



HAL
open science

Changement optimal en argumentation Un cadre modélisant l'impact d'un système sur un autre système

Pierre Bisquert, Claudette Cayrol, Florence Dupin de Saint-Cyr, Marie-Christine
Lagasquie-Schiex

► **To cite this version:**

Pierre Bisquert, Claudette Cayrol, Florence Dupin de Saint-Cyr, Marie-Christine Lagasquie-Schiex. Changement optimal en argumentation Un cadre modélisant l'impact d'un système sur un autre système. [Rapport de recherche] IRIT-2013-23, IRIT - Institut de recherche en informatique de Toulouse. 2013. <hal-02884057>

HAL Id: hal-02884057

<https://hal.science/hal-02884057v1>

Submitted on 29 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

**Changement optimal en argumentation
Un cadre modélisant l'impact d'un
système sur un autre système**

Pierre Bisquert Claudette Cayrol
Florence Dupin de Saint-Cyr
Marie-Christine Lagasque-Schiex

IRIT, Université Paul Sabatier,
118 route de Narbonne, 31062 Toulouse, France
{bisquert, ccayrol, dupin, lagasq}@irit.fr

Rapport IRIT
RR- -2013-23- -FR

Avril 2013

Résumé

Ce document définit un cadre théorique pour étudier le changement en argumentation. Ce cadre permet de prendre en compte le raisonnement d'un agent qui désire modifier un système d'argumentation cible afin d'atteindre certains buts. Les modifications sont des ajouts/retraits d'arguments ou d'attaques. L'agent est contraint par ses propres connaissances représentées par un deuxième système d'argumentation. Nous proposons une intuition de ce travail à travers un exemple prenant place dans une audience au tribunal puis nous présentons un protocole permettant de gérer un dialogue de persuasion d'un auditoire. Enfin nous étudions le choix d'arguments en traitant la notion du changement optimal, c'est-à-dire la comparaison de plusieurs changements pour choisir celui qui optimise un, ou plusieurs, critères particuliers. Pour ce faire, nous présentons différents critères et les relations de préférence sur les changements qui en découlent.

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

Table des matières

1	Introduction	1
2	Rappel et bases de travail	2
2.1	Système d'argumentation	2
2.2	Opérations de changement	3
2.3	Propriétés du changement	4
3	Intuition du travail	5
3.1	Protagonistes	6
3.2	Interprétation des notions d'argumentation abstraite	7
3.2.1	Interprétation des sémantiques d'acceptabilité	7
3.2.2	Interprétation du statut d'un argument	9
3.3	Protocole du jeu de l'audience	10
4	Nouvelles définitions	14
4.1	Buts	14
4.2	Univers	15
4.3	Opérations de changement	21
5	Choix des opérations de changement	26
5.1	Déroulement	27
5.2	Changement optimal	28
5.2.1	Coût de programmes	30
5.2.2	Distance structurelle	31
5.2.3	Distance sémantique	34
5.2.4	Comparaison de programmes	35
5.2.5	Combinaison de critères	37
6	Conclusion	39
A	Exemples de buts et autres ambitions	41
A.1	Buts absolus	41
A.1.1	Niveau de l'ensemble des extensions	41
A.1.2	Niveau d'un argument particulier	42
A.2	Buts relatifs	42
A.2.1	Niveau de l'ensemble des extensions	43
A.2.2	Niveau d'ensembles d'arguments	43

Table des figures

1	Illustration présentant un échange typique entre le procureur et l'avocat.	13
2	Illustration de la notion d'abstraction de l'univers de référence.	17
3	Illustration de la notion de système d'argumentation.	20
4	Illustration des contraintes sur les opérations.	21
5	Illustration des différentes (et arbitraires) possibilités de changement du système d'argumentation des jurés (\mathbf{U}_J).	30

Liste des tableaux

1	Arguments mis en jeu lors de l'audience au tribunal.	16
2	Liste des divers programmes que le procureur peut exécuter. .	29

1 Introduction

Un système d’argumentation abstrait Dung (1995) se compose d’un ensemble d’éléments nommés arguments et d’une relation binaire, nommée relation d’attaque, qui représente des conflits entre arguments. Une sémantique pour une telle structure permet de déterminer des ensembles d’arguments conjointement acceptables. Depuis quelques années, les chercheurs en argumentation ont commencé à s’intéresser à la dynamique de ces systèmes (cf. Boella et al. (2009b,a); Cayrol et al. (2010); Baumann and Brewka (2010); Moguillansky et al. (2010); Liao et al. (2011)).

S’inscrivant dans le domaine de la dynamique en argumentation, le cadre que nous allons décrire dans ce document est proche de la planification 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 public, les arguments échangés publiquement ainsi que leurs interactions peuvent constituer le système cible. Les moyens d’actions de l’agent sont des ajouts ou des retraits d’arguments ou d’attaques sur le système cible. Dans un débat public, l’ajout d’un argument consiste simplement à l’énoncer. Le retrait d’un argument paraît moins naturel mais il peut se produire (Bisquert et al. (2012a)) lors d’une objection ou d’une disqualification totale d’un énoncé que l’on ne reconnaît pas comme argument. L’ajout et le retrait d’attaque peuvent respectivement venir du constat que deux arguments déjà énoncés et indépendants sont en fait en conflit, et du constat qu’il n’y a finalement pas d’opposition entre deux arguments considérés contradictoires.

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, (puisque’il ne peut pas, par exemple, ajouter des arguments ou des attaques qu’il ne connaît pas) et cela 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 travail, 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 deux changements.

Dans la section 2, nous rappellerons quelques notions sur lesquelles repose notre travail. Nous proposerons une intuition de notre démarche et de quelques-unes des notions classiques de l’argumentation grâce à un exemple concret dans la section 3. Nous présenterons alors notre cadre théorique dans la section 4. Enfin, nous aborderons le problème de la comparaison et du choix d’un changement dans la section 5.

2 Rappel et bases de travail

Avant d’aller plus loin dans le sujet qui nous intéresse, il convient de rappeler les bases sur lesquelles il repose.

2.1 Système d’argumentation

Le travail présenté dans cet article se place dans le cadre formel proposé par Dung (1995). Celui-ci définit ce qu’est un système d’argumentation, notion que nous reprendrons et modifierons plus loin pour nous permettre de prendre en compte l’“ignorance” d’un agent.

Définition 1 (Système d’argumentation (Dung (1995))). *Un système d’argumentation est une paire $\langle \mathbf{A}, \mathbf{R} \rangle$, où \mathbf{A} est un ensemble non vide fini d’arguments et \mathbf{R} est une relation binaire sur \mathbf{A} , appelée relation d’attaque. Soit $A, B \in \mathbf{A}$, ARB signifie que A attaque B . $\langle \mathbf{A}, \mathbf{R} \rangle$ est représenté par un graphe d’argumentation \mathcal{G} dont les sommets et les arcs correspondent respectivement aux arguments et aux attaques¹.*

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 (Attaque à partir et vers un ensemble). *Soit $A \in \mathbf{A}$ et $\mathcal{S} \subseteq \mathbf{A}$,*

- \mathcal{S} attaque A si et seulement si $\exists X \in \mathcal{S}$ tel que XRA .
- A attaque \mathcal{S} si et seulement si $\exists X \in \mathcal{S}$ tel que ARX .

Les ensembles acceptables d’arguments (“extensions”) sont déterminés à l’aide de sémantiques reposant essentiellement sur les notions suivantes :

Définition 3 (Sans conflit, défense et admissibilité). *Soit $A \in \mathbf{A}$ et $\mathcal{S} \subseteq \mathbf{A}$*

1. Dans ce document, nous utiliserons librement $\langle \mathbf{A}, \mathbf{R} \rangle$ 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 \mathbf{A} et \mathcal{G} .

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

L'ensemble des extensions de $\langle \mathbf{A}, \mathbf{R} \rangle$ est noté \mathbf{E} (avec $\mathcal{E}_1, \dots, \mathcal{E}_n$ dénotant les extensions). Ainsi, en ne nous intéressant qu'aux sémantiques les plus utilisées proposées par Dung (1995), nous avons :

Définition 4 (Sémantiques d'acceptabilité). Soit $\mathcal{E} \subseteq \mathbf{A}$.

- \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} .

Par ailleurs, le statut d'un argument est fonction de sa présence dans les extensions de la sémantique choisie. Par exemple, un argument est "accepté sceptiquement" (resp. "crédulement") s'il apparaît dans toutes les extensions (resp. au moins une extension) et "rejeté" s'il n'appartient à aucune d'entre elles.

2.2 Opérations de changement

Notre travail nécessite également les notions introduites dans Cayrol et al. (2010). 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 pour tenir compte de la notion modifiée de système d'argumentation.

Définition 5 (Opérations élémentaires de changement (Cayrol et al. (2010))). Soit $\langle \mathbf{A}, \mathbf{R} \rangle$ un système d'argumentation, Z un argument, $i_0 =$

(X, Y) avec $X \in \mathbf{A}$ et $Y \in \mathbf{A}$ une interaction entre deux arguments de \mathbf{A} et \mathcal{I}_z un ensemble d'interactions concernant Z ².

- Ajouter une interaction $i_0 \notin \mathbf{R}$ est une opération de changement, notée \oplus_i , fournissant un nouveau système d'argumentation tel que :

$$\langle \mathbf{A}, \mathbf{R} \rangle \oplus_i i_0 = \langle \mathbf{A}, \mathbf{R} \cup \{i_0\} \rangle$$

- Supprimer une interaction $i_0 \in \mathbf{R}$ est une opération de changement, notée \ominus_i , fournissant un nouveau système d'argumentation tel que :

$$\langle \mathbf{A}, \mathbf{R} \rangle \ominus_i i_0 = \langle \mathbf{A}, \mathbf{R} \setminus \{i_0\} \rangle$$

- Ajouter un argument $Z \notin \mathbf{A}$ et $\mathcal{I}_z \not\subseteq \mathbf{R}$ est une opération de changement, notée \oplus_i^a , fournissant un nouveau système d'argumentation tel que :

$$\langle \mathbf{A}, \mathbf{R} \rangle \oplus_i^a (Z, \mathcal{I}_z) = \langle \mathbf{A} \cup \{Z\}, \mathbf{R} \cup \mathcal{I}_z \rangle$$

- Supprimer un argument $Z \in \mathbf{A}$ et $\mathcal{I}_z \subseteq \mathbf{R}$ est une opération de changement, notée \ominus_i^a , fournissant un nouveau système d'argumentation tel que³ :

$$\langle \mathbf{A}, \mathbf{R} \rangle \ominus_i^a (Z, \mathcal{I}_z) = \langle \mathbf{A} \setminus \{Z\}, \mathbf{R} \setminus \mathcal{I}_z \rangle$$

Une opération élémentaire de changement au sens de Cayrol et al. (2010) 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 \ominus_i et \oplus_i) ou un couple (argument, ensemble d'interactions concernant l'argument) (pour \ominus_i^a et \oplus_i^a). L'ensemble des extensions de $\langle \mathbf{A}', \mathbf{R}' \rangle$ y est noté \mathbf{E}' (avec $\mathcal{E}'_1, \dots, \mathcal{E}'_n$ dénotant les extensions).

Notons que Cayrol et al. (2010) conçoit une opération comme faite à partir d'un système d'argumentation cible et "créant" un autre système d'argumentation résultant. Cette définition d'opération sera reprise plus tard dans le but de la restreindre.

2.3 Propriétés du changement

Enfin, notre dernière source d'inspiration se révèle être Bisquert et al. (2012c) pour sa notion de propriété du changement. Définie indépendamment de la notion d'opération de changement en elle-même, une propriété

2. Notons que nous supposons que Z ne s'attaque pas lui-même, et que $\forall (X, Y) \in \mathcal{I}_z$, on a soit $(X = Z \text{ et } Y \neq Z, Y \in \mathbf{A})$ ou $(Y = Z \text{ et } X \neq Z, X \in \mathbf{A})$.

3. Notons que dans le cas de la suppression, \mathcal{I}_z est l'ensemble de *toutes* les interactions concernant Z dans $\langle \mathbf{A}, \mathbf{R} \rangle$.

du changement est une modification structurelle d'un système d'argumentation causée par une de ces opérations. Bisquert et al. (2012c) en distingue plusieurs niveaux suivant qu'elles concernent l'ensemble des extensions, des ensembles d'arguments ou le statut d'un argument particulier. Pour chacun de ces niveaux, plusieurs cas et sous-cas sont définis ; à titre d'exemple, un changement satisfait la propriété *extensive* si et seulement si $|\mathbf{E}| < |\mathbf{E}'|$, c'est-à-dire si le nombre d'extensions du système augmente strictement.

Nous nous abstenons de rappeler ici toutes ces propriétés. Néanmoins, ces dernières, et en particulier leurs caractérisations, guideront notre choix lorsque viendra le moment de déterminer un changement optimal ; elles seront brièvement rappelées à ce moment-là.

3 Intuition du travail

Du sujet qu'est le changement en argumentation découlent de nombreuses interrogations. “*Comment le représenter ? Quelles en sont les conséquences ?*” ne sont que les premières questions d'une longue série. D'autres points, moins classiques, ont émergé plus récemment et se concentrent sur l'usage de tels changements : “*à 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 minimal** et **la prise en compte de l'audience**, qui nous ont amenés à proposer la formalisation décrite dans ce document. Mais avant de rentrer dans le vif du sujet, c'est-à-dire la partie formelle, attardons-nous quelques minutes sur l'idée, l'intuition, de ce que nous tenterons de construire au fil de ce document.

Pour ce faire, nous présenterons ces idées à travers l'exemple décrit dans Bisquert et al. (2012b) et Bisquert et al. (2012a), exemple simple et se prêtant bien à la notion d'argumentation : une audience de tribunal. Bien entendu, si cet exemple permet d'illustrer aisément les notions mises en jeu, il ne résume pas à lui seul le cadre applicatif de notre travail ; nous proposerons, en fin de ce document, d'autres exemples et contextes où il pourrait se révéler intéressant. Cet exemple sera présenté en trois parties, la première se concentrant sur les protagonistes, la seconde présentant les buts pour chacun des participants et, enfin, la troisième illustrant le déroulement de l'audience.

3.1 Protagonistes

Ce jeu s’inspire du fonctionnement d’une audience d’un tribunal mettant en scène quatre entités aux rôles bien distincts qui interagissent pour déterminer si un argument A_0 , dit l’argument-sujet, est acceptable. Puisque nous nous plaçons dans un contexte respectant la présomption d’innocence, l’argument A_0 statue que l’inculpé n’est pas coupable du crime dont il est accusé.

- Le **procureur** veut faire rejeter l’argument A_0 , argument représentant le sujet de l’audience et l’innocence de l’accusé. Il a sa disposition un ensemble d’arguments, dont il ne partage pas nécessairement tous les éléments avec son adversaire.
- L’**avocat de la défense**, disposant également d’un ensemble d’arguments, a un comportement similaire au procureur, à la différence qu’il tente de faire accepter l’argument A_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 supprimer l’argument correspondant), ou la rejeter.
- Les **jurés**⁴ 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 A_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 A_0 est acceptable ou non.

Dans cet exemple, les deux orateurs ne sont pas intéressés par le fait de se convaincre mutuellement. Par contre, ils souhaitent vivement convaincre ceux qui vont prendre la décision finale, c’est-à-dire les jurés. Les deux protagonistes vont donc jouer leurs coups sur un système d’argumentation 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. Si un des deux protagonistes n’a pas connaissance de l’argument que son adversaire énonce, il va l’ajouter dans son propre système.

4. Bien qu’au pluriel, les jurés sont une seule et même entité décisionnaire.

3.2 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 A_0 soit rejeté lors de la délibération ; l’avocat, souhaitant l’inverse, doit faire en sorte que l’argument A_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 A_0 (*i.e.* son appartenance à une ou plusieurs extensions).

3.2.1 Interprétation des sémantiques d’acceptabilité

Les sémantiques d’acceptabilité permettent de déterminer les ensembles d’arguments acceptables. 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*⁵, seyant bien à la question de la culpabilité d’une personne, chose particulièrement délicate. Bien qu’il n’ait pas son mot à dire, 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

5. La charge de la preuve est l’obligation qui incombe à une partie à un procès de prouver un fait ou un droit. Elle est supportée par l’une ou l’autre des parties, selon que le procès est régi par le système accusatoire ou le système inquisitoire. (Citation Jurispedia.org)

laisse aucun doute aux jurés. À l'inverse, 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 co-exister 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 au 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 au 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 Il a été prouvé dans Dung (1995) que tout ensemble d'arguments stable est un *ensemble préféré* (l'inverse n'étant pas vrai, bien que cela soit souvent le cas). On peut donc 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 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 semi-stable, lequel attaque le plus possible d’arguments. 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.

3.2.2 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 Contrairement au cas précédent, un argument dit accepté crédulement est un argument appartenant à au moins une extensions 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 de manière sceptique, c'est-à-dire sans aucun doute possible, qu'il est coupable ; il suffira donc que son innocence soit acceptée de manière crédule, c'est-à-dire en ayant des doutes sur sa culpabilité. Ceci peut être résumé dans notre exemple par le fait que l'avocat, qui est intéressé par une acceptation sceptique de A_0 peut, s'il n'y arrive pas, se rabattre sur une acceptation crédule de A_0 .

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 A_0 représentant l'innocence de l'accusé.

3.3 Protocole du jeu de l'audience

Cette partie sera pour nous l'occasion de présenter le déroulement d'un échange d'arguments dans notre exemple de l'audience. Nous y présentons tout d'abord un protocole gérant l'audience au tribunal de manière générale, comprenant les divers échanges d'arguments, les effets qui peuvent en découler, etc. Nous présentons ensuite un protocole gérant l'acte à proprement parler d'un agent où nous décrivons comment un agent, en fonction du système des jurés, choisit une action.

Ainsi, examinons tout d'abord le protocole général de l'audience, qui voit le procureur et l'avocat s'affronter pour déterminer l'innocence ou la culpabilité de l'accusé.

1. Tout d'abord, l'argument A_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'ajout d'un ou plusieurs arguments et/ou attaques directement dans le système d'argumentation des jurés.
 - Les *actions suppressives*, c'est-à-dire la suppression d'un ou plusieurs arguments et/ou attaques du système d'argumentation des jurés, (par exemple objection, retrait, etc.). Notons le comportement particulier de l'objection : elle 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 supprimés 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, le jeu s'arrête et les jurés délibèrent grâce à leur système d'argumentation.

La figure 1 présente un court exemple illustrant ce protocole.

Ainsi, il existe deux mécanismes de changement ayant des contraintes différentes :

- le premier étant fait par un agent sur un système cible autre que le sien,
- le second étant fait par un agent sur son propre système.

Ce dernier peut être étudié en utilisant directement les travaux de Cayrol et al. (2010) et ne sera par conséquent pas traité ici.

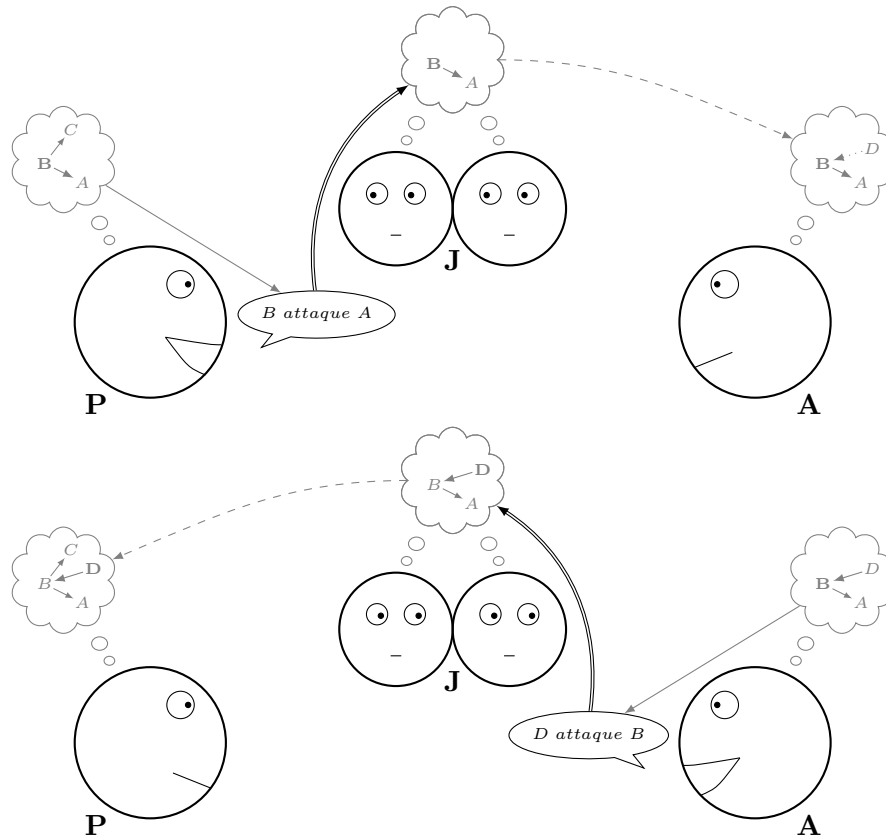


Figure 1 – Illustration présentant un échange typique entre le procureur (**P**) et l’avocat (**A**) sous les yeux des jurés (**J**). On y voit tout d’abord la prise de parole du procureur, énonçant l’argument *B* et l’attaque de *B* vers *A*, ce qui modifie le système d’argumentation des jurés ; celui de l’avocat est ensuite mis à jour avec cette nouvelle information, ce qui lui permet de calculer l’attaque de *D* vers *B*. Cette nouvelle attaque est bien utile à l’avocat, qui s’empresse de l’énoncer (ainsi que *D*), modifiant à son tour le système des jurés ; le système du procureur est finalement mis à jour pour y inclure ces nouvelles informations.

4 Nouvelles définitions

Cette section nous permettra d’introduire formellement certaines notions utiles à notre travail, notions que nous tenterons d’exemplifier au fur et à mesure. À ce titre, nous reprendrons et modifierons des notions que nous avons eu l’occasion de présenter dans la section 2.

4.1 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 Bisquert et al. (2012c), 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 (\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 .

Nous considérons deux types de buts, à savoir “absolus” et “relatifs”. Par ailleurs, nous divisons chacun de ces types en trois niveaux suivant les différents niveaux de la typologie, à savoir :

- l’ensemble des extensions,

- les ensembles d'arguments et
- un argument en particulier.

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 $|E'| = 1$, $\mathcal{E}' \neq \emptyset$. Notons que, puisque le système cible n'est pas pris en compte, un agent peut avoir un ou plusieurs buts étant déjà satisfaits; 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 et son système résultat. 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 qu'une partie de la condition soit satisfaite par le système d'argumentation cible.

Le lecteur pourra se référer à l'annexe A pour une liste plus complète de buts, ainsi qu'une intuition pour chacun d'entre eux.

4.2 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 dans lesquels il peut piocher pour, par exemple, dialoguer. 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 infini d'arguments manipulables dans un domaine (ou contexte) donné.⁶ Notons que nous ne précisons pas, à dessein,

6. Il peut donc exister plusieurs univers.

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 qui ont été extraits de ce texte. Nous pouvons enfin, tel l'exemple qui suit, considérer que l'univers de référence est fourni explicitement.

Exemple 1. *Lors d'une audience au tribunal concernant le prévenu M. X, de multiples arguments peuvent être mis en jeu pour déterminer la culpabilité de ce dernier. La table 1 présente les divers arguments qui peuvent être avancés durant cette audience, c'est-à-dire l'univers de référence dans ce contexte particulier.*

	Argument
A_0	<i>M. X n'est pas coupable d'homicide volontaire avec préméditation sur la personne de M^{me} X, sa femme.</i>
A_1	<i>M. X est coupable d'homicide volontaire avec préméditation sur la personne de M^{me} X, sa femme.</i>
A_2	<i>L'accusé a un alibi, sa secrétaire ayant juré sur l'honneur qu'elle l'avait vu à l'heure de crime.</i>
A_3	<i>La personnalité effacée et influençable de la secrétaire ne peut que mettre en doute la véracité de ses propos.</i>
A_4	<i>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.</i>
A_5	<i>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.</i>
A_6	<i>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.</i>
A_7	<i>L'accusé est un homme connu pour être vénal et son "amour" pour une femme très riche ne pourrait être qu'appât du gain.</i>

Table 1 – *Arguments mis en jeu lors de l'audience au tribunal.*

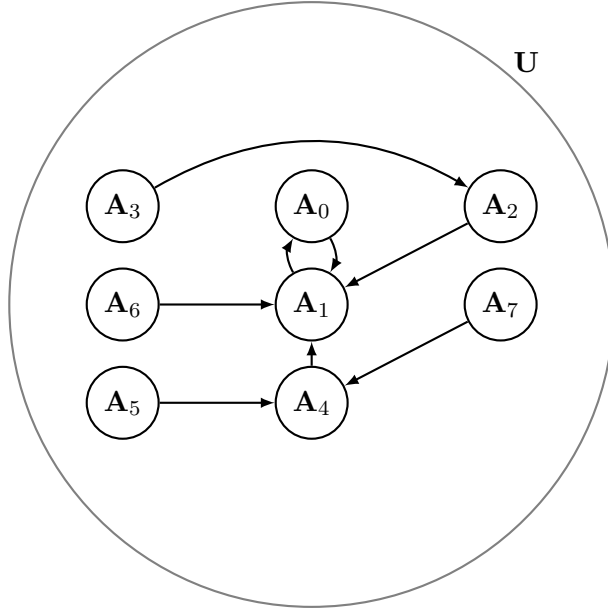


Figure 2 – Illustration de la notion d’abstraction de l’univers de référence.

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 A_i) et la structure qui en découle.

Définition 6 (Abstraction de l’univers de référence). Une abstraction d’univers de référence \mathbf{U} est une paire $\langle \mathbf{A}_{\mathbf{U}}, \mathbf{R}_{\mathbf{U}} \rangle$, où $\mathbf{A}_{\mathbf{U}}$ est l’ensemble non vide de tous les arguments de cet univers et $\mathbf{R}_{\mathbf{U}} \subseteq \mathbf{A}_{\mathbf{U}} \times \mathbf{A}_{\mathbf{U}}$ est une relation binaire sur $\mathbf{A}_{\mathbf{U}}$.⁷

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

Exemple 1 (cont.). En abstrayant l’univers de référence représenté par la table 1, nous obtenons le graphe illustré par la figure 2.

De la notion d’abstraction de l’univers de référence découle directement une nouvelle définition du système d’argumentation. Celle-ci se distingue de

7. Dans la suite de ce document, \mathbf{U} réfèrera autant à un univers de référence qu’à son abstraction.

la définition de Dung (1995) par le fait que le système d’argumentation est relatif à un univers de référence.

Définition 7 (Système d’argumentation). *Étant donné un univers de référence \mathbf{U} , un système d’argumentation issu de \mathbf{U} est une paire $\langle \mathbf{A}, \mathbf{R} \rangle$, où $\mathbf{A} \subseteq \mathbf{A}_{\mathbf{U}}$ est un ensemble non vide fini d’arguments et $\mathbf{R} \subseteq \mathbf{R}_{\mathbf{U}} \cap \mathbf{A} \times \mathbf{A}$ est une relation binaire sur \mathbf{A} . $\langle \mathbf{A}, \mathbf{R} \rangle$ est représenté par un graphe d’argumentation \mathcal{G} dont les sommets et les arcs correspondent respectivement à \mathbf{A} et \mathbf{R} . L’ensemble des graphes représentant un système d’argumentation issu de \mathbf{U} est noté $\mathbf{G}_{\mathbf{U}}$.⁸*

Dans la suite de ce travail, un système d’argumentation sera le plus souvent associé à un agent. Ainsi, nous désignerons par $\mathbf{U}_k = \langle \mathbf{A}_k, \mathbf{R}_k \rangle$ le système d’argumentation de l’agent k issu de l’univers de référence \mathbf{U} , lequel représente 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 1 (cont.). *Pour déterminer la culpabilité du prévenu, deux acteurs, 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 tous leurs arguments). Ainsi, pour chacun de ces acteurs, les arguments qu’il connaît constituent son propre système d’argumentation, tel que le montre la figure 3 où apparaissent :*

- \mathbf{U} l’univers de référence, dans lequel se trouvent les arguments A_0 à A_7 .
- \mathbf{U}_{proc} le système d’argumentation du procureur, contenant les arguments A_0, A_1, A_3, A_4, A_5 et A_6 .
- \mathbf{U}_{avoc} le système d’argumentation de l’avocat, contenant les arguments A_0, A_1, A_2, A_4 et A_7 .

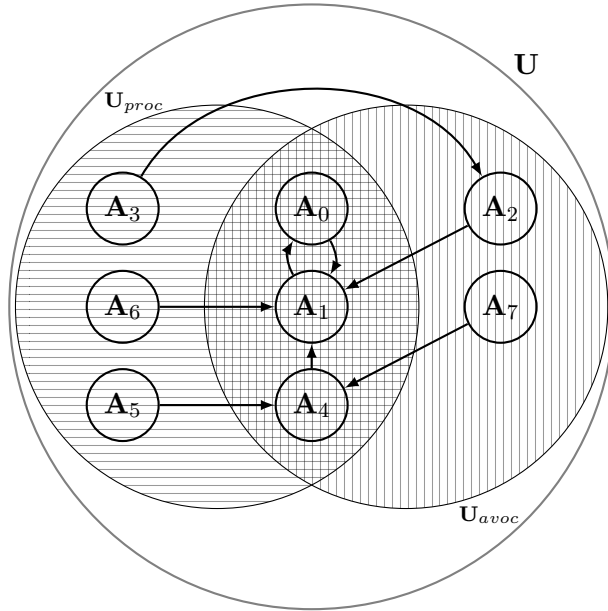
Notons qu’il est possible que certains arguments et interactions soient absents des systèmes d’argumentation de tous les agents. Ceci est représenté graphiquement

- *pour les arguments* : par des sommets du graphe n’étant contenus dans aucun système d’argumentation,

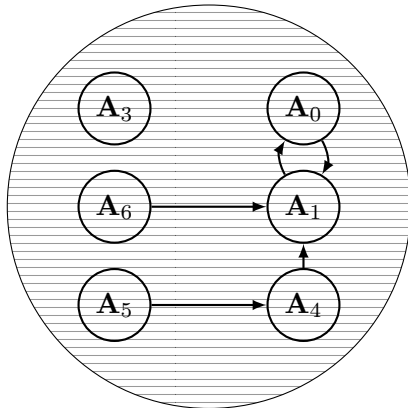
8. Dans ce document, nous utiliserons librement $\langle \mathbf{A}, \mathbf{R} \rangle$ ou \mathcal{G} pour faire allusion à un système d’argumentation. Par ailleurs, notons que dans le cas où un seul univers de référence est utilisé, nous parlerons de “système d’argumentation” en lieu et place de “système d’argumentation issu de \mathbf{U} ”.

- *pour les interactions* : par des arcs n'étant contenus entièrement dans aucun système d'argumentation (telle l'attaque de l'argument A_3 vers l'argument A_2 , qui n'est contenue entièrement ni dans \mathbf{U}_{proc} , ni dans \mathbf{U}_{avoc} sur la figure 3).

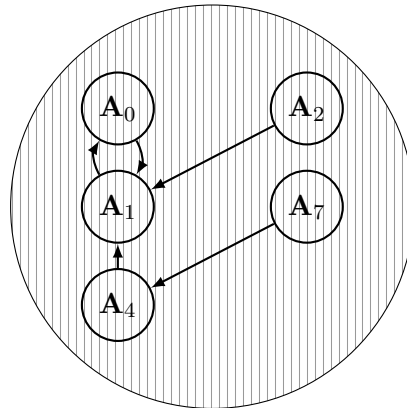
Dans le cas contraire, l'univers de référence se définit par l'union des arguments, et l'union des interactions les concernant, connus par les agents. Dans notre exemple, ceci aurait été le cas si l'argument A_3 n'attaquait pas l'argument A_2 .



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



(b) Système d'argumentation du procureur.



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

Figure 3 – Illustration de la notion de système d'argumentation.

4.3 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. (2010). Nous considérons des opérations de changement *élémentaires*, c'est-à-dire n'agissant que sur un argument ou une attaque appartenant à un univers particulier.

Les travaux de Cayrol et al. (2010) 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 4 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.

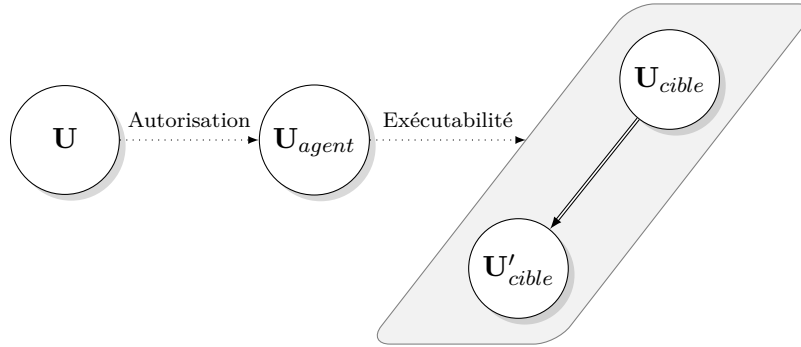


Figure 4 – *Illustration des contraintes sur les opérations. La zone grisée représente les travaux sur lesquels nous nous basons, c'est-à-dire la modification d'un système d'argumentation (système cible : \mathbf{U}_{cible}) menant à un nouveau système (système résultat : \mathbf{U}'_{cible}) par l'intermédiaire 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 (ce qui est représenté par le lien entre l'univers \mathbf{U} et \mathbf{U}_{agent}) les effectuant ; 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})).*

Ainsi, nous raffinons la notion d'opération élémentaire au sens de Cayrol et al. (2010) en quatre points : en premier lieu nous nous pencherons de manière abstraite sur cette notion en définissant sa *syntaxe* ; puis nous la

ramènerons à un agent pour déterminer si une opération est *autorisée* ou non pour celui-ci; nous la restreindrons ensuite en fonction du contexte, ce qui permettra de déterminer si une opération peut être *exécutée* par un agent sur un système d'argumentation cible; enfin, nous étudierons l'*impact* d'une opération sur un système d'argumentation.

Définition 8 (Opération élémentaire). Soit $\mathbf{U} = \langle \mathbf{A}_{\mathbf{U}}, \mathbf{R}_{\mathbf{U}} \rangle$ un univers de référence. Un triplet $\mathcal{O} = \langle op, arg, att \rangle$ est appelé opération élémentaire ssi $\mathcal{O}.arg \subseteq \mathbf{A}_{\mathbf{U}}$, $\mathcal{O}.att \subseteq \mathbf{R}_{\mathbf{U}}$ et

- soit $\mathcal{O}.op = \oplus_i^a$ avec $|\mathcal{O}.arg| = 1$ et $\forall (X, Y) \in \mathcal{O}.att$, $(X \neq Y)$ et $(X \in \mathcal{O}.arg$ ou $Y \in \mathcal{O}.arg)$ ou
- soit $\mathcal{O}.op = \ominus_i^a$ avec $|\mathcal{O}.arg| = 1$ et $\mathcal{O}.att = \emptyset$ ou
- soit $\mathcal{O}.op \in \{\oplus_i, \ominus_i\}$ avec $\mathcal{O}.arg = \emptyset$ et $|\mathcal{O}.att| = 1$.

L'ensemble des opérations élémentaires est noté \mathbf{O} .

Notons 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.

Exemple 2. À partir de l'univers de référence représenté par la figure 2, plusieurs opérations élémentaires sont syntaxiquement correctes. En voici une liste non exhaustive :

- $\langle \oplus_i, \emptyset, \{(A_3, A_2)\} \rangle$,
- $\langle \oplus_i, \emptyset, \{(A_5, A_4)\} \rangle$,
- $\langle \ominus_i, \emptyset, \{(A_6, A_1)\} \rangle$,
- $\langle \ominus_i, \emptyset, \{(A_7, A_4)\} \rangle$,
- $\langle \oplus_i^a, \{A_1\}, \emptyset \rangle$,
- $\langle \oplus_i^a, \{A_2\}, \{(A_2, A_1)\} \rangle$,
- $\langle \ominus_i^a, \{A_4\}, \emptyset \rangle$,
- $\langle \ominus_i^a, \{A_7\}, \emptyset \rangle$.

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 4 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 9 (Opération autorisée pour un agent). Soit un agent k et $\mathbf{U}_k = \langle \mathbf{A}_k, \mathbf{R}_k \rangle$ son système d'argumentation. Une opération élémentaire \mathcal{O} est autorisée pour k ssi

- $\mathcal{O}.arg \subseteq \mathbf{A}_k$ et

- $\mathcal{O}.att \subseteq \mathbf{R}_k$ ⁹.

L'ensemble des opérations autorisées pour k est noté \mathbf{O}_k .

Exemple 3. Parmi les quelques opérations élémentaires listées dans l'exemple 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 :

- $\langle \oplus_i, \emptyset, \{(A_3, A_2)\} \rangle$,
- $\langle \oplus_i^a, \{A_2\}, \{(A_2, A_1)\} \rangle$,
- $\langle \ominus_i, \emptyset, \{(A_7, A_4)\} \rangle$,
- $\langle \ominus_i^a, \{A_7\}, \emptyset \rangle$.

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 4 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}). 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 10 (Opération exécutable par un agent sur un système d'argumentation). Soit k un agent et $\mathcal{G} = \langle \mathbf{A}, \mathbf{R} \rangle$ un système d'argumentation quelconque. Une opération exécutable par k sur \mathcal{G} est une opération $\mathcal{O} = \langle op, arg, att \rangle$ autorisée pour k telle que :

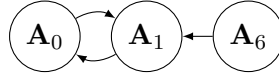
- si $\mathcal{O}.op = \oplus_i^a$, alors $\mathcal{O}.arg \not\subseteq \mathbf{A}$ et $\forall (X, Y) \in \mathcal{O}.att$, ($X \in \mathbf{A}$ ou $Y \in \mathbf{A}$).

9. Ainsi que nous l'avons déjà mentionné, dans le cas particulier d'un ajout 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.

- si $\mathcal{O}.op = \ominus_i^a$, alors $\mathcal{O}.arg \subseteq \mathbf{A}$.
- si $\mathcal{O}.op = \oplus_i$, alors $\mathcal{O}.att \not\subseteq \mathbf{R}$ et $\forall (X, Y) \in \mathcal{O}.att, (X \in \mathbf{A} \text{ et } Y \in \mathbf{A})$.
- si $\mathcal{O}.op = \ominus_i$, alors $\mathcal{O}.att \subseteq \mathbf{R}$.

L'ensemble des opérations exécutables par k sur \mathcal{G} est noté $\Gamma_k^{\mathcal{G}}$. Par définition, $\Gamma_k^{\mathcal{G}} \subseteq \mathcal{O}_k$.

Exemple 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. Pour cela, il n'a à sa disposition que quatre opérations autorisées sur celles vues dans l'exemple 2 :

- $\mathcal{O}_1 = \langle \oplus_i^a, \{A_1\}, \emptyset \rangle$,
- $\mathcal{O}_2 = \langle \ominus_i, \emptyset, \{(A_6, A_1)\} \rangle$,
- $\mathcal{O}_3 = \langle \ominus_i^a, \{A_4\}, \emptyset \rangle$,
- $\mathcal{O}_4 = \langle \oplus_i, \emptyset, \{(A_5, A_4)\} \rangle$.

Mais \mathcal{O}_1 n'est pas exécutable, puisque A_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 \mathcal{O}_3 et \mathcal{O}_4 ne peuvent être exécutées du fait de l'absence des arguments A_4 et A_5 dans le système des jurés. Il ne reste donc plus qu'à tenter de faire supprimer l'attaque de A_6 vers A_1 grâce à \mathcal{O}_2 , ce qui ne le fera pas gagner, mais le rapprochera de son objectif.

Une opération exécutable sur un système d'argumentation a un impact sur ce dernier. Cet impact est représenté sur la figure 4 par le lien entre \mathbf{U}_{cible} et le système résultat (\mathbf{U}'_{cible}).

Définition 11 (Impact d'opération exécutable par un agent sur un système d'argumentation). Soit k un agent, $\mathcal{G} = \langle \mathbf{A}, \mathbf{R} \rangle$ un système d'argumentation quelconque et $\mathcal{O} = \langle op, arg, att \rangle$ une opération exécutable par k sur \mathcal{G} . \mathcal{O} fournit un nouveau système d'argumentation $\mathcal{G}' = \mathcal{O}(\mathcal{G}) = \langle \mathbf{A}', \mathbf{R}' \rangle$ tel que :

- $\mathcal{O}.op = \oplus_i^a$:

$$\mathcal{G}' = \langle \mathbf{A} \cup \mathcal{O}.arg, \mathbf{R} \cup \mathcal{O}.att \rangle$$
- $\mathcal{O}.op = \ominus_i^a$:

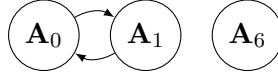
$$\mathcal{G}' = \langle \mathbf{A} \setminus \mathcal{O}.arg, \mathbf{R} \setminus \{(X, Y) \in \mathbf{R}_U \mid X \in \mathcal{O}.arg \text{ ou } Y \in \mathcal{O}.arg\} \rangle$$
- $\mathcal{O}.op = \oplus_i$:

$$\mathcal{G}' = \langle \mathbf{A}, \mathbf{R} \cup \mathcal{O}.att \rangle$$
- $\mathcal{O}.op = \ominus_i$:

$$\mathcal{G}' = \langle \mathbf{A}, \mathbf{R} \setminus \mathcal{O}.att \rangle$$

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 $\mathcal{O}.arg$.

Exemple 5. En exécutant l'opération $\langle \ominus_i, \emptyset, \{(A_6, A_1)\} \rangle$, le procureur a modifié le système d'argumentation des jurés en supprimant l'attaque de l'argument A_6 vers l'argument A_1 :



Ainsi, la menace de l'argument A_6 est pour le moment écartée. Mais ça ne suffit pas à faire accepter A_1 !

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 12 (Programme exécutable par un agent sur un système d'argumentation). Soit k un agent et \mathcal{G} un système d'argumentation quelconque. Un programme δ exécutable par k sur \mathcal{G} est une suite ordonnée finie de n opérations $(\mathcal{O}_1, \dots, \mathcal{O}_n)$ telle que :

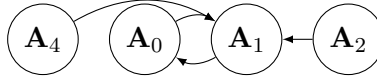
- Pour $n = 1$: \mathcal{O}_1 est exécutable par k sur \mathcal{G} (i.e. $\mathcal{O}_1 \in \Gamma_k^{\mathcal{G}}$). On a alors $\delta(\mathcal{G}) = \mathcal{O}_1(\mathcal{G})$.
- Pour $n > 1$: $(\mathcal{O}_1, \dots, \mathcal{O}_{n-1})$ est un programme δ' exécutable par k sur \mathcal{G} tel que $\delta'(\mathcal{G}) = \mathcal{G}'$ et $\mathcal{O}_n \in \Gamma_k^{\mathcal{G}'}$ est une opération exécutable par k sur \mathcal{G}' . On a alors $\delta(\mathcal{G}) = \mathcal{O}_n(\mathcal{G}')$.
- Par extension, une suite vide est aussi un programme. On a alors $\delta(\mathcal{G}) = \mathcal{G}$.

n est appelé cardinal de δ , noté $|\delta|$. Par abus de notation, on écrira $\mathcal{O} \in \delta$ pour $\exists \mathcal{O}_i \in \delta = (\mathcal{O}_1, \dots, \mathcal{O}_n)$ telle que $\mathcal{O} = \mathcal{O}_i$. On notera $\Delta_k^{\mathcal{G}}$ l'ensemble de tous les programmes exécutables par l'agent k sur \mathcal{G} .

Nous raffinons séance tenante cette définition pour y inclure la notion de programme réalisant un ensemble de buts.

Définition 13 (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 δ de k sur \mathcal{G} réalisant \mathcal{B}_k est un programme exécutable par k sur \mathcal{G} tel que $\delta(\mathcal{G}) = \mathcal{G}'$ et tous les buts $b \in \mathcal{B}_k$ sont satisfaits dans \mathcal{G}' . On notera $\Delta_{\mathcal{B}_k}^{\mathcal{G}}$ l'ensemble de tous les programmes de l'agent k sur \mathcal{G} réalisant l'ensemble de buts \mathcal{B}_k . Par définition, $\Delta_{\mathcal{B}_k}^{\mathcal{G}} \subseteq \Delta_k^{\mathcal{G}}$.

Exemple 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. 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 A_2 et s'est rendu compte que son argument A_3 pouvait le contrer. Il a désormais à sa disposition, entre autres choses, deux opérations qui collent parfaitement à la situation :

- $\mathcal{O}_3 = \langle \oplus_i^a, \{A_3\}, \{(A_3, A_2)\} \rangle$
- $\mathcal{O}_4 = \langle \oplus_i^a, \{A_5\}, \{(A_5, A_4)\} \rangle$

Grâce à ces deux opérations, il peut revenir dans la partie, en une prise de parole contrer les deux arguments de l'avocat et montrer qu'il ne se laissera pas faire.

5 Choix des opérations de changement

Nous avons jusqu'ici présenté l'intuition de notre travail ainsi que les bases théoriques et formelles sur lesquelles nous le construisons. Il est désormais temps d'étudier plus en profondeur le cœur de notre sujet, c'est-à-dire le choix à proprement parler d'un changement à effectuer, en fonction du but à satisfaire. Pour cela, nous présenterons en premier lieu les différentes étapes d'un tel choix, puis nous étudierons les critères pouvant guider ce choix.

5.1 D eroulement

Dans ce travail, nous nous int eressons principalement au changement d’un syst eme d’argumentation, et pourquoi un tel changement est op er e. Ce sujet est bien repr esent e par la deuxi eme  etape du protocole pr esent e  a la section 3.3. Aussi, nous nous concentrons sur cette derni ere, le reste  tant laiss e pour des travaux futurs. Nous  tudions ici de quelle fa on un agent choisit un programme   ex ecuter en fonction de son propre syst eme d’argumentation et du syst eme cible.

Par ailleurs, nous n’ tudierons ici que le cas concernant des programmes dits simples, c’est- a-dire des programmes n’ayant qu’une seule op eration ; des programmes plus complexes, contenant plusieurs op erations  l ementaires, sont  galement laiss es pour des travaux futurs. Ceci nous permettra dans un premier temps d’aborder de mani ere simplifi ee la notion de *propri ete du changement*, notion repr esentant les modifications que peut subir un syst eme d’argumentation par l’ex ecution d’une op eration de changement ; le lecteur pourra se reporter   Bisquert et al. (2012c) pour de plus amples d etails.

Soit un agent k ayant un ensemble de buts \mathcal{B}_k et \mathbf{P} l’ensemble des propri etes du changement. Pour d eterminer le programme r ealisant \mathcal{B}_k   ex ecuter, k suit la proc edure suivante :

1. L’agent k parcourt \mathbf{P} et en extrait les propri etes qui concernent une op eration qui lui est possible de faire, c’est- a-dire o u il est possible d’apparier ce qui est mis en jeu dans la propri ete avec ce qu’il connait, pour cr eer l’ensemble \mathbf{P}_{aut} . Ce nouvel ensemble regroupe les propri etes du changement correspondant   un sous-ensemble des op erations autoris ees pour k .
2. De \mathbf{P}_{aut} , k ne garde que les propri etes o u la condition sur \mathcal{G} (si elle existe) correspond au syst eme d’argumentation cible pour donner naissance   l’ensemble \mathbf{P}_{exe} . Cet ensemble regroupe les propri etes du changement correspondant   un sous-ensemble des op erations ex ecutables par k sur le syst eme cible.
3. Enfin, k raffine l’ensemble \mathbf{P}_{exe} en ne gardant que les propri etes indiquant que les buts appartenant   \mathcal{B}_k sont satisfaits dans \mathcal{G}' . Cet ensemble, nomm e \mathbf{P}_{rea} , r epertorie les propri etes du changement correspondant   un sous-ensemble des programmes (simples) de k sur \mathcal{G} r ealisant \mathcal{B}_k .

Notons que dans les pr ec edents travaux, par exemple Cayrol et al. (2010), la premi ere  etape de la proc edure pr ec edente n’aurait pas pu  tre faite (puisque la notion d’op eration autoris ee n’y  tait pas utilis ee). Ceci ne remet

pas en cause les travaux qui ont été faits alors, notamment au regard des diverses propriétés qui ont pu être démontrées. En effet, cette absence de la notion d'opération autorisée signifie implicitement que toutes les opérations sont autorisées pour un agent non spécifié. Si nous ne considérons que les propriétés pouvant "s'appliquer" au système d'un agent k , nous procédons en fait à une instanciation du cadre général en ajoutant des restrictions sur le domaine de définition, ce qui ne modifie en rien la validité des travaux antérieurs.

Cette procédure nous permet donc d'obtenir un ensemble de propriétés adéquates au regard de la situation qui nous permettra de trouver des programmes réalisant les buts de l'agent. Mais, au sein de cet ensemble, comment choisir un *meilleur* programme? La section suivante se propose de répondre à cette question en présentant une approche permettant de déterminer le changement optimal au regard d'un critère particulier.

5.2 Changement optimal

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 une même modification. 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.

Nous pouvons distinguer trois critères principaux, qu'il est possible de raffiner en critères plus précis, portant sur :

- le coût de programmes,
- la distance structurelle de systèmes d'argumentation,
- la distance sémantique de 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 (cont.). *Le procureur a donc choisi d'utiliser le programme $(\langle \oplus_i^a, \{A_3\}, \{(A_3, A_2)\} \rangle, \langle \oplus_i^a, \{A_5\}, \{(A_5, A_4)\} \rangle)$. Mais comment en est-il venu à faire ce choix ? Qu'est-ce qui l'a poussé à énoncer cela plutôt qu'autre chose ?*

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 $|E'_{A_1}| \neq 0$).

Ainsi, divers choix possibles s'offraient à lui ; la table 2 en propose une partie en considérant que l'opération $\langle \oplus_i^a, \{A_3\}, \{(A_3, A_2)\} \rangle$, notée \mathcal{O}_1 , est présente dans chacun des programmes étudiés.

En outre, les systèmes d'argumentation qui seraient obtenus par l'utilisation de ces programmes sont illustrés par la figure 5.

Programmes	Extensions des systèmes obtenus
δ_1 $(\mathcal{O}_1, \langle \oplus_i^a, \{A_5\}, \{(A_5, A_4)\} \rangle)$	$\{\{A_0, A_5\}, \{A_1, A_5\}\}$
δ_2 $(\mathcal{O}_1, \langle \oplus_i^a, \{A_5\}, \{(A_5, A_4)\} \rangle, \langle \ominus_i, \emptyset, \{(A_0, A_1)\} \rangle)$	$\{\{A_1, A_5\}\}$
δ_3 $(\mathcal{O}_1, \langle \oplus_i^a, \{A_5\}, \{(A_5, A_4)\} \rangle, \langle \ominus_i^a, \{A_4\}, \emptyset \rangle)$	$\{\{A_0, A_5\}, \{A_1, A_5\}\}$
δ_4 $(\mathcal{O}_1, \langle \oplus_i^a, \{A_5\}, \{(A_5, A_4)\} \rangle, \langle \ominus_i^a, \{A_5\}, \emptyset \rangle, \langle \ominus_i^a, \{A_4\}, \emptyset \rangle)$	$\{\{A_0\}, \{A_1\}\}$
δ_5 $(\mathcal{O}_1, \langle \ominus_i^a, \{A_4\}, \emptyset \rangle)$	$\{\{A_0\}, \{A_1\}\}$
δ_6 $(\mathcal{O}_1, \langle \ominus_i, \emptyset, \{(A_4, A_1)\} \rangle)$	$\{\{A_0, A_4\}, \{A_1, A_4\}\}$
δ_7 $(\mathcal{O}_1, \langle \ominus_i, \emptyset, \{(A_4, A_1)\} \rangle, \langle \ominus_i, \emptyset, \{(A_0, A_1)\} \rangle)$	$\{\{A_1, A_4\}\}$
δ_8 $(\mathcal{O}_1, \langle \ominus_i^a, \{A_4\}, \emptyset \rangle, \langle \ominus_i, \emptyset, \{(A_0, A_1)\} \rangle)$	$\{\{A_1\}\}$

Table 2 – *Liste des divers programmes que le procureur peut exécuter.*

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 $\delta(\mathcal{G})$, il nous paraît plus intuitif dans ce cas de définir une valeur associée au critère grâce au couple $(\mathcal{G}, \delta(\mathcal{G}))$.

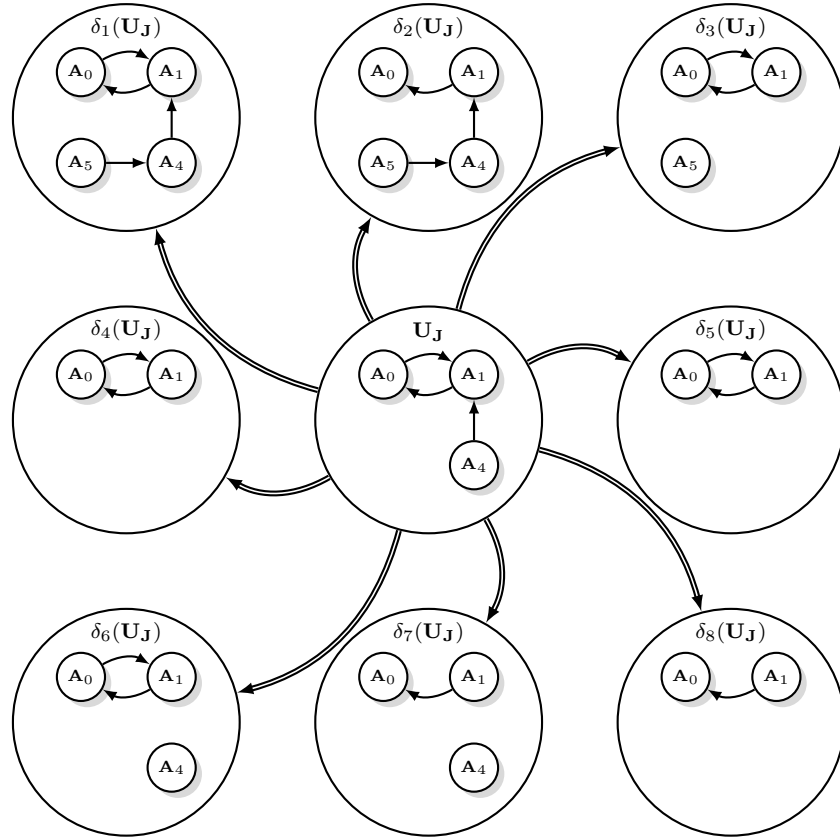


Figure 5 – Illustration des différentes (et arbitraires) possibilités de changement du système d’argumentation des jurés (U_J).

Voyons maintenant quelques critères possibles.

5.2.1 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, qu’il peut par ailleurs utiliser. Par exemple, il est possible d’utiliser une fonction de coût d’une opération (élémentaire) pour un agent.

Définition 14 (Coût d’une opération). Soit $Coût$ une fonction telle que

$$\begin{aligned} Coût : Agents \times \mathbf{O} &\rightarrow \mathbb{N} \cup \{\infty\} \\ (k, \mathcal{O}) &\mapsto \infty \text{ ssi } \mathcal{O} \notin \mathbf{O}_k \end{aligned}$$

où $Agents$ est l’ensemble des agents, \mathbf{O} est l’ensemble des opérations élémentaires et \mathbf{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, tel que 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 toute opération a le même coût. Ce dernier pourrait néanmoins être modulé grâce à la fonction $Coût$ définie précédemment.

Exemple 7. Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table 2 :

- $|\delta_1| = 1$
- $|\delta_2| = 2$
- $|\delta_3| = 2$
- $|\delta_4| = 3$
- $|\delta_5| = 1$
- $|\delta_6| = 1$
- $|\delta_7| = 2$
- $|\delta_8| = 2$

5.2.2 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 15. La distance d'arguments perdus entre deux systèmes d'argumentation $\langle \mathbf{A}, \mathbf{R} \rangle$ et $\langle \mathbf{A}', \mathbf{R}' \rangle$ est un entier naturel défini par la fonction suivante :

$$\mathcal{F}_{pa} : \quad \mathbf{G}_{\mathbf{U}} \times \mathbf{G}_{\mathbf{U}} \rightarrow \mathbb{N}$$

$$(\langle \mathbf{A}, \mathbf{R} \rangle, \langle \mathbf{A}', \mathbf{R}' \rangle) \mapsto |\mathbf{A} \setminus \mathbf{A}'|.$$

Exemple 8. Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table 2 :

- $\mathcal{F}_{pa}(\mathbf{U}_{\mathbf{J}}, \delta_1(\mathbf{U}_{\mathbf{J}})) = 0$
- $\mathcal{F}_{pa}(\mathbf{U}_{\mathbf{J}}, \delta_2(\mathbf{U}_{\mathbf{J}})) = 0$
- $\mathcal{F}_{pa}(\mathbf{U}_{\mathbf{J}}, \delta_3(\mathbf{U}_{\mathbf{J}})) = 1$
- $\mathcal{F}_{pa}(\mathbf{U}_{\mathbf{J}}, \delta_4(\mathbf{U}_{\mathbf{J}})) = 1$
- $\mathcal{F}_{pa}(\mathbf{U}_{\mathbf{J}}, \delta_5(\mathbf{U}_{\mathbf{J}})) = 1$
- $\mathcal{F}_{pa}(\mathbf{U}_{\mathbf{J}}, \delta_6(\mathbf{U}_{\mathbf{J}})) = 0$
- $\mathcal{F}_{pa}(\mathbf{U}_{\mathbf{J}}, \delta_7(\mathbf{U}_{\mathbf{J}})) = 0$
- $\mathcal{F}_{pa}(\mathbf{U}_{\mathbf{J}}, \delta_8(\mathbf{U}_{\mathbf{J}})) = 1$

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 16. La distance d'arguments gagnés entre deux systèmes d'argumentation $\langle \mathbf{A}, \mathbf{R} \rangle$ et $\langle \mathbf{A}', \mathbf{R}' \rangle$ est un entier naturel défini par la fonction suivante :

$$\mathcal{F}_{ga} : \quad \mathbf{G}_{\mathbf{U}} \times \mathbf{G}_{\mathbf{U}} \rightarrow \mathbb{N}$$

$$(\langle \mathbf{A}, \mathbf{R} \rangle, \langle \mathbf{A}', \mathbf{R}' \rangle) \mapsto |\mathbf{A}' \setminus \mathbf{A}|.$$

Exemple 9. Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table 2 :

- $\mathcal{F}_{ga}(\mathbf{U}_{\mathbf{J}}, \delta_1(\mathbf{U}_{\mathbf{J}})) = 1$
- $\mathcal{F}_{ga}(\mathbf{U}_{\mathbf{J}}, \delta_2(\mathbf{U}_{\mathbf{J}})) = 1$
- $\mathcal{F}_{ga}(\mathbf{U}_{\mathbf{J}}, \delta_3(\mathbf{U}_{\mathbf{J}})) = 1$
- $\mathcal{F}_{ga}(\mathbf{U}_{\mathbf{J}}, \delta_4(\mathbf{U}_{\mathbf{J}})) = 0$
- $\mathcal{F}_{ga}(\mathbf{U}_{\mathbf{J}}, \delta_5(\mathbf{U}_{\mathbf{J}})) = 0$
- $\mathcal{F}_{ga}(\mathbf{U}_{\mathbf{J}}, \delta_6(\mathbf{U}_{\mathbf{J}})) = 0$
- $\mathcal{F}_{ga}(\mathbf{U}_{\mathbf{J}}, \delta_7(\mathbf{U}_{\mathbf{J}})) = 0$
- $\mathcal{F}_{ga}(\mathbf{U}_{\mathbf{J}}, \delta_8(\mathbf{U}_{\mathbf{J}})) = 0$

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 17. *La distance d'attaques perdues entre deux systèmes d'argumentation $\langle \mathbf{A}, \mathbf{R} \rangle$ et $\langle \mathbf{A}', \mathbf{R}' \rangle$ est un entier naturel défini par la fonction suivante :*

$$\mathcal{F}_{pi} : \quad \mathbf{G}_U \times \mathbf{G}_U \rightarrow \mathbb{N}$$

$$(\langle \mathbf{A}, \mathbf{R} \rangle, \langle \mathbf{A}', \mathbf{R}' \rangle) \mapsto |\mathbf{R} \setminus \mathbf{R}'|.$$

Exemple 10. *Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table 2 :*

- $\mathcal{F}_{pi}(\mathbf{U}_J, \delta_1(\mathbf{U}_J)) = 0$
- $\mathcal{F}_{pi}(\mathbf{U}_J, \delta_2(\mathbf{U}_J)) = 1$
- $\mathcal{F}_{pi}(\mathbf{U}_J, \delta_3(\mathbf{U}_J)) = 1$
- $\mathcal{F}_{pi}(\mathbf{U}_J, \delta_4(\mathbf{U}_J)) = 1$
- $\mathcal{F}_{pi}(\mathbf{U}_J, \delta_5(\mathbf{U}_J)) = 1$
- $\mathcal{F}_{pi}(\mathbf{U}_J, \delta_6(\mathbf{U}_J)) = 1$
- $\mathcal{F}_{pi}(\mathbf{U}_J, \delta_7(\mathbf{U}_J)) = 2$
- $\mathcal{F}_{pi}(\mathbf{U}_J, \delta_8(\mathbf{U}_J)) = 2$

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 18. *La distance d'attaques gagnées entre deux systèmes d'argumentation $\langle \mathbf{A}, \mathbf{R} \rangle$ et $\langle \mathbf{A}', \mathbf{R}' \rangle$ est un entier naturel défini par la fonction suivante :*

$$\mathcal{F}_{gi} : \quad \mathbf{G}_U \times \mathbf{G}_U \rightarrow \mathbb{N}$$

$$(\langle \mathbf{A}, \mathbf{R} \rangle, \langle \mathbf{A}', \mathbf{R}' \rangle) \mapsto |\mathbf{R}' \setminus \mathbf{R}|.$$

Exemple 11. *Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table 2 :*

- $\mathcal{F}_{gi}(\mathbf{U}_J, \delta_1(\mathbf{U}_J)) = 1$
- $\mathcal{F}_{gi}(\mathbf{U}_J, \delta_2(\mathbf{U}_J)) = 1$
- $\mathcal{F}_{gi}(\mathbf{U}_J, \delta_3(\mathbf{U}_J)) = 0$
- $\mathcal{F}_{gi}(\mathbf{U}_J, \delta_4(\mathbf{U}_J)) = 0$
- $\mathcal{F}_{gi}(\mathbf{U}_J, \delta_5(\mathbf{U}_J)) = 0$
- $\mathcal{F}_{gi}(\mathbf{U}_J, \delta_6(\mathbf{U}_J)) = 0$
- $\mathcal{F}_{gi}(\mathbf{U}_J, \delta_7(\mathbf{U}_J)) = 0$
- $\mathcal{F}_{gi}(\mathbf{U}_J, \delta_8(\mathbf{U}_J)) = 0$

5.2.3 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 crêdulement 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 crêdulement dans le premier système et rejetés dans le second, c’est-à-dire le nombre d’arguments acceptés crêdulement qui ont été perdus dans le changement.

Définition 19. *La distance d’arguments acceptés crêdulement perdus entre deux systèmes d’argumentation $\langle \mathbf{A}, \mathbf{R} \rangle$ et $\langle \mathbf{A}', \mathbf{R}' \rangle$ est un entier naturel défini par la fonction suivante :*

$$\mathcal{F}_{pe} : \quad \mathbf{G}_U \times \mathbf{G}_U \rightarrow \mathbb{N}$$

$$(\langle \mathbf{A}, \mathbf{R} \rangle, \langle \mathbf{A}', \mathbf{R}' \rangle) \mapsto \left| \bigcup_{\mathcal{E} \in \mathbf{E}} \mathcal{E} \setminus \bigcup_{\mathcal{E}' \in \mathbf{E}'} \mathcal{E}' \right|.$$

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 12. *Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table 2 :*

- $\mathcal{F}_{pe}(\mathbf{U}_J, \delta_1(\mathbf{U}_J)) = 1$
- $\mathcal{F}_{pe}(\mathbf{U}_J, \delta_2(\mathbf{U}_J)) = 2$
- $\mathcal{F}_{pe}(\mathbf{U}_J, \delta_3(\mathbf{U}_J)) = 1$
- $\mathcal{F}_{pe}(\mathbf{U}_J, \delta_4(\mathbf{U}_J)) = 1$
- $\mathcal{F}_{pe}(\mathbf{U}_J, \delta_5(\mathbf{U}_J)) = 1$
- $\mathcal{F}_{pe}(\mathbf{U}_J, \delta_6(\mathbf{U}_J)) = 0$
- $\mathcal{F}_{pe}(\mathbf{U}_J, \delta_7(\mathbf{U}_J)) = 1$
- $\mathcal{F}_{pe}(\mathbf{U}_J, \delta_8(\mathbf{U}_J)) = 2$

Distance de gain d’acceptabilité. Ce critère, noté ge se base, à l’inverse du précédent, sur le nombre d’arguments étant rejetés avant le changement et étant acceptés crêdulement 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 crûdement dans le second, c'est-à-dire le nombre d'arguments acceptés crûdement qui ont été gagnés dans le changement.

Définition 20. *La distance d'arguments acceptés crûdement gagnés entre deux systèmes d'argumentation $\langle \mathbf{A}, \mathbf{R} \rangle$ et $\langle \mathbf{A}', \mathbf{R}' \rangle$ est un entier naturel défini par la fonction suivante :*

$$\mathcal{F}_{ge} : \quad \mathbf{G}_{\mathbf{U}} \times \mathbf{G}_{\mathbf{U}} \rightarrow \mathbb{N}$$

$$(\langle \mathbf{A}, \mathbf{R} \rangle, \langle \mathbf{A}', \mathbf{R}' \rangle) \mapsto \left| \bigcup_{\mathcal{E}' \in \mathbf{E}'} \mathcal{E}' \setminus \bigcup_{\mathcal{E} \in \mathbf{E}} \mathcal{E} \right|.$$

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 13. *Si le procureur avait utilisé ce critère, il aurait obtenu les valeurs suivantes pour les programmes listés dans la table 2 :*

- $\mathcal{F}_{ge}(\mathbf{U}_{\mathbf{J}}, \delta_1(\mathbf{U}_{\mathbf{J}})) = 1$
- $\mathcal{F}_{ge}(\mathbf{U}_{\mathbf{J}}, \delta_2(\mathbf{U}_{\mathbf{J}})) = 2$
- $\mathcal{F}_{ge}(\mathbf{U}_{\mathbf{J}}, \delta_3(\mathbf{U}_{\mathbf{J}})) = 2$
- $\mathcal{F}_{ge}(\mathbf{U}_{\mathbf{J}}, \delta_4(\mathbf{U}_{\mathbf{J}})) = 1$
- $\mathcal{F}_{ge}(\mathbf{U}_{\mathbf{J}}, \delta_5(\mathbf{U}_{\mathbf{J}})) = 1$
- $\mathcal{F}_{ge}(\mathbf{U}_{\mathbf{J}}, \delta_6(\mathbf{U}_{\mathbf{J}})) = 1$
- $\mathcal{F}_{ge}(\mathbf{U}_{\mathbf{J}}, \delta_7(\mathbf{U}_{\mathbf{J}})) = 1$
- $\mathcal{F}_{ge}(\mathbf{U}_{\mathbf{J}}, \delta_8(\mathbf{U}_{\mathbf{J}})) = 1$

5.2.4 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 and 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, ces derniers ne pouvant alors pas nécessairement être tous comparés).

Les critères définis ci-avant sont du premier type. 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 δ_1 est meilleur qu'un programme δ_2 au regard d'un critère particulier si δ_1 a une valeur plus basse que δ_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 21. *Soit un agent $k \in \text{Agents}$, $\mathcal{G} \in \mathbf{G}_{\mathbf{U}}$ un système d'argumentation quelconque, $\omega \in \{t, pa, ga, pi, gi, pe, ge\}$ un critère de comparaison, \mathcal{F}_ω la fonction correspondant au critère ω , $\delta_1 \in \Delta_k^{\mathcal{G}}$ et $\delta_2 \in \Delta_k^{\mathcal{G}}$ deux programmes exécutables par k sur \mathcal{G} . Soit \preceq^ω la relation de préférence sur Δ_k telle que $\delta_1 \preceq^\omega \delta_2$ (δ_1 est au moins aussi bon que δ_2) ssi*

$$\mathcal{F}_\omega(\mathcal{G}, \delta_1(\mathcal{G})) \leq \mathcal{F}_\omega(\mathcal{G}, \delta_2(\mathcal{G})).$$

Notons que les définitions des relations \approx^ω et \prec^ω se font de manière classique à partir de \preceq^ω .

Exemple 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 désire pas que privilégier le critère portant sur la taille des programmes (t), il obtiendrait :*

$$\begin{array}{l} \delta_1 \approx^t \delta_5 \approx^t \delta_6 \\ \prec^t \delta_2 \approx^t \delta_3 \approx^t \delta_7 \approx^t \delta_8 \\ \prec^t \delta_4 \end{array}$$

Les trois meilleurs changements au regard du critère t sont donc les programmes δ_1 , δ_5 et δ_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 :

$$\begin{array}{c} \delta_6 \\ \prec^{pe} \delta_1 \approx^{pe} \delta_3 \approx^{pe} \delta_5 \approx^{pe} \delta_7 \\ \prec^{pe} \delta_2 \approx^{pe} \delta_8 \end{array}$$

Le meilleur changement au regard du critère pe aurait donc été le programme δ_6 .

5.2.5 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 \mathbf{A} et \mathbf{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 and 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).¹⁰ La comparaison de deux programmes est alors similaire à précédemment, les valeurs associées aux programmes étant juste des additions de valeurs suivant plusieurs critères. Nous obtenons ainsi :

Définition 22. *Soit un agent $k \in Agents$, $\mathcal{G} \in \mathbf{GU}$ un système d’argumentation quelconque, $\Omega \subseteq \{pa, ga, pi, gi, pe, ge\}$ un ensemble non vide de critères, \mathcal{F}_{ω_i} la fonction correspondant au critère $\omega_i \in \Omega$ et $\delta \in \Delta_k$ un programme. La valeur globale de δ est égale à :*

¹⁰. 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.

$$\sum_{\omega_i \in \Omega} \mathcal{F}_{\omega_i}(\mathcal{G}, \delta(\mathcal{G})).$$

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 n relations ω_i , on cherche une relation Ω qui synthétise les relations ω_i .

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 23. *Soit un agent $k \in Agents$, $\Omega \subseteq \{t, pa, ga, pi, gi, pe, ge\}$ un ensemble non vide de n critères, $\delta_1 \in \Delta_k$ et $\delta_2 \in \Delta_k$ deux programmes. On définit $\preceq^{\Omega-par}$ sur Δ_k par $\delta_1 \preceq^{\Omega-par} \delta_2$ ssi*

$$\forall i \leq n, \delta_1 \preceq^{\omega_i} \delta_2.$$

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 :

Définition 24. *Soit un agent $k \in Agents$, Ω une suite ordonnée de n critères distincts telle que $\forall \omega_i \in \Omega, \omega_i \in \{t, pa, ga, pi, gi, pe, ge\}$, $\delta_1 \in \Delta_k$ et $\delta_2 \in \Delta_k$ deux programmes. On définit $\prec^{\Omega-lex}$ sur Δ_k par $\delta_1 \prec^{\Omega-lex} \delta_2$ ssi*

$$\exists i \leq n \text{ t.q. } \delta_1 \prec^{\omega_i} \delta_2 \text{ et}$$

$$\forall j < i, \delta_1 \approx^{\omega_j} \delta_2.$$

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 25. Soit un agent $k \in \text{Agents}$, $\Omega \subseteq \{t, pa, ga, pi, gi, pe, ge\}$ un ensemble non vide de critères, $\delta_1 \in \Delta_k$ et $\delta_2 \in \Delta_k$ deux programmes. On définit $\preceq^{\Omega\text{-maj}}$ sur Δ_k par $\delta_1 \preceq^{\Omega\text{-maj}} \delta_2$ ssi

$$\begin{aligned} & |\{\omega_i \in \Omega \mid \delta_1 \preceq^{\omega_i} \delta_2\}| \\ & \geq \\ & |\{\omega_i \in \Omega \mid \delta_2 \preceq^{\omega_i} \delta_1\}|. \end{aligned}$$

Exemple 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.

Ainsi, le procureur obtient :

$$\begin{array}{ccccccc} \delta_1 & \prec^{(t,pa,pi)} & \delta_6 & \prec^{(t,pa,pi)} & \delta_5 & \prec^{(t,pa,pi)} & \delta_2 \\ \prec^{(t,pa,pi)} & \delta_7 & \prec^{(t,pa,pi)} & \delta_3 & \prec^{(t,pa,pi)} & \delta_8 & \prec^{(t,pa,pi)} & \delta_4 \end{array}$$

Le meilleur changement au regard de la suite de critères (t, pa, pi) combinés lexicographiquement est donc le programme δ_1 , et c'est donc ce programme qu'il va utiliser pour répondre à son adversaire.

6 Conclusion

Dans ce document, 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 prises dans un système d'argumentation "source" (représentant les connaissances de cet agent). Nous avons proposé une intuition "concrète" de ce travail à travers l'exemple d'une audience au tribunal, en expliquant à quelles situations les notions théoriques pourraient correspondre. Nous avons de plus présenté un protocole permettant de gérer l'organisation globale d'un dialogue de persuasion d'un auditoire, et nous avons étudié plus profondément le cas du choix d'un ou plusieurs arguments par un agent. Ceci nous a permis, en conclusion, d'aborder 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.

Ce travail ouvre de nombreuses perspectives :

- 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 de buts d'un agent dans différentes situations (une ou plusieurs opérations, prise en compte des critères d'optimalité, etc.).
- 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 : existe-t-il un lien entre optimalité et exécutabilité ? Entre équivalence et optimalité ?
- Nous nous intéressons également à la réécriture d'une opération élémentaire en une ou plusieurs opérations "basiques", c'est-à-dire ne concernant qu'un et un seul argument ou une et une seule attaque, de manière à permettre une définition plus homogène des opérations.
- Enfin, à plus long terme, nous souhaitons porter notre attention sur la notion de stratégie et son impact en termes de comparaison de programme : 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, etc. Une étude des possibilités de planification permettra de plus de considérer la génération automatique de programmes contenant plusieurs opérations.

A Exemples de buts et autres ambitions

Au sein de cette annexe, nous souhaitons présenter des exemples de buts et les intuitions possibles qui leurs sont associées. Nous débuterons par les buts absolus et finirons par les buts relatifs.

Notons que les buts et intuitions présentés ici se basent sur la typologie des propriétés du changement présentée dans Bisquert et al. (2012c). 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. Nous laissons les deux restants pour des travaux futurs.

A.1 Buts absolus

Pour rappel, un but absolu prend uniquement en compte le système d'argumentation résultat (c'est-à-dire le système obtenu après modification du système cible) et peut donc servir à représenter un désir durant toute la discussion.

A.1.1 Niveau de l'ensemble des extensions

$|\mathbf{E}'| = 0$ ou $|\mathbf{E}'| = 1$, $\mathcal{E}' = \emptyset$. Georges 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'escargot éclopé. Pour cela, il doit faire en sorte qu'aucune extension n'existe, ou que l'extension soit vide.

$|\mathbf{E}'| = 1$, $\mathcal{E}' \neq \emptyset$. Jean est irrité. Il se rend bien compte que Georges 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.

$|\mathbf{E}'| > 1$. Georges ne s'attendait pas à ça : Jean 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.

A.1.2 Niveau d'un argument particulier

$|\mathbf{E}'_X| = 0$. Georges 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.

$|\mathbf{E}'_X| \neq 0, |\mathbf{E}'_X| = |\mathbf{E}'|$. Jean 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. Georges 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, Georges en a bien conscience. Cette affaire est médiatiquement capitale, et le ministère public n'a pas le droit à l'erreur. Il doit être indubitable, ne pas laisser l'ombre d'un doute. *Voilà ma chance!*, se dit Georges. Tout ce qu'il a à faire, c'est d'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.

A.2 Buts relatifs

Pour rappel, un but relatif prend en compte à la fois le système d'argumentation cible et le système d'argumentation résultat ; il peut donc servir

à représenter un désir localisé à un moment précis de la discussion.

A.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. Georges 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. Georges 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. Georges 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. Georges 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.

A.2.2 Niveau d'ensembles d'arguments

$\forall \mathcal{E}_i \in \mathbf{E}, \exists \mathcal{E}'_j \in \mathbf{E}', \mathcal{E}_i \subseteq \mathcal{E}'_j$. Georges 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. Georges 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, Georges 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 Georges et Jean fait rage. Les arguments fusent des deux côtés et la cadence s'intensifie. Georges s'est laissé avoir par cette hausse de rythme et a fait 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.

$\bigcup_{1 \leq i \leq |\mathbf{E}|} \mathcal{E}_i \subseteq \bigcup_{1 \leq j \leq |\mathbf{E}'|} \mathcal{E}'_j$. Jean 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 crûdement avant son intervention doit le rester après qu'il ait parlé.

$\bigcup_{1 \leq j \leq |\mathbf{E}'|} \mathcal{E}'_j \subseteq \bigcup_{1 \leq i \leq |\mathbf{E}|} \mathcal{E}_i$. Georges 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 crûde, même un des siens. Pour cela, il doit faire en sorte que tout argument accepté crûdement 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 Jean 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.

$\bigcap_{1 \leq j \leq |\mathbf{E}'|} \mathcal{E}'_j \subseteq \bigcap_{1 \leq i \leq |\mathbf{E}|} \mathcal{E}_i$. Georges voit bien le manège de son rival. Pour le contrecarrer, Georges 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 crûdement ou rejeter.

Références

- Baumann, R. and 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.
- Bisquert, P., Cayrol, C., de Saint-Cyr, F. D., and Lagasquie-Schiex, M.-C. (2012a). Duality between addition and removal. In Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B., and Yager, R. R., editors, *Advances on Computational Intelligence*, volume 297 of *Communications in Computer and Information Science*, pages 219–229. Springer.
- Bisquert, P., Cayrol, C., Dupin de Saint Cyr Bannay, F., and Lagasquie-Schiex, M.-C. (2012b). Changement dans un système d’argumentation : suppression d’un argument. *Revue d’Intelligence Artificielle*, 26(3/2012) :225–254.
- Bisquert, P., Cayrol, C., Dupin de Saint Cyr Bannay, F., and Lagasquie-Schiex, M.-C. (2012c). Le changement en argumentation : Typologie et dualités. Rapport de recherche RR- -2012-2- -FR, IRIT, Université Paul Sabatier, Toulouse.
- Boella, G., Kaci, S., and van der Torre, L. (2009a). Dynamics in argumentation with single extensions : Abstraction principles and the grounded extension. In *Proc. of ECSQARU (LNAI 5590)*, pages 107–118.
- Boella, G., Kaci, S., and van der Torre, L. (2009b). Dynamics in argumentation with single extensions : Attack refinement and the grounded extension. In *Proc. of AAMAS*, pages 1213–1214.
- Cayrol, C., Dupin de Saint Cyr, F., and Lagasquie-Schiex, M.-C. (2010). Change in abstract argumentation frameworks : Adding an argument. *Journal of Artificial Intelligence Research*, 38 :49–84.
- 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.
- Grabisch, M. and Perny, P. (2003). Agrégation multicritère. In Bouchon-Meunier, B. and Marsala, C., editors, *Logique floue, principes, aide à la décision*, pages 81–120. Hermès.

Liao, B., Jin, L., and Koons, R. C. (2011). Dynamics of argumentation systems : A division-based method. *Artificial Intelligence*, 175(11) :1790 – 1814.

Moguillansky, M. O., Rotstein, N. D., Falappa, M. A., García, A. J., and Simari, G. R. (2010). Argument theory change through defeater activation. In *Proceeding of the 2010 conference on Computational Models of Argument : Proceedings of COMMA 2010*, pages 359–366, Amsterdam, The Netherlands, The Netherlands. IOS Press.