(\mathcal{G}, \mathcal{G}_2)$. Si nous prenons $\alpha = f(\mathcal{G}_1) \vee f(\mathcal{G}_2)$ et $\beta = f(\mathcal{G}_2) \vee f(\mathcal{G}_1)$, alors cela nous mène à $[f(\mathcal{G})\Diamond_{\mathcal{T}}\alpha] = \{\mathcal{G}_1\}$ et $[f(\mathcal{G})\Diamond_{\mathcal{T}}\beta] = \{\mathcal{G}_2\}$.
- **E5** et **U9** sont satisfaits puisque l'ordre des littéraux n'est pas changé dans la conjonction.
- **E8** est satisfait par définition.

Indépendance de E5 : soit $\Diamond_{\mathcal{T}}$ tel que $\forall \varphi, \alpha \in \mathscr{L}$, $\exists \mathcal{G}_0 \in \mathbf{G_U}$ tel que si $(\varphi, \alpha) \models \mathcal{T}$, alors $[\varphi\Diamond_{\mathcal{T}}\alpha] = \{\mathcal{G}_0\}$ sinon $[\varphi\Diamond_{\mathcal{T}}\alpha] = \varnothing$, et $\mathbf{G_U}$ contient au moins deux graphes $\mathcal{G}_0$ et $\mathcal{G}_1$, et $\mathcal{T}$ contient au moins une transition vers G_1.

- **E3** est satisfait.
- **U4** est satisfait.
- **E5** n'est pas satisfait : puisque $\mathcal{G}_1 \neq \mathcal{G}_0$ et $\exists \mathcal{G} \in \mathbf{G_U}$ tel que$(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$, nous avons $[f(\mathcal{G})\Diamond_{\mathcal{T}}f(\mathcal{G}_1)] = \{\mathcal{G}_0\}$, donc $[(f(\mathcal{G})\Diamond_{\mathcal{T}}f(\mathcal{G}_1)) \wedge f(\mathcal{G}_0)] = \{\mathcal{G}_0\}$, tandis que $[f(\mathcal{G})\Diamond_{\mathcal{T}}(f(\mathcal{G}_1) \wedge f(\mathcal{G}_0))] = \varnothing$ (puisque $[f(\mathcal{G}_1) \wedge f(\mathcal{G}_0)] = \varnothing$, on a $(f(\mathcal{G}), f(\mathcal{G}_1) \wedge f(\mathcal{G}_0)) \not\models \mathcal{T}$).
- **E8** est satisfait : si $\varphi \neq \varnothing$ et $[\varphi\Diamond_{\mathcal{T}}\alpha] = \varnothing$, alors $(\varphi, \alpha) \not\models \mathcal{T}$, ce qui implique $(\varphi \vee \psi, \alpha) \not\models \mathcal{T}$, donc $[(\varphi \vee \psi)\Diamond_{\mathcal{T}}\alpha] = \varnothing$, et donc **E8** est satisfait. Sinon $(\varphi, \alpha) \models \mathcal{T}$ et $(\psi, \alpha) \models \mathcal{T}$, donc $(\varphi \vee \psi, \alpha) \models \mathcal{T}$, ce qui signifie que $[\varphi \vee \psi\Diamond_{\mathcal{T}}\alpha] = \{\mathcal{G}_0\} = [\varphi\Diamond_{\mathcal{T}}\alpha] = [\psi\Diamond_{\mathcal{T}}\alpha]$.
- **U9** est satisfait : si $[(\varphi\Diamond_{\mathcal{T}}\alpha) \wedge \beta] \neq \varnothing$, alors cela signifie que $[\varphi\Diamond_{\mathcal{T}}\alpha] \neq \varnothing$, donc $(\varphi, \alpha) \models \mathcal{T}$, c'est-à-dire $[\varphi\Diamond_{\mathcal{T}}\alpha] = \{\mathcal{G}_0\}$. Si $(\varphi, \alpha \wedge \beta) \not\models \mathcal{T}$, alors $[\varphi\Diamond_{\mathcal{T}}(\alpha \wedge \beta)] = \varnothing$ (et donc **E9** est satisfait), sinon $[\varphi\Diamond_{\mathcal{T}}(\alpha \wedge \beta)] = \{\mathcal{G}_0\}$ (et donc **E9** est également satisfait).

Indépendance de E8 : soit $\diamondsuit_{\mathcal{T}}$ tel que $\forall \varphi, \alpha \in \mathscr{L}$, si $|[\varphi]| = 1$, alors $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \{\mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}\}$, sinon si $|[\varphi]| \neq 1$ et $(\varphi, \alpha) \in \mathcal{T}$, alors $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \{\mathcal{G}_0\}$, sinon $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$.

- **E3** est satisfait.
- **U4** est satisfait.
- **E5** est satisfait : $[(\varphi \diamondsuit_{\mathcal{T}} \alpha) \wedge \beta] = \{\mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et $\mathcal{G}_1 \in [\beta]\}$ $= [\varphi \diamondsuit_{\mathcal{T}} (\alpha \wedge \beta)]$. D'où le résultat.
- **E8** n'est pas satisfait : si $\exists \mathcal{G}_1$ et $\mathcal{G}_2$ tels que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et $(\mathcal{G}, \mathcal{G}_2) \in \mathcal{T}$, alors $[(f(\mathcal{G}_1) \vee f(\mathcal{G}_2)) \diamondsuit_{\mathcal{T}} \alpha] = \{\mathcal{G}_0\}$ et $[f(\mathcal{G}_1) \diamondsuit_{\mathcal{T}} \alpha] = \{\mathcal{G}' \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}') \in \mathcal{T}\}$ et $[f(\mathcal{G}_2) \diamondsuit_{\mathcal{T}} \alpha] = \{\mathcal{G}'' \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}'') \in \mathcal{T}\}$. On a $[f(\mathcal{G}_1) \diamondsuit_{\mathcal{T}} \alpha] \cup [f(\mathcal{G}_2) \diamondsuit_{\mathcal{T}} \alpha] \neq \{\mathcal{G}_0\}$. Ainsi **E8** n'est pas satisfait.
- **U9** est satisfait : cela peut être prouvé par la même méthode que celle utilisée pour **E5**.

Indépendance de U9 : soit $\diamondsuit_{\mathcal{T}}$ tel que $\forall \varphi, \alpha \in \mathscr{L}$, $\exists \mathcal{G}_0 \in \mathbf{G_U}$, si $(\varphi, \alpha) \models \mathcal{T}$, alors (si $[\alpha] \neq \mathbf{G_U}$, alors $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = [\alpha]$ sinon $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \{\mathcal{G}_0\}$) sinon $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$.

- **E3** est satisfait.
- **U4** est satisfait.
- **E5** est satisfait : si $\mathcal{G}_1 \in [(f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha) \wedge \beta]$, alors si $[\alpha] \neq \mathbf{G_U}$, alors $\mathcal{G}_1 \in ([\alpha] \cap [\beta])$. Ainsi, $[\alpha \wedge \beta] \neq \varnothing$ et $[\alpha \wedge \beta] \neq \mathbf{G_U}$; cela signifie que $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (\alpha \wedge \beta)] = [\alpha \wedge \beta]$, donc **E5** est satisfait. Or, si $[\alpha] = \mathbf{G_U}$, alors $[(f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha) \wedge \beta] = \{\mathcal{G}_0\} \cap [\beta]$; cela signifie que $\mathcal{G}_1 = \mathcal{G}_0$ et $\mathcal{G}_0 \in [\beta]$, donc $\mathcal{G}_0 \in [\alpha \wedge \beta]$. Donc, soit $[\alpha \wedge \beta] = \mathbf{G_U}$ et $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (\alpha \wedge \beta)] = \{\mathcal{G}_0\}$, soit $\alpha \wedge \beta \neq \mathbf{G_U}$ et $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (\alpha \wedge \beta)] = [\alpha \wedge \beta]$. Dans les deux cas, **E5** est satisfait.
- **E8** est satisfait : la définition de $\diamondsuit_{\mathcal{T}}$ ne dépend pas du premier paramètre (excepté pour la détection d'échec) ; la preuve est similaire à celle concernant l'indépendance de **E5**.
- **U9** n'est pas satisfait : par exemple, considérons $\mathcal{G}_1 \in \mathbf{G_U}$ tel que $\mathcal{G}_1 \neq \mathcal{G}_0$ et $(\varphi, f(\mathcal{G}_1)) \models \mathcal{T}$, nous avons $[(\varphi \diamondsuit_{\mathcal{T}} \top) \wedge (f(\mathcal{G}_0) \vee f(\mathcal{G}_1))] \neq \varnothing$, alors que $[\varphi \diamondsuit_{\mathcal{T}} (\top \wedge (f(\mathcal{G}_0) \vee f(\mathcal{G}_1)))] = \{\mathcal{G}_0, \mathcal{G}_1\} \not\subseteq [(\varphi \diamondsuit_{\mathcal{T}} \top) \wedge (f(\mathcal{G}_0) \vee f(\mathcal{G}_1))] = \{\mathcal{G}_0\}$.

□

Démonstration de la proposition 7.6 page 147.
La preuve est directe à partir des équations du théorème 7.2 page 147. □

Démonstration de la proposition 7.7 page 147.
Soit $\varphi, \alpha \in \mathscr{L}$ tel que $[\varphi] \neq \varnothing$ et $[\varphi \diamondsuit_{\mathcal{T}} \alpha] \neq \varnothing$ et $\diamondsuit_{\mathcal{T}}$ satisfait **E3**. Cela signifie que $(\varphi, \alpha) \models \mathcal{T}$, c'est-à-dire $\forall \mathcal{G} \in [\varphi]$, $\exists \mathcal{G}_1 \in [\alpha]$, $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$. Or, $\forall \beta \in \mathscr{L}$, $[\alpha] \subseteq [\alpha \vee \beta]$,

donc $\mathcal{G}_1 \in [\alpha \vee \beta]$. Cela signifie que $(\varphi, \alpha \vee \beta) \models \mathcal{T}$. Grâce à **E3**, nous avons $[\varphi \diamondsuit_{\mathcal{T}} (\alpha \vee \beta)] \neq \varnothing$. $\square$

Démonstration de la proposition 7.8 page 148.
Soit $\mathcal{T}$ une relation réflexive sur $\mathbf{G_U} \times \mathbf{G_U}$.

1. Soit $\diamondsuit_{\mathcal{T}} : \mathscr{L} \times \mathscr{L} \to \mathscr{L}$ un opérateur satisfaisant **U2**, **E3**, **U4**, **E5**, **E8**, **U9**. Pour tout $\mathcal{G} \in \mathbf{G_U}$, définissons $\preccurlyeq_{\mathcal{G}}$ tel que $\mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_2$ ssi $\mathcal{G}_1 \in [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (f(\mathcal{G}_1) \vee f(\mathcal{G}_2))]$ ou $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (f(\mathcal{G}_1) \vee f(\mathcal{G}_2))] = \varnothing$. Grâce au théorème 7.2 page 147, $\preccurlyeq_{\mathcal{G}}$ est un préordre complet sur $\mathbf{G_U}$ respectant $\mathcal{T}$. $\forall \mathcal{G}_1 \in \mathbf{G_U}$, calculons $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (f(\mathcal{G}) \vee f(\mathcal{G}_1))]$. Grâce à **U2**, puisque $f(\mathcal{G}) \models f(\mathcal{G}) \vee f(\mathcal{G}_1)$, nous avons $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} (f(\mathcal{G}) \vee f(\mathcal{G}_1))] = [f(\mathcal{G})]$. Donc, $\mathcal{G} \prec_{\mathcal{G}} \mathcal{G}_1$. Cela signifie que l'assignation est fidèle.

2. $\forall \mathcal{G} \in \mathbf{G_U}$, soit $\preccurlyeq_{\mathcal{G}}$ un préordre complet sur $\mathbf{G_U} \times \mathbf{G_U}$ respectant $\mathcal{T}$, et tel que $\forall \mathcal{G}_1$, $\mathcal{G} \prec_{\mathcal{G}} \mathcal{G}_1$. Soit $\diamondsuit_{\mathcal{T}}$ défini par $\forall \mathcal{G} \in \mathbf{G_U}$, $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] = \{\mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et $\forall \mathcal{G}_2 \in [\alpha], \mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_2\}$ et $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$ si $\exists \mathcal{G} \in [\varphi]$ tel que $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$, sinon $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \bigcup_{\mathcal{G} \in [\varphi]} [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha]$. Grâce au théorème 7.2 page 147, $\diamondsuit_{\mathcal{T}}$ satisfait **U1**, **E3**, **U4**, **E5**, **E8** et **U9**. Vérifions si **U2** est satisfait : soit $\varphi, \alpha \in \mathscr{L}$, tels que $\varphi \models \alpha$.
   (a) Si $[\varphi] = \varnothing$, alors par la proposition 7.6 page 147 $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$, et donc **U2** est satisfait.
   (b) Si $[\varphi] \neq \varnothing$, alors soit $\mathcal{G} \in [\varphi]$, $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] = \{\mathcal{G}_1 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_1) \in \mathcal{T}$ et $\forall \mathcal{G}_2 \in [\alpha]$ tel que $(\mathcal{G}, \mathcal{G}_2) \in \mathcal{T}$, $\mathcal{G}_1 \preccurlyeq_{\mathcal{G}} \mathcal{G}_2\}$. Puisque $\varphi \models \alpha$, cela signifie que $\mathcal{G} \models \alpha$. De plus, puisque $\mathcal{T}$ est réflexive, alors $(\mathcal{G}, \mathcal{G}) \in \mathcal{T}$; enfin, grâce au fait que l'assignation respecte $\mathcal{T}$, $\forall \mathcal{G}_1$, $\mathcal{G} \prec_{\mathcal{G}} \mathcal{G}_1$. Donc $\mathcal{G} \in [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha]$ et $\forall \mathcal{G}_1 \neq \mathcal{G}$, $\mathcal{G}_1 \notin [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha]$. Cela signifie que $[f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] = \{\mathcal{G}\} = [f(\mathcal{G})]$. Cela est vrai pour tout $\mathcal{G} \in [\varphi]$, donc $\bigcup_{\mathcal{G} \in [\varphi]} [f(\mathcal{G}) \diamondsuit_{\mathcal{T}} \alpha] = \bigcup_{\mathcal{G} \in [\varphi]} [f(\mathcal{G})] = [\varphi]$, et donc **U2** est satisfait.

$\square$

Démonstration de la proposition 7.9 page 148.
Soit $\mathcal{T} = \mathbf{G_U} \times \mathbf{G_U}$.

1. Soit $\diamondsuit_{\mathcal{T}}$ un opérateur satisfaisant **U2**, **E3**, **U4**, **E5**, **E8**, **U9**. Montrons que ce dernier satisfait **U1**, **U3**, **U5** et **U8**.

   U1 : cela est dû à la proposition 7.4 page 146.

   U3 : si $[\varphi] \neq \varnothing$, alors grâce à **E3**, $[\varphi \diamondsuit_{\mathcal{T}} \alpha] \neq \varnothing$ ssi $(\varphi, \alpha) \in \mathcal{T}$. Ainsi, $\mathcal{T}$ étant égal à $\mathbf{G_U} \times \mathbf{G_U}$, si $[\alpha] \neq \varnothing$, alors $(\varphi, \alpha) \in \mathcal{T}$, donc $[\varphi \diamondsuit_{\mathcal{T}} \alpha] \neq \varnothing$.

   U5 : cela est dû aux postulats **U8** et **E5**.

   U8 : si $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$ ou $[\psi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$, et $[\varphi] \neq \varnothing$, alors, grâce à **E8**, nous avons $[(\varphi \vee \psi) \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$, donc **U8** est satisfait. Sinon, grâce à **E8**, **U8** est satisfait.

2. Soit $\diamondsuit_{\mathcal{T}}$ un opérateur satisfaisant **U1**, **U2**, **U3**, **U4**, **U5**, **U8** et **U9**. Montrons que ce dernier satisfait **E3**, **E5** et **E8**.

   E3 : quand $\mathcal{T} = \mathbf{G_U} \times \mathbf{G_U}$, $(\varphi, \alpha) \in \mathcal{T} \Leftrightarrow [\alpha] \neq \varnothing$, d'où le résultat.

   E5 : c'est un cas particulier de **U5**.

   E8 : il y a quatre cas,

   - Si $[\varphi] \neq \varnothing$ et $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$, alors, grâce à **U3**, cela signifie que $[\alpha] = \varnothing$, donc, grâce à **U1**, $[(\varphi \vee \psi) \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$, et donc **E8** est satisfait.
   - Si $[\varphi] = \varnothing$ et $[\alpha] = \varnothing$, alors, grâce à **U1**, $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$, donc **U8** s'applique, et donc **E8** est satisfait.
   - Si $[\varphi] = \varnothing$ et $[\alpha] \neq \varnothing$, alors, grâce à **U2**, $[\varphi \diamondsuit_{\mathcal{T}} \alpha] = \varnothing$, donc **U8** s'applique, et donc **E8** est satisfait.
   - Si $[\varphi] \neq \varnothing$ et $[\alpha] \neq \varnothing$, alors, grâce à **U3**, $[\varphi \diamondsuit_{\mathcal{T}} \alpha] \neq \varnothing$, de même que pour ψ. Cela signifie que $[\varphi \diamondsuit_{\mathcal{T}} \alpha] \neq \varnothing$ et $[\psi \diamondsuit_{\mathcal{T}} \alpha] \neq \varnothing$; dans ce cas, **U8** s'applique, et donc **E8** est satisfait.

$\square$

Annexe D

Encore quelques exemples

Sommaire

Cette annexe prolonge nos exemples et justifications. Elle est composée d'une liste de buts possibles, accompagnée d'une intuition pour chacun d'entre eux (section D.1), et de deux exemples illustrant l'intérêt de l'argumentation abstraite et du changement dans le cadre de la prise en compte du contexte et de la prise en compte des préférences ou priorités (section D.2 page 225).

D.1 Buts et autres ambitions

Dans cette annexe, nous souhaitons présenter des exemples de buts et les intuitions possibles qui leurs sont associées. Nous débuterons par les buts absolus (ne concernant que le système cible après le changement) et finirons par les buts relatifs (concernant à la fois le système cible avant et après le changement).

Notons que les buts et intuitions présentés ici se basent sur la typologie des propriétés du changement présentée dans le chapitre 4 page 59. Cette typologie sépare les propriétés du changement en trois niveaux : l'ensemble des extensions, les ensembles d'arguments et un argument particulier. Sur les six cas obtenus en séparant chacun des niveaux entre buts absolus et buts relatifs, nous ne traitons ici que quatre d'entre eux.

Avant de se lancer dans la suite de cette annexe, il est important de noter que ces exemples de buts, s'ils en sont souvent inspirés, ne relèvent pas des applications présentées dans le chapitre 3 page 29. Bien que les personnes et situations mises en jeu portent (éventuellement) le même nom, nous supposons qu'elles sont différentes. [1] En effet, les apparentes incohérences avec ces applications sont la conséquence de la variété des buts. Ces derniers trouvent plus aisément une intuition dans un contexte ou dans un autre ; à ce titre, nous changeons de contexte sans crier gare suivant le but que nous présentons.

Nous prions le lecteur de ne pas tenir compte de ces incohérences, et de lire ce qui suit avec un regard neuf.

D.1.1 Buts absolus

Un but absolu, prenant uniquement en compte le système d'argumentation résultat (c'est-à-dire le système obtenu après modification du système cible), peut servir à représenter un désir durant toute la discussion.

D.1.1.1 Niveau de l'ensemble des extensions

$|\mathbf{E}'| = 0$ **ou** $|\mathbf{E}'| = 1,\ \mathcal{E}' = \varnothing$**.**

Jean le sait, il ne pourra présenter suffisamment d'arguments à ses amis pour éviter d'aller au cinéma. Mais il n'a pas dit son dernier mot, oh non ! S'il ne peut les convaincre, alors il les mènera dans une impasse. Il lui suffit de supprimer toutes les options possibles pour les empêcher de prendre une décision et ainsi éviter une soirée aussi intéressante qu'une course d'escargots éclopés. Pour cela, il doit faire en sorte qu'aucune extension n'existe, ou que l'extension soit vide.

$|\mathbf{E}'| = 1,\ \mathcal{E}' \neq \varnothing$**.**

Georges est irrité. Il se rend bien compte que Jean tente de saboter la soirée cinéma qu'ils avaient mis trois mois à organiser. *Un film documentaire sur la culture des fruits rouges en Antarctique, c'est pourtant très intéressant !*, se dit-il, amer. Mais ce qui l'ennuie vraiment, c'est de rester dehors, dans le froid, à couper les cheveux en dix-huit avec ses futurs-anciens amis. Ce qu'il veut, c'est qu'une décision soit prise, que ce pugilat verbal s'arrête et qu'enfin on puisse tenter de sauver cette soirée en forme de naufrage. Pour cela, il veut une seule option, n'importe laquelle ; ainsi, il doit se débrouiller pour qu'il n'existe qu'une, et une seule, extension.

1. L'amateur de science-fiction y verra une analogie avec les univers parallèles.

$|\mathbf{E}'| > 1$.

Jean ne s'attendait pas à ça : Georges a avancé un argument proprement machiavélique. Il lui est maintenant impossible de jouer la carte de l'impasse décisionnelle. Ayant plus d'un tour dans son sac à malice, il sait comment tourner une nouvelle fois la situation à son avantage. S'il ne peut annihiler toute option acceptable, alors il doit mettre ses amis face à un dilemme ou tout autre choix cornélien prolongeant l'inactivité pathologique d'un groupe de jeunes personnes désœuvrées. En somme, qu'ils ne puissent choisir entre deux films. Pour cela, son but est d'augmenter le nombre d'options possibles, et ainsi d'avoir plusieurs extensions.

D.1.1.2 Niveau d'un argument particulier

$|\mathbf{E}'_X| = 0$.

Jean se doit de défendre son client, coûte que coûte. Son adversaire est coriace, mais il doit à tout prix contrer toutes ses offensives. L'argument de la culpabilité de son client doit être rejeté par les jurés. Pour cela, il doit faire en sorte que cet argument n'appartienne à aucune des extensions de leur système d'argumentation.

$|\mathbf{E}'_X| \neq 0,\ |\mathbf{E}'_X| = |\mathbf{E}'|$.

Georges est sûr de lui. Il a bien dirigé son accusation et, par quelques témoins trop heureux de collaborer, mis en difficulté la défense. Une dernière diatribe, une ode aux valeurs familiales, quelques clins d'œil, et tous les jurés seront à ses pieds. Il pourra sans problème faire accepter unanimement la culpabilité de l'accusé. Pour cela, son argument doit appartenir à toutes les extensions du système d'argumentation des jurés.

$|\mathbf{E}'_X| \neq 0,\ |\mathbf{E}'_X| < |\mathbf{E}'|$.

Jamais une affaire n'aura été aussi dure. Jean a été défait de toutes ses tentatives d'argumentaire par un procureur menant aussi bien les mots que les investigations. Mais l'origine sociale de son client, la gravité des accusations, la pression des journalistes, Jean en a bien conscience. Cette affaire est médiatiquement capitale, et le ministère public n'a pas le droit à l'erreur. Il ne doit pas laisser l'ombre d'un doute. *Voilà ma chance !*, se dit Jean. Tout ce qu'il a à faire, c'est inoculer un germe de doute, et l'arbre de la relaxe fleurira à nouveau. Pour cela, il doit trouver un moyen pour que l'argument n'appartienne plus à au moins une des extensions du système d'argumentation des jurés.

D.1.2 Buts relatifs

Un but relatif, prenant en compte à la fois le système d'argumentation cible et le système d'argumentation résultat, il peut servir à représenter un désir localisé à un moment précis de la discussion.

D.1.2.1 Niveau de l'ensemble des extensions

$|\mathbf{E}| > |\mathbf{E}'|$.

Passer en revue tous les choix possibles prendrait trop de temps, et l'urgence d'une décision se fait sentir. Jean n'a pas le choix, il doit recentrer le débat sur quelques options uniquement. Il doit donc réduire le nombre d'extensions.

$|\mathbf{E}| < |\mathbf{E}'|$.

Le débat tourne en rond. Les deux options proposées ont été discutées en long, en large et en travers et il s'avère qu'elles ne sont satisfaisantes ni l'une ni l'autre. Jean doit élargir le débat et proposer d'autres options. Pour cela, le nombre d'extensions doit être accru.

$|\mathbf{E}| = |\mathbf{E}'|$.

Ça y est ! Les deux camps sont proches d'un compromis, ils sont d'accord sur un ensemble d'options. Jean doit encore faire en sorte que ses collègues acceptent une faveur, mais il doit procéder avec prudence et ne pas risquer de relancer un débat sans fin. Pour cela, il doit s'assurer que sa prochaine action ne modifiera pas le nombre d'extensions.

$\mathbf{E} = \mathbf{E}'$.

Un long débat vient de se terminer et le temps est maintenant venu de délibérer, de prendre une décision. Mais de nombreux arguments ont été avancés, et il n'est pas aisé de prendre toutes ces informations en compte. Jean propose donc d'effectuer un pré-traitement et d'alléger cet ensemble d'arguments pour obtenir un nouvel ensemble permettant une délibération plus limpide. Afin de ne pas altérer le résultat du débat, lui et ses amis doivent néanmoins s'assurer que ce nouvel ensemble reste strictement équivalent à l'ancien en termes de conclusions.

D.1.2.2 Niveau d'ensembles d'arguments

$\forall \mathcal{E}_i \in \mathbf{E},\ \exists \mathcal{E}'_j \in \mathbf{E}',\ \mathcal{E}_i \subseteq \mathcal{E}'_j$.

Jean est content de l'évolution de la discussion. Quelle que soit l'option choisie, elle tiendra compte de ce qu'il désire. Un restaurant ? Pas un chinois alors. Un film ? Que de l'intimiste français. Une soirée chez quelqu'un ? Avec grand plaisir, tant que ce n'est pas chez lui. Jean se voit comme quelqu'un de souple, *tant que l'on respecte mes desiderata*, se dit-il, non sans complaisance. En définitive, peu lui importe ce qu'ils feront ce soir, Jean souhaite juste ne rien perdre de ce qui a été accepté jusque-là car cela reflète en tout point ses goûts. Pour cela, tous les arguments acceptés doivent le rester après son intervention.

$\forall \mathcal{E}'_j \in \mathbf{E}',\ \exists \mathcal{E}_i \in \mathbf{E},\ \mathcal{E}'_j \subseteq \mathcal{E}_i$.

Le débat entre Jean et Georges fait rage. Les arguments fusent des deux côtés et la cadence s'intensifie. Jean s'est laissé avoir par cette hausse de rythme et a commis une erreur. Il est conscient que sa dernière intervention a justifié un des arguments de son adversaire. Cette fois-ci, il ne refera pas la même fredaine, quitte à perdre certains de ses propres arguments : son prochain argument contredira ce qu'il a dit avant et mettra des bâtons dans les roues de son adversaire. Pour cela, aucun argument ne doit être nouvellement accepté après son intervention.

$$\bigcup_{1 \leq i \leq |\mathbf{E}|} \mathcal{E}_i \subseteq \bigcup_{1 \leq j \leq |\mathbf{E}'|} \mathcal{E}'_j.$$

Georges a réussi à imposer ses volontés dans chacune des options possibles et il veut que cela reste ainsi. Il ne souhaite pas mettre à mal ses propres arguments avec sa prochaine intervention. Décidant de ne prendre aucun risque, il se résout à faire en sorte qu'aucun argument, y compris ceux de son adversaire, ne puisse être rejeté par son intervention. Ainsi, tout argument accepté crédulement avant son intervention doit le rester après qu'il a parlé.

$$\bigcup_{1 \leq j \leq |\mathbf{E}'|} \mathcal{E}'_j \subseteq \bigcup_{1 \leq i \leq |\mathbf{E}|} \mathcal{E}_i.$$

Jean ne veut pas risquer de justifier un seul des arguments de son adversaire, même partiellement. Il décide qu'avec son intervention, aucun argument ne sera nouvellement accepté de façon crédule, même un des siens. Pour cela, il doit s'assurer que tout argument accepté crédulement l'était déjà avant son intervention.

$$\bigcap_{1 \leq i \leq |\mathbf{E}|} \mathcal{E}_i \subseteq \bigcap_{1 \leq j \leq |\mathbf{E}'|} \mathcal{E}'_j.$$

Les électeurs sont échaudés et Georges n'a plus le loisir d'être vague. Il sait que dorénavant, ce sont des arguments solides qui permettront de gagner des voix. Il décide donc d'assurer les arguments acceptés de façon sceptique, y compris ceux de son adversaire, et fait en sorte de n'en perdre aucun du fait de son intervention.

$$\bigcap_{1 \leq j \leq |\mathbf{E}'|} \mathcal{E}'_j \subseteq \bigcap_{1 \leq i \leq |\mathbf{E}|} \mathcal{E}_i.$$

Jean voit bien le manège de son rival. Pour le contrecarrer, Jean décide de faire l'inverse et de ne justifier aucun argument de façon sceptique, voire de “casser” certains de ces arguments et de les faire accepter crédulement ou rejeter.

D.2 Justifications supplémentaires du changement

Cette annexe décrit, sans rentrer outre mesure dans les détails, deux causes typiques où le changement en argumentation a de l'intérêt. Nous commençons par approfondir l'évolution du contexte (section D.2.1), puis nous nous penchons sur le changement de priorités d'un agent (section D.2.2).

D.2.1 Un exemple de changements induits par l'évolution du contexte

Dans cet exemple, trois personnes (facétieusement dénommées Rosa, Georges et Jean) discutent de la culpabilité d'un médecin concernant la mort d'un de ses patients. Agent à part entière, chacun d'entre eux possède sa propre base de connaissances et son propre système d'argumentation. Nous allons voir que, suivant le contexte et les connaissances propres à chacun, un enthymème, *i.e.* un argument tronqué, peut être perçu et compris de façon tout à fait différente.

Les arguments échangés (x_1, x_2 et x_3) sont les suivants :

- Rosa (x_1) : "*C'est un médecin ; un médecin ne peut être le meurtrier d'un de ses patients. Il n'est donc pas un meurtrier.*"
- Georges (x_2) : "*Il a pourtant injecté une dose mortelle de médicaments à son patient. Il est donc un meurtrier.*"
- Jean (x_3) : "*Il a tout simplement procédé à une euthanasie.*"

Notons en premier lieu que l'argument x_3 est un argument tronqué : la conclusion a été omise à l'oral par Jean. Cet argument est donc laissé à la libre interprétation des deux autres protagonistes ; nous allons en voir les conséquences.

Plaçons-nous du point de vue de Rosa : pour celle-ci, l'euthanasie renvoie à la juridiction luxembourgeoise, où elle est légale. Ainsi, sa base de connaissances stocke la règle *si euthanasie alors non meurtrier*, ce qui lui permet de compléter l'argument x_3 en : "*Il a tout simplement procédé à une euthanasie. Il n'est donc pas un meurtrier.*" L'argument x_3 l'emporte donc, *pour elle*, sur l'argument x_2. La figure D.1 présente le système d'argumentation de Rosa.

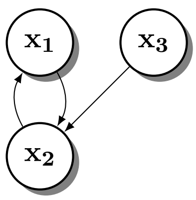


Figure D.1 – *Système d'argumentation de Rosa.*

Plaçons-nous maintenant du point de vue de Georges : pour celui-ci, l'euthanasie renvoie à la juridiction française, où elle est formellement interdite et assimilée à un meurtre. Contrairement à Rosa, Georges possède dans sa base de connaissances la règle *si euthanasie alors meurtrier*, ce qui lui permet de compléter l'argument x_3 en : "*Il a tout simplement procédé à une euthanasie. Il est donc un meurtrier.*"

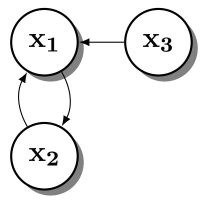


Figure D.2 – *Système d'argumentation de Georges.*

L'argument x_3 l'emporte donc, *pour lui*, sur l'argument x_1. La figure D.2 présente le système d'argumentation de Georges.

Après avoir examiné leur propre système d'argumentation, et sûrs de leur affaire, Rosa et Georges s'exclament en chœur “ *Vois-tu, Jean est d'accord avec moi !*”. Interloqués par un tel synchronisme labial, Rosa et Georges demandent à Jean de préciser de la plus diligente des manières son propos. Ce dernier explique que l'euthanasie est pour lui loin d'être un meurtre ; de la même façon que Rosa, il possède dans sa base de connaissances la règle *si euthanasie alors non meurtrier*.

L'agent Georges, qui s'est manifestement égaré dans son interprétation (il n'avait pas la règle “adéquate” dans sa base de connaissances), se doit de remettre à jour son système d'argumentation en supprimant l'argument x_3 qui n'a plus lieu d'être et rajouter un nouvel argument, que nous nommerons x_3', attaquant l'argument x_2. La figure D.3 présente le nouveau système d'argumentation de Georges. S'il veut convaincre les autres agents, Georges doit alors trouver un nouvel argument.

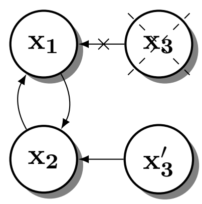


Figure D.3 – *Nouveau système d'argumentation de Georges.*

Ainsi, nous voyons, grâce à cet exemple, que le contexte dans lequel se place un dialogue et les connaissances des agents peuvent *influer* sur la perception d'un argument tronqué et lui octroyer des attaques différentes suivant les circonstances. Par ailleurs, lorsque ces connaissances sont mises en commun, cet argument peut se révéler erroné et doit dans ce cas être supprimé et éventuellement remplacé par un nouvel argument adéquat.

D.2.2 Un exemple de changements induits par l'évolution de priorités

Nous nous penchons, dans cet exemple, sur la création d'arguments en fonction de règles ainsi que sur les modifications que peuvent subir ces arguments selon l'évolution des priorités entre ces règles. Notons que nous nous inspirons librement de l'exemple de Liao *et al.* (2011).

Supposons que nous sommes en présence d'un système à base de règles (voir par exemple le travail de Capobianco *et al.* (2005)) pouvant déterminer quel comportement avoir en présence d'un nouvel email. Ces règles, qui sont de priorités diverses, permettent de créer des arguments qui vont alors interagir pour déterminer l'action à préconiser. Notons que lorsque deux arguments ont des conclusions contradictoires, et qu'ils sont issus de règles de priorités égales, ces derniers s'entre-attaquent ; par contre, si ces deux arguments sont issus de règles qui ont des priorités différentes, ils sont alors de priorité différente et l'argument ayant la plus grande priorité est le seul à pouvoir attaquer.

Les règles sont les suivantes :

- r_1 : Un email provenant du réseau local ne présente pas de danger : $local(email) \to \neg danger(email)$.
- r_2 : Un email contenant un virus présente un danger : $virus(email) \to danger(email)$.
- r_3 : un email ne présentant pas de danger doit être déplacé dans le courrier entrant : $\neg danger(email) \to courrier\_entrant(email)$.
- r_4 : un email présentant un danger doit être déplacé dans la corbeille : $danger(email) \to corbeille(email)$.

Supposons que dans un premier temps, toutes les règles possèdent la même priorité.

Le système reçoit alors un email, nommé m_1, possédant les caractéristiques suivantes :

- m_1 provient du réseau local.
- m_1 est suspecté de contenir un virus.

En se basant sur les règles précédentes, il est possible de construire quatre arguments (tronqués) [2] :

- $x_1 = (\{local(m_1), r_1\}, \neg danger(m_1))$
- $x_2 = (\{virus(m_1), r_2\}, danger(m_1))$
- $x_3 = (\{local(m_1), r_1, r_3\}, courrier\_entrant(m_1))$
- $x_4 = (\{virus(m_1), r_2, r_4\}, corbeille(m_1))$

La figure D.4 page ci-contre présente le système d'argumentation alors obtenu.

Dans une telle situation, il paraît difficile de prendre une décision puisqu'il existe deux solutions possibles (mettre le courrier dans la corbeille ou dans le courrier entrant) et qu'aucune ne ressort de l'argumentaire fourni [3].

2. Notons qu'on ne construit pas tous les arguments possibles et qu'on ne précise pas la façon dont ces arguments, ainsi que leurs attaques, sont construits.

3. Nous avons vu dans le chapitre 2 page 11 que cela se traduit, par exemple avec la sémantique préférée, par deux extensions $\mathcal{E}_1 = \{x_1, x_3\}$ et $\mathcal{E}_2 = \{x_2, x_4\}$.

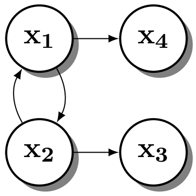


Figure D.4 – *Système d'argumentation lorsque les règles sont de même priorité.*

Supposons maintenant que la prudence est privilégiée. La règle r_2 devient ainsi plus prioritaire que les autres et en conséquence, l'argument x_2 est préféré à n'importe quel autre. De ce fait, l'attaque de l'argument x_1 vers x_2 n'est plus valable. La figure D.5 présente le système d'argumentation après avoir modifié les priorités.

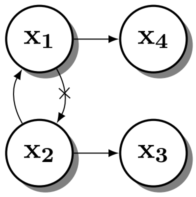


Figure D.5 – *Système d'argumentation lorsque la règle r_2 est plus prioritaire que les autres.*

Il semble à présent possible de prendre la décision de mettre m_1 dans la corbeille.[4]

4. Ce qui se traduira, en utilisant ici encore la sémantique préférée, par l'existence d'une seule extension, à savoir $\mathcal{E} = \{x_2, x_4\}$.

Index

Study of change in abstract argumentation
From theory to practice

Abstract

Argumentation, in the field of artificial intelligence, is a formalism allowing to reason with incomplete and/or contradictory information as well as to model an exchange of arguments between several agents. An argumentation system usually consists of a set of arguments interacting with each other, and from which it is possible to extract one or several consistent points of view.

In this thesis, we are mainly concerned with the *abstract argumentation* in which arguments are handled as abstract entities whose meaning is unknown and in which the interactions represent conflicts. This allows us to focus on the particular point of the dynamics in abstract argumentation systems, that is to say the *changes* that could impact these systems, particularly in the context of a dialogue.

We start with justifying the interest of such a formal framework, then we study the *how* and the *why* of change in abstract argumentation. The *how* is tackled by establishing a list of changes that an argumentation system can undergo and by studying the conditions under which they may occur. The *why* is addressed by introducing the notion of goal motivating a change and by choosing the best change to make in order to satisfy a goal, taking into account constraints on the agent to convince. Finally, we make our study concrete by proposing a tool that implements the concepts introduced and we study its performance.

Keywords: Abstract argumentation, dynamics of an argumentation system, change.

Étude du changement en argumentation
De la théorie à la pratique

Soutenue le 06/12/2013 par

Pierre Bisquert

Sous la direction de

Marie-Christine Lagasquie-Schiex
Professeur d'Université
Université Paul Sabatier

Florence Dupin de Saint-Cyr
Maître de Conférences
Université Paul Sabatier

À l'Institut de Recherche en Informatique de Toulouse

En vue de l'obtention du

Doctorat de l'Université de Toulouse

En Informatique

Résumé

L'argumentation, au sens de l'intelligence artificielle, est un formalisme permettant de raisonner à partir d'informations incomplètes et/ou contradictoires ainsi que de modéliser un échange d'arguments entre plusieurs agents. Un système d'argumentation consiste généralement en un ensemble d'arguments interagissant les uns avec les autres, et duquel il est possible d'extraire un ou plusieurs points de vue cohérents.

Dans cette thèse, nous nous plaçons dans le cadre de l'*argumentation abstraite* dans lequel les arguments sont manipulés en tant qu'entités abstraites dont le sens nous est inconnu et dans lequel les interactions représentent des conflits. Ceci nous permet de nous concentrer sur le point particulier de la dynamique dans les systèmes d'argumentation abstraits, c'est-à-dire les *changements* pouvant impacter ces systèmes, notamment dans le cadre d'un dialogue.

Nous commençons par justifier l'intérêt d'un tel cadre formel puis nous nous intéressons au *comment* et au *pourquoi* du changement en argumentation abstraite. Le *comment* est approché en établissant une liste des modifications que peut subir un système d'argumentation et en étudiant sous quelles conditions elles peuvent survenir. Le *pourquoi* est abordé par l'introduction de la notion de but motivant un changement et le choix du meilleur changement à faire pour satisfaire un but en prenant en considération des contraintes portant sur l'agent à convaincre. Enfin, nous concrétisons notre étude en proposant un outil logiciel implémentant les notions introduites et nous étudions ses performances.

Mots clefs : Argumentation abstraite, dynamique d'un système d'argumentation, changement.

Institut de Recherche en Informatique de Toulouse – UMR 5505

Université Paul Sabatier, 118 Route de Narbonne, 31062 Toulouse