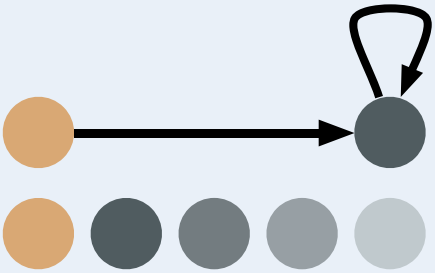


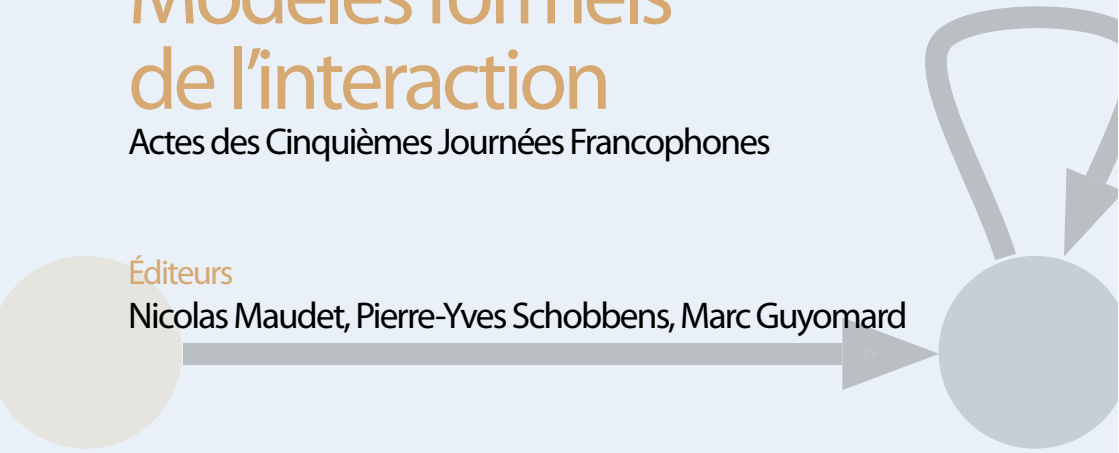
# MFI'09

# Modèles formels de l'interaction

## Actes des Cinquièmes Journées Francophones

Éditeurs

Nicolas Maudet, Pierre-Yves Schobbens, Marc Guyomard



Lannion, Juin 2009

ENSSAT LANNION — IRISA — UNIVERSITÉ DE RENNES 1 — INRIA — Côtes d'Armor

**MFI'09 : Modèles formels de l'interaction**

L'interactivité est une tendance majeure des systèmes informatiques actuels et un important champ de recherche. Elle se décline sous plusieurs aspects :

- interaction entre utilisateurs et systèmes informatiques,
- interaction entre entités informatiques autonomes (agents) interconnectées sur un réseau local ou sur la «toile», en vue de coopérer, de concourir ou tout simplement de coexister.
- sans oublier l'intégration de ces deux aspects dans les divers agents conversationnels, agents de recherche, assistants personnels...

Ces tendances sont à l'origine d'un besoin croissant de modèles formels de l'interaction, intégrant les règles, normes et protocoles divers, ainsi que les connaissances spécifiques des agents (en particulier sur les autres agents – humains ou artificiels – et leur comportement). Ces modèles doivent permettre de concevoir, spécifier, valider et contrôler de tels agents coopératifs et communicationnels.

Le but de ces journées est de rassembler des chercheurs de différentes communautés scientifiques (informatique, économie, psychologie cognitive, linguistique, sociologie, etc.) ayant en commun la volonté de formaliser tel ou tel aspect de l'interaction entre agents artificiels ou/et humains.

Pour ces raisons, au vu d'une analyse de l'importance croissante du thème de l'interaction dans les systèmes informatiques ouverts qui se dessinent aujourd'hui, il nous paraît opportun de rassembler et d'affirmer une communauté spécifique, particulièrement une communauté francophone.

**Avec le support de**

orange™

APAST
ASSOCIATION POUR L'ANIMATION SCIENTIFIQUE DU TRÉGOR
Siège social : APAST, Cosmopolis-ABRET 22560 Pleumeur-Bodou

Communauté d'Agglomération LANNION-TRÉGOR

ISBN 978-2-95-276302-8

# Cinquièmes Journées Francophones Modèles Formels de l'Interaction

# MFI'09

Du 3 au 5 juin 2009
à Lannion

*Présidents du comité de programme*

Nicolas Maudet & Pierre-Yves Schobbens

Site Web des journées : `http ://www.info.fundp.ac.be/mfi09/`

Imprimerie de l'université de Rennes 1
Campus scientifique de Beaulieu
263 av. du Général Leclerc
CS 74205
35042 Rennes cedex
http ://www.univ-rennes1.fr/imprimerie

ISBN 978-2-9527630-2-8

# Préface

MFI'09 constitue déjà la cinquième édition des Journées, qui se caractérisent par un programme rassemblant des chercheurs et des approches issus de différentes communautés, mais travaillant sur la notion commune d'interaction. Après les succès des éditions de Toulouse (2001), Lille (2003), Caen (2005) et Paris (2007), c'est cette année Lannion qui accueille l'événement.

L'interactivité est une tendance majeure des systèmes informatiques actuels et un important champ de recherche. Elle se décline sous plusieurs aspects :
- interaction entre utilisateurs et systèmes informatiques,
- interaction entre entités informatiques autonomes (agents) interconnectées sur un réseau local ou sur la « toile », en vue de coopérer, de concourir ou tout simplement de coexister,
- sans oublier l'intégration de ces deux aspects dans les divers agents conversationnels, agents de recherche, assistants personnels, ...

Ces tendances sont à l'origine d'un besoin croissant de modèles formels de l'interaction, intégrant les règles, normes et protocoles divers, ainsi que les connaissances spécifiques des agents (en particulier sur les autres agents – humains ou artificiels – et leur comportement). Ces modèles doivent permettre de concevoir, spécifier, valider et contrôler de tels agents coopératifs et communicationnels.

Le but de ces journées est de rassembler des chercheurs de différentes communautés scientifiques (informatique, économie, psychologie cognitive, linguistique, sociologie, etc.) ayant en commun la volonté de formaliser tel ou tel aspect de l'interaction entre agents artificiels ou/et humains.

Cette année, le programme de MFI est composé de trois exposés invités dans l'esprit de cette conférence et de 26 présentations (20 longues et 6 courtes), sélectionnées parmi 32 soumissions. Chaque article a été évalué par trois relecteurs. Les communications acceptées proviennent en grande majorité de laboratoires français, mais on trouve aussi des laboratoires britanniques, canadiens, italiens, luxembourgeois et vénézuéliens. Les affiliations des auteurs reflètent l'aspect pluridisciplinaire des journées : on y trouve des chercheurs en informatique (en particulier en intelligence artificielle, en systèmes multi-agents et en interaction homme-machine), en économie mathématique, en psychologie cognitive et en linguistique.

Les trois exposés invités que nous avons l'honneur d'accueillir cette année sont :
- Paul Goldberg, professeur au Département d'Informatique, Université de Liverpool, sur le thème *The Complexity of Computing a Nash Equilibrium*,

- Philippe Palanque, professeur à l'IHCS-IRIT, Université Paul Sabatier, Toulouse 3 sur le thème *Défis pour les approches formelles pour l'ingénierie des systèmes interactifs*,
- Leon van der Torre, professeur au département d'Informatique et Communications, Faculté des Sciences, Technologie et Communication, Université de Luxembourg, sur le thème *Changing conditional norms*.

Nous tenons à chaleureusement remercier les personnes qui ont contribué au succès de MFI'09 : le comité de programme ainsi que les relecteurs supplémentaires et le comité d'organisation. Par ailleurs, si, conformément à l'esprit de cette conférence, les frais d'inscription de cette édition ont à nouveau pu être maintenus à des montants très abordables, c'est grâce au soutien financier de nombreux acteurs locaux et régionaux que nous remercions sincèrement.

Enfin, il nous semble impossible de conclure cette préface sans évoquer le contexte général dans lequel s'est effectuée la préparation de ces Journées. Cette année 2008-2009 aura en effet été marquée par des mouvements d'une ampleur historique dans les universités et les organismes de recherche français, en réponse à des réformes largement contestées dans la communauté. Nous espérons que MFI'09 aura constitué un îlot de sérénité dans cette année trouble, et vous donnons rendez-vous en 2011 pour de nouvelles aventures.

Nicolas Maudet & Pierre-Yves Schobbens
Présidents du comité de programme

Marc Guyomard
Président du comité d'organisation

# Comité de programme

**Présidents**

| | | |
|---|---|---|
| Nicolas Maudet | LAMSADE, Université Paris-Dauphine | Paris (France) |
| Pierre-Yves Schobbens | Facultés Universitaires (FUNDP) | Namur (Belgique) |

**Membres**

| | | |
|---|---|---|
| Leila Amgoud | Paul Egré | Vincent Louis |
| Nicholas Asher | Amal El Fallah Seghrouchni | Pierre Marquis |
| Philippe Balbiani | Christophe Garion | Philippe Mathieu |
| Martine Batt | Marc Guyomard | Abdel-Illah Mouaddib |
| Bruno Beaufils | Andreas Herzig | Philippe Muller |
| Jean-François Bonnefon | Nicolas Houy | Odile Papini |
| Elise Bonzon | Marc-Philippe Huget | Patrice Perny |
| Jean Caelen | Sébastien Konieczny | Gabriella Pigozzi |
| Brahim Chaib-draa | Frederic Koriche | Nicolas Sabouret |
| François Charpillet | Philippe Lamarre | David Sadek |
| Philippe Chatalic | Jérôme Lang | Hans van Ditmarsch |
| Yann Chevaleyre | Yves Lesperance | Jean-Christophe Vergnaud |
| Jean-Louis Dessalles | Dominique Longin | |

**Relecteurs additionnels**

| | |
|---|---|
| Srdjan Vesic | Maroua Bouzid |

# Comité d'organisation

**Président**

| | | |
|---|---|---|
| Marc Guyomard | ENSSAT/IRISA, Université de Rennes 1 | Lannion (France) |

**Membres**

| | | |
|---|---|---|
| Edith Blin-Guyot | INRIA Rennes, Service REV | Rennes (France) |
| Andreas Classen | Aspirant FNRS, Facultés Univ. de Namur | Namur (Belgique) |
| Michel Guillou | ENSSAT Lannion, Université de Rennes 1 | Lannion (France) |
| Joëlle Thépault | ENSSAT Lannion, Université de Rennes 1 | Lannion (France) |
| Nelly Vaucelle | ENSSAT Lannion, Université de Rennes 1 | Lannion (France) |

# Subventions

MFI'09 a été subventionné par les organismes suivants :
- l'APAST (Association pour l'Animation Scientifique du Trégor)
- le Centre de Recherche INRIA Rennes – Bretagne Atlantique
- la Communauté Lannion Trégor Agglomération
- le Conseil Général des Côtes d'Armor
- l'ENSSAT (Université de Rennes 1), Lannion
- Orange
- l'Université de Rennes 1

# Sommaire/Contents

**Préface** i

## MFI'09 – Actes des cinquièmes Journées Francophones

### Articles

**L. Amgoud, F. Dupin de Saint-Cyr**
*Extraire le noyau d'un dialogue de persuasion pour évaluer sa qualité* 3

**L. Amgoud, S. Vesic**
*An axiomatic account of preference-based argumentation systems* 15

**P. Balbiani, F. Cheikh, G. Feuillade**
*Résultats de complexité pour le problème de la composition d'agents* 25

**P. Balbiani, H. van Ditmarsch, P. Seban**
*Reasoning about permitted announcements* 35

**S. Benferhat, S. Lagrue, J. Rossit**
*Analyse du comportement d'opérateurs de fusion basés sur la somme : du cadre commensurable au cadre incommensurable* 47

**E. Bonzon, C. Devred, M-C. Lagasquie-Schiex**
*Argumentation et CP-jeux booléens* 57

**F. Bouchet, J.-P. Sansonnet**
*Apports complémentaires de la subjectivité et des biais cognitifs à la rationalité dans le contexte de la fonction d'assistance* 69

**J.-R. Bourguet, L. Amgoud, R. Thomopoulos**
*Contribution aux comparaisons formelles des modèles de préférence en argumentation* 81

**A. Burkov, B. Chaib-draa**
*Multiagent Learning and Optimality Criteria in Repeated Game Self-play* 93

**E. Chanoni, T. Lecroq, A. Pauchet**
*Extraction de motifs dans des dialogues annotés par programmation dynamique* 101

**R. Demolombe**
*La confiance dans les relations entre agents : essai de modélisation en Logique Modale* 113

**A. Dimulescu, J.-L. Dessalles**
*Prédire l'intérêt dans la communication événementielle* 125

**P. Everaere, S. Konieczny, P. Marquis**
*Un point de vue épistémique sur la fusion de croyances : l'identification du monde réel* 135

**C. Garion, S. Roussel, L. Cholvy**
*Une logique modale pour raisonner sur la cohérence et la complétude de réglementations* 147

**S. Hoet, N. Sabouret**
*Utiliser des protocoles d'interaction et de la mémoire pour l'apprentissage par renforcement d'actes de communication* 159

**N. Houy**
*Une axiomatisation des choix prudents* 171

**G. Jeantet, O. Spanjaard**
*Choix résolu et utilité espérée dépendant du rang dans les diagrammes d'influence* 181

**S. Konieczny, R. Pino Pérez**
*Opérateurs de confluence* 193

**J. Lang, J. Mengin**
*Sur la complexité de l'apprentissage de préférences séparables de type ceteris paribus* 203

**S. Leblois, J.-F. Bonnefon**
*La bienveillance et les préférences comme paramètres de l'interprétation des recommandations vagues* 213

**Y. Lespérance, S. Shapiro**
*On Agent Programming Language Support for Rational Communication* 223

**M. Morge, P. Mathieu, P. Mancarella**
*Assumption-based argumentation for the minimal concession strategy of agents engaged in resource negotiation* 233

**A. Nongaillard, P. Mathieu**
*Négociation de ressources et maximisation de bien-être social* 243

**W. Ouerdane**
*Représentation des schémas d'arguments dans un contexte d'aide à la décision multicritère* 255

**G. Pigozzi, M. Slavkovik, L. van der Torre**
*Formal ex-post rationalization – A complete conclusion-based procedure for judgment aggregation* 267

**S. Roussel, L. Cholvy**
*Une définition en logique modale de la pertinence orientée agent* 279

**Conférences invitées**

**P. Goldberg**
*The Complexity of Computing a Nash Equilibrium* 291

**P. Palanque**
*Défis pour les approches formelles pour l'ingénierie des systèmes interactifs* 293

**L. van der Torre**
*Changing conditional norms* 295

# ARTICLES

# Extraire le noyau d'un dialogue de persuasion pour évaluer sa qualité

Leila Amgoud* Florence Dupin de Saint-Cyr*
amgoud@irit.fr bannay@irit.fr

* IRIT-CNRS, 118, route de Narbonne,
31062 Toulouse Cedex 9 France

**Résumé :**
Un dialogue de persuasion est un dialogue dans lequel les agents échangent des arguments à propos d'un sujet. Dans ce type de dialogue, les agents ne sont pas d'accord sur le statut du sujet et chacun essaye de persuader les autres de changer d'avis. Dans la littérature, plusieurs systèmes fondés sur la théorie de l'argumentation ont été proposés pour modéliser les dialogues de persuasion. Il est important de pouvoir analyser la qualité de ces dialogues. C'est pourquoi des *critères de qualité* doivent être définis afin d'être à même de réaliser cette analyse.
Cet article aborde cet important problème et propose un critère qui concerne la *concision* du dialogue. Un dialogue est concis si tous ses coups sont appropriés et utiles pour obtenir le même résultat. À partir d'un dialogue de persuasion, on calcule le dialogue "idéal" lui correspondant. Ce dialogue idéal est concis. Un dialogue de persuasion est de bonne qualité s'il est proche de son dialogue idéal.

**Mots-clés :** Argumentation, persuasion, mesures de qualité

**Abstract:**
A persuasion dialog is a dialog in which agents exchange arguments on a subject. In this kind of dialog, the agents disagree about the status of the subject and each one tries to persuade the others to change their mind. Several systems, grounded on argumentation theory, have been proposed in the literature for modeling persuasion dialogs. It is important to be able to analyze the quality of these dialogs. Hence, *quality criteria* have to be defined in order to perform this analysis.
This paper tackles this important problem and proposes one criterion that concerns the *conciseness* of a dialog. A dialog is concise if all its moves are relevant and useful in order to reach the same outcome as the original dialog. From a given persuasion dialog, we compute its corresponding "ideal" dialog. This ideal dialog is concise. A persuasion dialog is of good quality if it is close to its ideal dialog.

**Keywords:** Argumentation, persuasion, quality measures

## 1 Introduction

La persuasion est un des principaux types de dialogue utilisé dans la vie quotidienne. Un dialogue de persuasion concerne des agents qui sont en désaccord sur une question, et chacun d'eux essaie de persuader les autres de changer d'avis. Dans ce but, les agents échangent des arguments de différentes forces. Dans la littérature, plusieurs systèmes ont été proposés pour permettre à des agents de s'engager dans des dialogues de persuasion [4, 5, 7, 8, 10, 11, 13]. Un système de dialogue se base sur trois composants i) un *langage de communication* spécifiant les échanges que les agents auront le droit de faire au cours du dialogue pour faire passer des informations, des arguments etc. ii) un *protocole* spécifiant l'ensemble de règles régissant la bonne marche des dialogues, précisant par exemple qui est autorisé à parler et quand ? et iii) les stratégies des agents ; ce sont les différentes tactiques employées par les agents pour choisir ce qu'ils vont dire à chaque étape du dialogue. Les systèmes actuels permettent à des agents de s'engager dans des dialogues qui respectent les règles d'un protocole. Ainsi, les seules propriétés garanties pour un dialogue issu de tels systèmes sont celles liées au protocole. Par exemple, on peut prouver qu'un dialogue se termine, que les tours de paroles alternent équitablement entre les agents (si une telle règle est spécifiée dans le protocole), que les agents peuvent se référer seulement au mouvement précédent ou au contraire qu'ils ont le droit de répondre à un coup ayant eu lieu plus tôt dans le dialogue, etc. Les propriétés héritées d'un protocole sont liées à la manière dont le dialogue est produit. Cependant, le protocole n'est pas concerné par la *qualité* de ce dialogue. De plus, sous un même protocole, il est possible de produire différents dialogues sur le même sujet. Il est important de pouvoir comparer leur qualité. Une telle comparaison peut aider à raffiner les protocoles et à les rendre plus efficaces.

Bien qu'il y ait beaucoup de travaux sur les protocoles de dialogue (par exemple [9]), aucun travail n'a consisté à définir des critères pour évaluer les dialogues de persuasion produits sous ces protocoles, excepté une proposition très préliminaire dans [2]. Cet article propose trois types de critères pour analyser un dialogue (fini) de persuasion. Le premier type concerne la qualité des arguments échangés dans ce dialogue. Le deuxième s'intéresse au comportement des agents impliqués dans ce dialogue. Le troisième concerne le dialogue dans son ensemble. Dans le présent article, nous nous attachons à étudier plus précisément ce troisième type de critère de qualité. Nous proposons un critère basé sur la

concision du dialogue généré. Un dialogue est concis si tous les coups (c.-à-d. les arguments échangés) sont *en rapport avec* le sujet (c.-à-d. qu'ils ne dévient pas du sujet du dialogue) et *utiles* (c.-à-d. importants pour déterminer le résultat du dialogue). Inspiré des travaux sur les procédures de preuve en théorie d'argumentation permettant de vérifier si un argument est accepté ou pas [1], nous calculons et caractérisons un sous-dialogue concis extrait du dialogue original. Ce sous-dialogue est dit *idéal*. Plus le dialogue original est proche de son sous-dialogue idéal, meilleure est sa qualité. Cet article est une version française de l'article [3] (paru dans les actes de la conférence ECSQARU'09) auquel on a adjoint toutes les démonstrations (non incluses dans la version anglaise).

Le papier est organisé comme suit : la section 2 contient un bref exposé de notions de bases de la théorie d'argumentation. La section 3 présente le cadre des dialogues de persuasion. La section 4 définit les notions de coup approprié et de coup utile dans un dialogue. La section 5 présente le concept de dialogue idéal fondé sur un arbre idéal d'arguments construit à partir du dialogue initial.

## 2 Notions de base en argumentation

L'argumentation est un modèle de raisonnement basé sur la construction et la comparaison des arguments. Les arguments sont des raisons de croire certaines affirmations, ou d'effectuer certaines actions. Ici, les origines des arguments sont supposées inconnues. Ils sont symbolisés par des lettres grecques minuscules. Dans [6], un système d'argumentation est défini par :

**Définition 1 (Système d'argumentation)** *Un* système d'argumentation *est une paire* $\mathsf{AS} = \langle \mathcal{A}, \mathcal{R} \rangle$*, où* $\mathcal{A}$ *est un ensemble d'arguments et* $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ *est une relation d'attaque. On dit qu'un argument* $\alpha$ *attaque un argument* $\beta$ *ssi* $(\alpha, \beta) \in \mathcal{R}$.

Notons qu'à chaque système d'argumentation est associé un graphe orienté dont les noeuds sont les différents arguments, et les arcs représentent la relation d'attaque entre eux. Puisque les arguments sont en conflits, il est important de savoir quels arguments sont acceptables. Dans ce but, dans [6], différentes *sémantiques d'acceptabilité* ont été proposées. Ici, nous ne considérons que le cas de la sémantique *basique*. Les sémantiques restantes pourront être étudiées ultérieurement.

**Définition 2 (Défense–Extension basique)**
*Soit* $\mathsf{AS} = \langle \mathcal{A}, \mathcal{R} \rangle$ *et* $\mathcal{B} \subseteq \mathcal{A}$.
- $\mathcal{B}$ défend *un argument* $\alpha \in \mathcal{A}$ *ssi* $\forall\ \beta \in \mathcal{A}$*, si* $(\beta, \alpha) \in \mathcal{R}$*, alors* $\exists \delta \in \mathcal{B}$ *t.q.* $(\delta, \beta) \in \mathcal{R}$.
- *L'*extension basique *de* $\mathsf{AS}$*, notée* $\mathcal{E}$*, est le plus petit point fixe de la fonction* $\mathcal{F}$ *avec* $\mathcal{F}(\mathcal{B}) = \{\alpha \in \mathcal{A} \mid \mathcal{B}$ *défend* $\alpha\}$.

Si le système d'argumentation est fini (c.à.d. chaque argument n'est attaqué que par un nombre fini d'arguments) alors $\mathcal{E} = \bigcup_{i>0} \mathcal{F}^i(\emptyset)$.

Maintenant que la sémantique d'acceptabilité est définie, nous pouvons définir le statut de n'importe quel argument.

**Définition 3 (Statut d'un argument)** *Soit* $\mathsf{AS} = \langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation et* $\mathcal{E}$ *son extension basique. Un argument* $\alpha \in \mathcal{A}$ *est* accepté *ssi* $\alpha \in \mathcal{E}$*, dans le cas contraire, il est* rejeté.
*On note* $\texttt{Status}(\alpha, \mathsf{AS})$ *le statut de* $\alpha$ *dans* $\mathsf{AS}$.

**Propriété 1 ([1])** *Soient* $\mathsf{AS} = \langle \mathcal{A}, \mathcal{R} \rangle$*,* $\mathcal{E}$ *son extension basique, et* $\alpha \in \mathcal{A}$*. Si* $\alpha \in \mathcal{E}$*, alors* $\alpha$ *est* indirectement défendu[1] *contre tous ses attaquants par des arguments non attaqués.*

## 3 Dialogues de persuasion

Cette section définit des dialogues de persuasion dans le même esprit que [4]. Un dialogue de persuasion consiste principalement en un échange d'arguments entre différents agents de l'ensemble $\texttt{AG} = \{a_1, \ldots, a_m\}$. Le sujet d'un tel dialogue est un argument, et le but du dialogue est de fournir le statut de cet argument. À la fin du dialogue, l'argument peut être "accepté" ou "rejeté", ce statut est le résultat du dialogue. Dans ce qui suit, nous supposons que les agents ne peuvent échanger *que* des arguments.

Chaque agent participant est censé pouvoir identifier tous les éléments de $\arg(\mathcal{L})$ et de $\mathcal{R}_{\mathcal{L}}$, où $\arg(\mathcal{L})$ est l'ensemble de tous les arguments qui peuvent être construits à partir d'un langage

[1] Un argument $\alpha$ est *indirectement défendu* par $\beta$ ssi il existe une séquence finie d'arguments $a_1, \ldots, a_{2n+1}$ telle que $\alpha = a_1$, $\beta = a_{2n+1}$, et $\forall i \in [\![1, 2n]\!]$, $(a_{i+1}, a_i) \in \mathcal{R}$, $n \in \mathbb{N}^*$.

logique $\mathcal{L}$ et $\mathcal{R}_\mathcal{L}$ est une relation binaire qui capture tous les conflits qui peuvent exister parmi des arguments de $\arg(\mathcal{L})$. Ainsi, $\mathcal{R}_\mathcal{L} \subseteq \arg(\mathcal{L}) \times \arg(\mathcal{L})$. Pour deux arguments $\alpha, \beta \in \arg(\mathcal{L})$, la paire $(\alpha, \beta) \in \mathcal{R}_\mathcal{L}$ signifie que l'argument $\alpha$ attaque l'argument $\beta$. Notons que cela ne signifie pas que les agents ont connaissance de tous les arguments. Mais, cela signifie que les agents emploient le même langage logique et les mêmes définitions pour les notions d'arguments et pour la relation de conflit.

**Définition 4 (Coup)** *Un* coup $m$ *est un triplet* $\langle S, H, \alpha \rangle$ *tel que :*
- $S \in \mathtt{AG}$ *est l'agent qui émet le coup,* $\mathtt{Speaker}(m) = S$
- $H \subseteq \mathtt{AG}$ *est l'ensemble d'agents auxquels le coup est adressé,* $\mathtt{Hearer}(m) = H$
- $\alpha \in \arg(\mathcal{L})$ *est le contenu du coup,* $\mathtt{Content}(m) = \alpha$.

Pendant un dialogue plusieurs coups peuvent être émis. Ces coups constituent une séquence notée $\langle m_1, \ldots, m_n \rangle$, où $m_1$ est le coup initial et $m_n$ le dernier coup. La séquence vide est notée $\langle\rangle$. Ces séquences sont construites sous un protocole donné. Un protocole est une fonction qui associe à chaque séquence de coups un ensemble de coups valides. Plusieurs protocoles ont été proposés dans la littérature, par exemple [4, 11]. Dans ce qui suit, nous ne nous restreignons pas à un protocole précis.

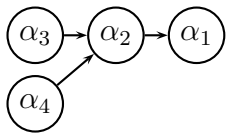
**Définition 5 (Dialogue de persuasion)** *Un* dialogue de persuasion $D$ *est une séquence finie non vide de coups* $\langle m_1, \ldots, m_n \rangle$ *t.q. le* sujet *de* $D$ *est* $\mathtt{Subject}(D) = \mathtt{Content}(m_1)$*, et la* longueur *de* $D$*, notée* $|D|$*, est le nombre de coups :* $n$*. Chaque sous-séquence* $\langle m_1, \ldots, m_i \rangle$ *est un* sous-dialogue $D^i$ *de* $D$*. Noté* $D^i \sqsubset D$.

À chaque dialogue de persuasion, on peut associer un système d'argumentation afin d'éva-luer le statut de chaque argument émis et afin de calculer le résultat du dialogue.

**Définition 6 (Système d'arg. d'un dialogue)** *Soit* $D = \langle m_1, \ldots, m_n \rangle$ *un dialogue de persuasion. Le* système d'argumentation *de* $D$ *est la paire* $\mathsf{AS}_D = \langle \mathtt{Args}(D), \mathtt{Confs}(D) \rangle$ *telle que :*
- $\mathtt{Args}(D) = \{\mathtt{Content}(m_i) \mid i \in [\![1, n]\!]\}$
- $\mathtt{Confs}(D) = \{(\alpha, \beta) \mid \alpha, \beta \in \mathtt{Args}(D)$ *et* $(\alpha, \beta) \in \mathcal{R}_\mathcal{L}\}$

En d'autres termes, $\mathtt{Args}(D)$ et $\mathtt{Confs}(D)$ renvoient respectivement, l'ensemble des arguments échangés pendant le dialogue et les différents conflits parmi ces arguments.

**Exemple 1** *Soit* $D_1$ *le dialogue de persuasion suivant entre les agents* $a_1$ *et* $a_2$*.* $D_1 = \langle\langle a_1, \{a_2\}, \alpha_1\rangle, \langle a_2, \{a_1\}, \alpha_2\rangle, \langle a_1, \{a_2\}, \alpha_3\rangle, \langle a_1, \{a_2\}, \alpha_4\rangle, \langle a_2, \{a_1\}, \alpha_1\rangle\rangle$*. Supposons qu'il existe des conflits dans* $\mathcal{R}_\mathcal{L}$ *parmi certains de ces arguments. Ces conflits sont résumés ainsi :*




*Ici,* $\mathtt{Args}(D_1) = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ *et* $\mathtt{Confs}(D_1) = \{(\alpha_2, \alpha_1), (\alpha_3, \alpha_2), (\alpha_4, \alpha_2)\}$.

**Propriété 2** *Soit* $D = \langle m_1, \ldots, m_n \rangle$ *un dialogue de persuasion.* $\forall D^j \sqsubset D$, $\mathtt{Args}(D^j) \subseteq \mathtt{Args}(D)$ *et* $\mathtt{Confs}(D^j) \subseteq \mathtt{Confs}(D)$.

Le résultat du dialogue de persuasion est le statut de l'argument à l'origine de la discussion (le sujet) :

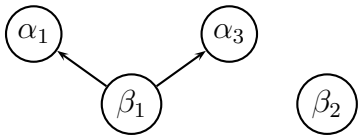
**Définition 7 (Résultat du dialogue)** *Soit* $D$ *un dialogue de persuasion. Le* résultat *de* $D$*, noté* $\mathtt{Output}(D)$*, est* $\mathtt{Status}(\mathtt{Subject}(D), \mathsf{AS}_D)$.

## 4 Critères de qualité d'un dialogue

Ici, nous nous intéressons à évaluer la concision d'un dialogue $D$ qui a déjà été produit sous un protocole donné. Ce dialogue est supposé *fini*. Notons que cette hypothèse n'est pas trop forte puisqu'une des principales propriétés d'un protocole est l'arrêt des dialogues qu'il permet de générer [12]. Une conséquence de cette hypothèse est que le système $\mathsf{AS}_D$ d'argumentation associé à $D$ est également fini. Dans ce qui suit, nous proposons deux critères pour évaluer l'importance des coups échangés dans $D$.

**Définition 8 (Coups appropriés et utiles)** *Soit* $D = \langle m_1, \ldots, m_n \rangle$ *un dialogue de persuasion. Un coup* $m_i$*,* $i \in [\![1, n]\!]$*, est* approprié *pour* $D$ *ssi il existe une chaîne entre* $\mathtt{Content}(m_i)$ *et* $\mathtt{Subject}(D)$ *dans le graphe orienté associé à* $\mathsf{AS}_D$*.* $m_i$ *est* utile *ssi il existe un chemin de* $\mathtt{Content}(m_i)$ *à* $\mathtt{Subject}(D)$ *dans ce graphe.*

**Exemple 2** *Soit $D_2$ un dialogue de persuasion. Soit* $\texttt{Args}(D_2) = \{\alpha_1, \alpha_3, \beta_1, \beta_2\}$. *Les conflits entre les 4 arguments sont décrits ci-dessous :*




*Supposons que* $\texttt{Subject}(D_2) = \alpha_1$. *Il est clair que les arguments $\alpha_3$ et $\beta_1$ sont appropriés, tandis que $\beta_2$ ne l'est pas. $\beta_1$ est utile mais pas $\alpha_3$.*

**Propriété 3** *Un coup utile dans un dialogue $D$ est approprié pour $D$.*

On peut définir une mesure de pertinence sur la base de la notion de coup approprié en calculant le pourcentage de coups qui sont appropriés dans le dialogue. Dans l'exemple 2, $\texttt{Pertinence}(D_2)$ = 3/4. Il est clair que plus ce degré est grand, meilleur est le dialogue. Quand le degré de pertinence d'un dialogue est égal à 1, cela signifie que les agents n'ont pas dévié du sujet du dialogue. Les coups utiles sont ceux qui ont une influence potentielle sur le statut du sujet. Mais cela ne signifie pas que leur présence a forcément un impact sur le résultat du dialogue, c.-à-d., sur le statut du sujet. Les coups qui ont un impact réel sur le statut du sujet sont dits "décisifs".

**Définition 9 (Coup décisif)** *Soit $D = \langle m_1, \ldots, m_n \rangle$ un dialogue de persuasion et $\mathsf{AS}_D$ son système d'argumentation. Un coup $m_i$ ($i = 1, \ldots, n$) est* décisif *dans $D$ ssi* $\texttt{Status}(\texttt{Subject}(D), \mathsf{AS}_D) \neq \texttt{Status}(\texttt{Subject}(D), \mathsf{AS}_D \ominus \texttt{Content}(m_i))$

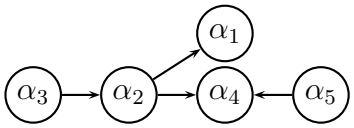
*avec* $\mathsf{AS}_D \ominus \texttt{Content}(m_i) = \langle A', R' \rangle$ *t.q.* $A' = \texttt{Args}(D) \setminus \{\texttt{Content}(m_i)\}$ *et* $R' = \texttt{Confs}(D) \setminus \{(x, \texttt{Content}(m_i)), (\texttt{Content}(m_i), x) \mid x \in \texttt{Args}(D)\}$.

On peut vérifier que si un coup est décisif dans un dialogue, alors il est utile. Ceci signifie qu'il existe un chemin du contenu de ce coup vers le sujet du dialogue dans le graphe associé au dialogue.

**Propriété 4** *Un coup décisif dans un dialogue de persuasion $D$ est utile dans $D$.*

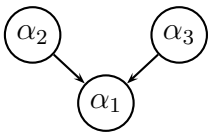
De la propriété ci-dessus, il suit que chaque coup décisif est également approprié. Notons que l'inverse n'est pas vrai comme le montre l'exemple suivant.

**Exemple 3** *Soit $D_3$ un dialogue de sujet $\alpha_1$, dont le graphe est le suivant :*




*L'ensemble $\{\alpha_1, \alpha_3, \alpha_5\}$ est l'unique extension basique de $\mathsf{AS}_{D_3}$. Il est clair que l'argument $\alpha_4$ est approprié par rapport à $\alpha_1$, mais il n'est pas décisif pour $D_3$. En effet, la suppression de $\alpha_4$ ne change pas le statut de $\alpha_1$ qui est accepté.*

**Exemple 4** *Soit $D_4$ un dialogue de sujet $\alpha_1$, admettant le graphe suivant :*




*Dans cet exemple, ni $\alpha_2$ ni $\alpha_3$ ne sont décisifs dans $D_4$. Cependant, cela ne signifie pas que les deux arguments devraient être supprimés puisque le statut de $\alpha_1$ dépend au moins de l'un d'entre eux (ils sont tous les deux utiles).*

Sur la base de la notion de coup décisif, on peut définir un degré pour le dialogue entier par le pourcentage des coups décisifs dans le dialogue.

## 5 Calcul du dialogue idéal

Il arrive que les dialogues courants contiennent des redondances dans le sens où les coups émis sont inutiles par rapport au sujet, ou n'ont aucun impact sur le résultat du dialogue. Dans de tels dialogue, seulement un sous-ensemble des arguments est nécessaire pour déterminer le statut du sujet. Notre but est de calculer le sous-ensemble qui renvoie exactement le même statut pour le sujet du dialogue que si l'on avait conservé la totalité des arguments, mais suffisant pour convaincre que ce résultat résiste à n'importe quelle attaque disponible dans le dialogue initial. Ce sous-ensemble formera le dialogue "idéal". Dans ce qui suit, nous fournissons

une procédure pour trouver ce sous-ensemble et ainsi le dialogue idéal.

Un sous-ensemble d'arguments adéquat contient les arguments d'un arbre de preuve pour le statut du sujet. Ceci est dû au fait qu'un arbre de preuve contient les arguments nécessaires pour obtenir le statut du sujet. Quand le sujet est accepté, l'arbre de preuve contient des défenseurs du sujet contre n'importe quelle attaque. Quand le sujet est rejeté, l'arbre de preuve contient au moins tout attaquant non attaqué. Par conséquent, les arbres de preuve semblent adaptés pour résumer parfaitement un dialogue. Cependant, il est important de souligner que l'on ne peut pas choisir n'importe quelle théorie de preuve existant dans la littérature. Ceci vient du fait que certaines théories de preuve ne sont pas concises. Dans [1], une comparaison des théories de preuve pour la sémantique basique prouve que celle que nous allons utiliser est la plus concise.

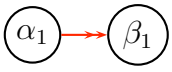
### 5.1 Dialogues Canoniques

Nous allons définir un sous-dialogue d'un dialogue de persuasion donné $D$ qui atteint le même résultat que $D$. Dans [1], une procédure de preuve est proposée pour examiner l'appartenance d'un argument à une extension basique. Les notions de base de cette procédure sont revisitées et adaptées afin de caractériser des dialogues canoniques.

**Définition 10 (Branche de dialogue)** *Soit $D$ un dialogue de persuasion et $\mathsf{AS}_D = \langle \mathtt{Args}(D), \mathtt{Confs}(D)\rangle$ son système d'argumentation. Une* branche de dialogue *pour $D$ est une séquence $\langle \alpha_0, \ldots, \alpha_p\rangle$ d'arguments t.q. $\forall i, j \in [\![0, p]\!]$*

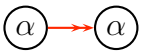
1. $\alpha_i \in \mathtt{Args}(D)$
2. $\alpha_0 = \mathtt{Subject}(D)$
3. *si* $i \neq 0$ *alors* $(\alpha_i, \alpha_{i-1}) \in \mathtt{Confs}(D)$
4. *si* $i$ *et* $j$ *sont pairs et* $i \neq j$ *alors* $\alpha_i \neq \alpha_j$
5. *si* $i$ *est pair et* $i \neq 0$ *alors* $(\alpha_{i-1}, \alpha_i) \notin \mathtt{Confs}(D)$
6. $\forall \beta \in \mathtt{Args}(D)$, $\langle \alpha_0, \ldots, \alpha_p, \beta\rangle$ *n'est pas une branche de dialogue pour $D$.*

**Exemple 5** *L'unique branche de dialogue qui peut être construite à partir du dialogue $D_2$ est décrite ci-dessous :*




**Exemple 6** *Soit $D_5$ un dialogue de persuasion ayant pour sujet $\alpha$ et pour graphe :*

*L'unique branche de dialogue qui peut être associée à ce dialogue est :*




**Propriété 5** *Une branche de dialogue est non vide et finie.*

Ce résultat vient des définitions d'une branche de dialogue et d'un dialogue de persuasion.

**Propriété 6** *Pour chaque branche $\langle \alpha_0, ..., \alpha_k\rangle$ d'un dialogue $D$ de persuasion il existe un chemin unique $\langle \alpha_k, \alpha_{k-1}, ..., \alpha_0\rangle$ de même longueur [2] $(k)$ dans le graphe orienté associé à $\mathsf{AS}_D$.*

Quand une branche de dialogue est de longueur paire sa feuille n'est pas attaquée dans le dialogue original.

**Théorème 1** *Soit $D$ un dialogue de persuasion et $\langle \alpha_0, \ldots \alpha_p\rangle$ une branche de dialogue donnée de $D$. Si $p$ est pair, alors $\nexists \beta \in \mathtt{Args}(D)$ tel que $(\beta, \alpha_p) \in \mathtt{Confs}(D)$.*

Nous présentons maintenant la notion d'arbre de dialogue.

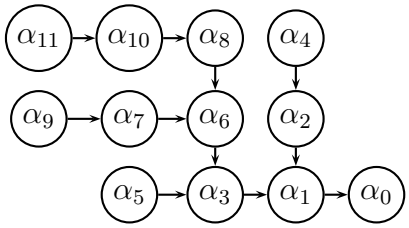
**Définition 11 (Arbre de dialogue)** *Soit $D$ un dialogue de persuasion et $\mathsf{AS}_D = \langle \mathtt{Args}(D), \mathtt{Confs}(D)\rangle$ son système d'argumentation. Un* arbre de dialogue *de $D$, noté $D^t$, est un arbre fini dont les branches sont toutes les branches de dialogues possibles qui peuvent être construites à partir de $D$.*

*On note $\mathsf{AS}_{D^t}$ le système d'argumentation associé à $D^t$, $\mathsf{AS}_{D^t} = \langle A^t, C^t\rangle$ t.q. $A^t = \{\alpha \in \mathtt{Args}(D)$ t.q. $\alpha$ apparaît en tant que noeud dans $D^t\}$ et $C^t = \{(\alpha, \beta) \in \mathtt{Confs}(D)$ t.q. $(\beta, \alpha)$ est un arc de $D^t\}$.*

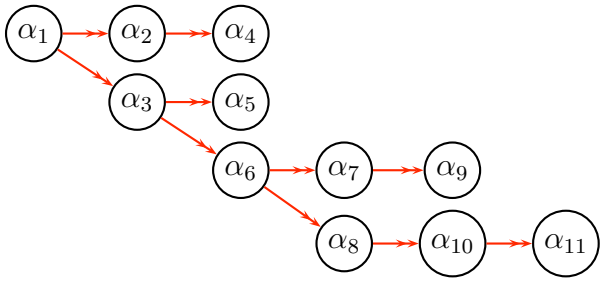
Par conséquent, un arbre de dialogue est un arbre dont la racine est le sujet du dialogue de persuasion.

[2] la longueur d'un chemin est défini par son nombre d'arcs.

**Exemple 7** *Considérons $D_6$ dont le sujet est $\alpha_1$ et dont le graphe est le suivant :*




*L'arbre de dialogue associé à ce dialogue est décrit ci-dessous :*




*Notons que l'argument $\alpha_0$ n'appartient pas à l'arbre de dialogue.*

**Propriété 7** *À chaque dialogue de persuasion correspond exactement un arbre de dialogue.*

Un résultat important établit que le statut du sujet du dialogue original $D$ de persuasion est exactement le même dans les systèmes $\mathsf{AS}_D$ d'argumentation et $\mathsf{AS}_{D^t}$ (où $\mathsf{AS}_{D^t}$ est le système d'argumentation dont les arguments sont tous les arguments qui apparaissent dans l'arbre $D^t$ de dialogue et dont les attaques sont obtenues en inversant les arcs entre ces arguments dans $D^t$).

**Théorème 2** *Soit $D$ un dialogue de persuasion et $\mathsf{AS}_D$ son système d'argumentation. On a :* $\texttt{Status}(\texttt{Subject}(D), \mathsf{AS}_D) = \texttt{Status}(\texttt{Subject}(D), \mathsf{AS}_{D^t})$.

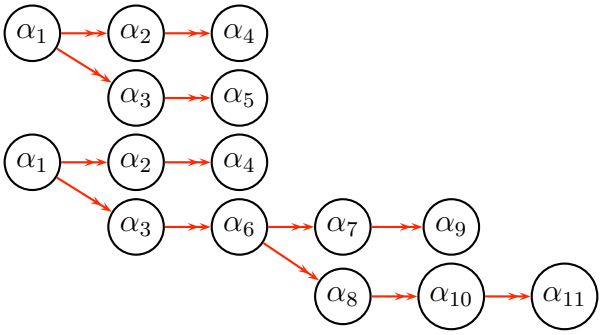
Afin de calculer le statut du sujet d'un dialogue, nous pouvons considérer l'arbre de dialogue en tant qu'arbre et/ou. Un noeud d'un niveau pair est un noeud et, tandis qu'un noeud de niveau impair est noeud ou. Cette distinction entre les noeuds est due au fait qu'un argument est accepté s'il peut être défendu contre tous ses attaquants. Un arbre de dialogue peut être décomposé en un ou plusieurs arbres appelés arbres canoniques. Un *arbre canonique* est un sous-arbre de $D^t$ dont la racine est $\texttt{Subject}(D)$ et qui contient tous les arcs sortant d'un noeud pair et exactement un arc sortant d'un noeud impair.

**Définition 12 (Arbre canonique)** *Soit $D$ un dialogue de persuasion, et soit $D^t$ son arbre de dialogue. $D^c$ est un* arbre canonique *de $D^t$ si c'est un sous-arbre de $D^t$ construit par niveaux comme suit :*

- $\texttt{Subject}(D)$ *est sa racine (de niveau 0)*
- *et inductivement : :*
  - *si $\alpha$ est un noeud de niveau pair dans $D^c$ alors pour* ***tout*** *$\beta \in D^t$ tel que $(\alpha, \beta) \in D^t$, le noeud $\beta$ et l'arc $(\alpha, \beta)$ est ajouté à $D^c$.*
  - *si $\alpha$ est un noeud de niveau impair dans $D^c$ et si $\alpha$ admet au moins un attaquant dans $D^t$ alors pour* ***exactement un*** *$\beta \in D^t$ tel que $(\alpha, \beta) \in D^t$, le noeud $\beta$ et l'arc $(\alpha, \beta)$ est ajouté à $D^c$.*

Notons qu'à partir d'un arbre de dialogue, on peut extraire au moins un arbre canonique. Soient $D^c_1, \ldots, D^c_m$ les arbres canoniques que l'on peut extraire. Nous noterons $\mathsf{AS}^c_1, \ldots \mathsf{AS}^c_m$ les systèmes d'argumentation leur correspondant. On peut vérifier que le statut de $\texttt{Subject}(D)$ n'est pas nécessairement le même dans ces différents systèmes.

**Exemple 8** *À partir de l'arbre de $D_6$, deux arbres canoniques peuvent être extraits :*




*On peut vérifier que l'argument $\alpha_1$ est accepté dans le système d'argumentation du haut tandis qu'il est rejeté dans celui du bas.*

Le résultat suivant caractérise le statut de $\texttt{Subject}(D)$ dans le système $\mathsf{AS}^c_i$ d'argumentation associé à un arbre canonique $D^c_i$.

**Théorème 3** *Soient $D$ un dialogue de persuasion, $D_i^c$ un arbre canonique et $\mathsf{AS}_i^c$ le système d'argumentation correspondant.* $\texttt{Subject}(D)$ *est accepté dans $\mathsf{AS}_i^c$ ssi toutes les branches de $D_i^c$ sont de longueur paire.*

Le résultat suivant découle immédiatement de ce théorème et du théorème 1.

**Corollaire 1** *Soient $D$ un dialogue de persuasion, $D_i^c$ un arbre canonique et $\mathsf{AS}_i^c$ le système d'argumentation correspondant.*
*Si* $\texttt{Subject}(D)$ *est accepté dans $\mathsf{AS}_i^c$, alors toutes les feuilles de $D_i^c$ ne sont pas attaquées dans $D$.*

Un résultat important montre le lien entre les résultats d'un dialogue $D$ et les résultats des différents arbres canoniques.

**Théorème 4** *Soient $D$ un dialogue de persuasion, $D_1^c$, ..., $D_m^c$ ses différents arbres canoniques et $\mathsf{AS}_1^c, \ldots, \mathsf{AS}_m^c$ leur système d'argumentation correspondant.*
$\texttt{Output}(D)$[3] *est accepté ssi* $\exists\ i \in [\![1, m]\!]$ *t.q.* $\texttt{Status}(\texttt{Subject}(D), \mathsf{AS}_i^c)$ *est accepté.*

Ce résultat est de grande importance puisqu'il prouve qu'un arbre canonique dont les branches sont toute de longueur paire est suffisant pour obtenir les mêmes résultats que le dialogue original dans le cas où le sujet est accepté. Quand le sujet est rejeté, l'arbre entier de dialogue est nécessaire pour assurer les résultats.

**Exemple 9** *Dans l'exemple 7, le sujet $\alpha_1$ du dialogue $D_6$ est accepté puisqu'il y a un arbre canonique dont les branches sont de longueur paire (c'est l'arbre canonique du haut dans l'exemple 8). On peut également vérifier que $\alpha_1$ est dans l'extension basique $\{\alpha_1, \alpha_4, \alpha_5, \alpha_8, \alpha_9, \alpha_{11}\}$ de $\mathsf{AS}_D$.*

Jusqu'ici, nous avons montré comment extraire des arbres canoniques à partir d'un graphe associé à un dialogue. Ces arbres canoniques ne contiennent que des coups utiles (donc appropriés) :

**Théorème 5** *Soit $D_i^c$ un arbre canonique d'un dialogue $D$ de persuasion. Tout coup basé sur un argument de $D_i^c$ est* utile *dans le dialogue $D$.*

[3] Rappelons que $\texttt{Output}(D) = \texttt{Status}(\texttt{Subject}(D), \mathsf{AS}_D)$.

Le théorème précédent donne une limite supérieure de l'ensemble de coups qui peuvent être employés pour construire un arbre canonique, une limite inférieure est l'ensemble des coups décisifs.

**Théorème 6** *Tout argument d'un coup décisif appartient à l'arbre de dialogue et à tout arbre canonique.*

La réciproque est fausse puisque beaucoup d'arguments ne sont pas décisifs, comme le montre l'exemple 4. En effet, il y a deux attaquants qui ne sont pas décisifs mais l'arbre de dialogue contient les deux (de même pour le seul dialogue canonique pour cet exemple).

### 5.2 Le dialogue idéal

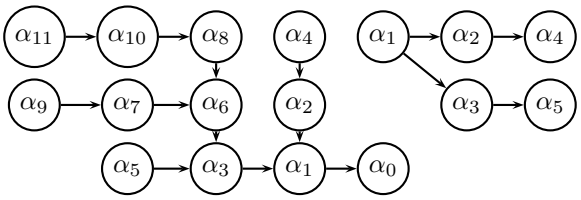
Dans la section précédente, nous avons prouvé que l'on pouvait construire un arbre de dialogue associé à tout dialogue de persuasion. Cet arbre de dialogue contient des attaquants direct et indirects du sujet ainsi que des défenseurs. De cet arbre de dialogue, des sous-arbres intéressants peuvent être extraits et s'appellent les arbres canoniques. Un arbre canonique est un sous-arbre contenant seulement certaines branches entières de l'arbre de dialogue (seulement un argument en faveur du sujet est choisi pour attaquer un attaquant tandis qu'un défenseur est choisi pour attaquer chaque argument contre le sujet). Dans le cas où le sujet du dialogue est accepté, on a montré qu'il existe au moins un arbre canonique tels que le sujet est accepté dans son système d'argumentation. Cet arbre canonique est un candidat pour être un arbre idéal puisqu'il est suffisant pour justifier l'acceptation du sujet contre toute attaque disponible dans le dialogue initial. Parmi tous ces candidats nous définissons l'arbre idéal en choisissant le plus petit. Dans le cas où le sujet est rejeté dans le dialogue initial, alors l'arbre de dialogue contient toutes les raisons de le rejeter, par conséquent nous proposons de considérer l'arbre de dialogue lui-même comme seul arbre idéal.

**Définition 13 (Arbres et dialogues idéaux)**
*Si un dialogue $D$ a pour résultat accepté - alors un* arbre idéal *associé à $D$ est un arbre canonique de $D$ dans lequel* $\texttt{Subject}(D)$ *est accepté et ayant un nombre minimal de noeuds parmi tous les arbres canoniques qui acceptent également* $\texttt{Subject}(D)$

- *sinon l'*arbre idéal *est l'arbre de dialogue de* $D$.

*Un dialogue utilisant une fois chaque argument d'un arbre idéal est appelé* dialogue idéal.

**Exemple 10** *Un dialogue idéal pour le dialogue D6 (du côté gauche) admet l'arbre suivant (du côté droit) :*

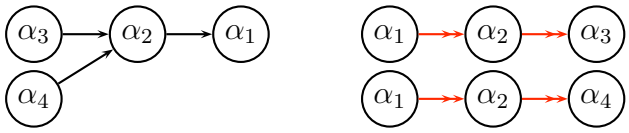



Étant donné la définition ci-dessus, un dialogue idéal contient exactement le même nombre de coups qui le nombre de noeuds du graphe idéal.

**Propriété 8** *Étant donné un dialogue $D$ dont le sujet est accepté. Un dialogue idéal $ID$ pour $D$ est le dialogue le plus court ayant le même résultat, et t.q. chaque argument en faveur du sujet dans $ID$ (y compris* $\texttt{Subject}(D)$ *lui-même) est défendu contre n'importe quelle attaque (existant dans $D$).*

Cette propriété assure que, quand le sujet est accepté dans le dialogue initial $D$, un dialogue idéal $ID$ est le dialogue le plus concis qui induit une acception. En d'autres termes, nous imposons que le dialogue idéal contienne un ensemble d'arguments qui résument $D$.

Notons que le dialogue idéal existe mais n'est pas toujours unique. Voici un exemple de système d'argumentation d'un dialogue qui mène à deux arbres idéaux (par conséquent il mènera au moins à deux dialogues idéaux).




Jusqu'ici, nous avons formellement défini la notion de dialogue idéal, et montré comment il est extrait à partir d'un dialogue de persuasion. Il est clair plus un dialogue est proche (au sens de l'inclusion ensembliste des arguments échangés) de sa version idéale, plus le dialogue est concis.

# 6 Conclusion

Dans cet article, nous avons proposé trois critères pour évaluer les coups d'un dialogue de persuasion par rapport à son sujet : la pertinence, l'utilité et le côté décisif. La pertinence exprime seulement que l'argument du coup a un lien avec le sujet (ce lien est basé sur la relation d'attaque du système d'argumentation). L'utilité est une pertinence plus forte puisqu'elle exige un lien dirigé de l'argument du coup vers le sujet. Les coups décisifs ont un impact plus lourd sur le dialogue, puisque leur omission change le résultat du dialogue.

Inspiré des travaux sur les théories de preuve pour la sémantique basique en argumentation, nous avons défini une notion de "dialogue idéal". Plus précisément, nous avons d'abord défini un arbre de dialogue associé à un dialogue donné comme un graphe qui contient chaque attaquants possibles et défenseurs directs et indirects du sujet. De cet arbre de dialogue, il est alors possible d'extraire des sous-arbres appelé "arbres idéaux" qui sont suffisant pour montrer que le sujet est accepté ou rejeté dans le dialogue original et ceci, contre tout attaque possible d'argument issu du dialogue initial. Un dialogue est bon s'il est proche de cet arbre idéal. Les dialogues idéaux ont des bonnes propriétés par rapport à la concision, à savoir qu'ils contiennent seulement des arguments utiles et appropriés pour le sujet du dialogue. De plus chaque argument d'un coup décisif appartient à tous les arbres idéaux.

À partir de nos résultats, il semble naturel de déduire des conditions pour qu'un protocole produise des dialogues de bonne qualité : (1) Les coups non appropriés et non utiles doivent être pénalisés jusqu'à ce qu'il y ait un ensemble d'arguments qui les rattache au sujet (2) L'ajout d'arguments en faveur du sujet qui sont attaqués par des arguments déjà présent n'a aucun intérêt (puisqu'ils n'appartiennent à aucun arbre idéal). Ainsi, les dialogues produits pourraient être plus *concis* (c.-à-d. que tous les arguments émis ont un impact (au moins potentiel) sur le résultat du dialogue), et plus *efficace* (c.-à-d. que ce sont les dialogues les plus courts qui peuvent être construit à partir de l'information échangée et qui atteignent le but de la persuasion).

Notons que dans notre proposition, l'ordre des arguments n'a pas à être contraint puisque le graphe généré n'en tient pas compte. La seule chose qui importe pour obtenir un résultat est

l'ensemble final d'interactions entre les arguments échangés. Les critères d'être lié au coup précédent ou au moins à un coup pas trop loin dans la séquence de dialogue pourraient être pris en considération pour analyser la qualité du dialogue. D'ailleurs, toutes les mesures déjà définies dans la littérature et citées dans l'introduction pourraient également être employées pour raffiner la relation de préférence sur des dialogues et finalement pourraient aider à formaliser les propriétés générales des protocoles permettant de produire de bons dialogues.

En outre, il peut arriver qu'à partir d'un ensemble de formules impliquées dans un ensemble d'arguments on puisse construire de nouveaux arguments. Ainsi on disposerait d'un nouvel ensemble d'arguments et d'un nouvel ensemble de relations d'attaque (système d'argumentation complet) associé à un dialogue. Dans ce cas, il pourrait être intéressant de définir les arbres de dialogue sur la base de ce système complet d'argumentation. Alors des dialogues plus efficaces pourraient être obtenus (mais ce n'est pas garanti). Cependant, certains arguments du système complet d'argumentation pourraient nécessiter la coopération des agents. Cela signifierait que dans un dialogue idéal mais réalisable, l'ordre d'émission des arguments serait contraint par le fait que chaque agent devrait pouvoir établir chaque argument à chaque étape.

## Références


[1] L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34 :197 – 216, 2002.

[2] L. Amgoud and F. Dupin de Saint-Cyr. Measures for persuasion dialogs : a preliminary investigation. In *2$^{nd}$ Int. Conf. on Computational Models of Argument*, pages 13–24, 2008.

[3] L. Amgoud and F. Dupin de Saint-Cyr. Extracting the core of a persuasion dialog to evaluate its quality. In *10$^{th}$ European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2009)*, Verona, Italy, July 1-3 2009.

[4] L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In *Proc. of the International Conference on Multi-Agent Systems*, pages 31–38, Boston, MA, 2000.

[5] T. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *J. of Logic and Computation*, 13(3) :429–448, 2003.

[6] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and $n$-person games. *Artificial Intelligence Journal*, 77 :321–357, 1995.

[7] P.E. Dunne and T.J. Bench-Capon. Two party immediate response disputes : Properties and efficiency. *Artificial Intelligence*, 149 :221–250, 2003.

[8] T. F. Gordon. The pleadings game. *Artificial Intelligence and Law*, 2 :239–292, 1993.

[9] M. Johnson, P. McBurney, and S. Parsons. When are two protocols the same ? In M-P. Huget, editor, *Communication in Multiagent Systems : Agent Communication Languages and Conversation Policies*, LNAI 2650, pages 253–268. Springer, 2003.

[10] S. Parsons and P. McBurney. Games that agents play : A formal framework for dialogues between autonomous agents. *J. of Logic, Language and Information*, 11(3) :315–334, 2002.

[11] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15 :1009–1040, 2005.

[12] P. Torroni. A study on the termination of negotiation dialogues. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1223–1230. ACM, 2002.

[13] S. Zabala, I. Lara, and H. Geffner. Beliefs, reasons and moves in a model for argumentative dialogues. In *Proc. 25th Latino-American Conf. on Computer Science*, 1999.

## Annexe : démonstrations

**Démonstration**[Propriété 2] Soient $D = \langle m_1, \ldots, m_n \rangle$ et $D' = \langle m'_0, \ldots, m'_k \rangle$ deux dialogues de persuasion tels que $D' \sqsubset D$, cela signifie que chaque $m'_i$ appartient à la séquence de $D$, donc $Content(m'_i) \in \texttt{Args}(D)$. Par conséquent, $\texttt{Args}(D') \subseteq \texttt{Args}(D)$. De plus, $\texttt{Confs}(D') \subseteq \texttt{Confs}(D)$. ■

**Démonstration**[Propriété 3] Soit $m$ un coup d'un dialogue de persuasion $D$. Si $m$ est utile alors il existe un chemin de $\texttt{Content}(m)$ à $\texttt{Subject}(D)$, donc $m$ est approprié pour $D$. ■

**Démonstration**[Propriété 4] Soit $m$ un coup décisif dans $D$, et supposons que $\texttt{Subject}(D)$ est accepté dans $\mathsf{AS}_D$. D'après la Propriété 1, $\texttt{Subject}(D)$ est indirectement défendu contre tous ses attaquants par un argument non attaqué. Puisque $m$ est décisif, $\texttt{Subject}(D)$ est rejeté dans $\mathsf{AS}_D \ominus \texttt{Content}(m)$. Cela signifie qu'au moins un attaquant n'est plus indirectement défendu par un argument non attaqué. Donc la suppression de $\texttt{Content}(m)$ a coupé un chemin allant d'un argument non-attaqué vers l'attaquant direct du sujet. Cela signifie que $\texttt{Content}(m)$ était utile dans $D$.

Si $\texttt{Subject}(D)$ est rejeté dans $\mathsf{AS}_D$, et accepté dans $\mathsf{AS}_D \ominus \texttt{Content}(m)$. Cela signifie que chaque attaquant est défendu par un argument non attaqué dans $\mathsf{AS}_D \ominus \texttt{Content}(m)$. Donc la suppression de $\texttt{Content}(m)$ a éliminé tous les attaquants directs ou indirects du sujet. Cela signifie que $\texttt{Content}(m)$ était sur un chemin entre un attaquant du sujet et le sujet, donc ce coup était utile dans $D$. ■

**Démonstration**[Propriété 5]
- Une branche de dialogue est non vide puisque le sujet du dialogue original de persuasion appartient à la branche.
- Supposons qu'il existe une branche infinie de dialogue pour un dialogue donné de persuasion $D$. Cela signifie qu'il existe une séquence infinie $\langle \alpha_0, \alpha_1, \ldots \rangle$ qui forme une branche de dialogue. Dans cette séquence, le nombre d'arguments d'indice pair et d'indice impair est infini. Selon la Définition 5, le dialogue de persuasion $D$ est fini, ainsi les deux ensembles $\texttt{Args}(D)$ et $\texttt{Confs}(D)$ sont finis. En conséquence, l'ensemble d'arguments qui appartiennent à la séquence $\langle \alpha_0, \alpha_1, \ldots \rangle$ est fini. Par conséquent, il y a au moins un argument d'indice pair qui est répété. Ce qui est impossible.

■

**Démonstration**[Propriété 6] Soit $D$ un dialogue de persuasion. Soit $\langle \alpha_0, ..., \alpha_k \rangle$ une branche de dialogue de l'arbre de dialogue associé à $D$. D'après la Définition 10.3, on déduit que $\forall i \in [\![1, k]\!], (\alpha_i, \alpha_{i-1}) \in Confs(D)$. Donc il existe un chemin de longueur $k$ dans $\mathsf{AS}_D$ de $\alpha_k$ à $\alpha_0$. D'après la Définition 10.2, $\alpha_0 = Subject(D)$. ■

**Démonstration**[Théorème 1] Soit $D$ un dialogue de persuasion et $\langle \alpha_0, \ldots \alpha_p \rangle$ une branche de dialogue de $D$. Si $\exists \beta \in \texttt{Args}(D)$ t.q. $(\beta, \alpha_p) \in \texttt{Confs}(D)$ alors une séquence commençant par $\langle \alpha_0, \ldots \alpha_p, \beta \rangle$ serait une branche ce qui est interdit par la Définition 10.6. ■

**Démonstration**[Propriété 7] Ce résultat découle directement de la définition de l'arbre de dialogue. En effet, la racine de l'arbre est le sujet du dialogue de persuasion et toutes les branches possibles sont considérées. ■

**Démonstration**[Théorème 2] Ce théorème montre qu'un arbre de dialogue est suffisant pour décider du statut du sujet. La preuve de ce théorème est basée sur deux théorèmes donnés plus loin qui se rapportent à la notion d'arbre canonique.
- si $\texttt{Subject}(D)$ est accepté dans $\mathsf{AS}_D$ alors, d'après le Théorème 4, il existe un arbre canonique $D^c_i$ tel que $\texttt{Subject}(D)$ est accepté dans $\mathsf{AS}^c_i$. De plus, la façon dont $D^c_i$ a été construit (par un procédé ET/OU) impose que $D^c_i$ contienne tous les fils directs du sujet dans $D^t$. Le Théorème 3 montre que toute branche de $D^c_i$ est de longueur paire. Chaque feuille de cet arbre canonique est par définition non attaquée dans $D^c_i$ et aussi par définition dans $\mathsf{AS}_{D^t}$. D'après la Définition 10.4, sur chaque branche de $\mathsf{AS}_{D^t}$ chaque noeud pair attaque strictement le noeud précédent. Par construction, pour chaque attaquant direct du sujet dans $\mathsf{AS}_{D^t}$, il existe au moins un défenseur non-attaqué dans $\mathsf{AS}_{D^t}$ (feuille de $D^c_i$), cette défense étant stricte, le sujet appartient donc à l'extension basique de $\mathsf{AS}_{D^t}$.
- si $\texttt{Subject}(D)$ est accepté dans $\mathsf{AS}_{D^t}$ alors il existe un défenseur non-attaqué contre tous les attaquant directs du sujet dans $\mathsf{AS}_{D^t}$. Cela signifie qu'il existe un arbre canonique issu

de $\mathsf{AS}_{D^t}$ dont toutes les branches sont de longueur paire. Le sujet est donc accepté dans cet arbre canonique d'après le théorème 3 ce qui implique que le sujet est accepté dans $D$ d'après le théorème 4.

■

**Démonstration**[Théorème 3] Soient $D$ un dialogue de persuasion, $D_i^c$ un arbre canonique et $\mathsf{AS}_i^c$ le système d'argumentation correspondant.

- Supposons que $\texttt{Subject}(D)$ est accepté dans $\mathsf{AS}_i^c$, et qu'il existe une branche de $D_i^c$ de longueur impaire. Cela signifie que la feuille de cette branche, disons $\alpha$, attaque indirectement $\texttt{Subject}(D)$ (la racine de la branche).
  - Soit $\alpha$ n'est pas attaquée dans $\mathsf{AS}_i^c$, alors cela implique qu'$\alpha$ est accepté et donc que le second noeud de la branche est un attaquant direct du sujet qui est non défendu par un argument non-attaqué, c.à.d., $\texttt{Subject}(D)$ n'est pas accepté dans $\mathsf{AS}_i^c$.
  - Soit $\alpha$ est attaqué dans $\mathsf{AS}_i^c$ alors il ne peut être attaqué que par un argument déjà présent dans la branche (donc lui-même attaqué), sinon la branche ne vérifierait pas la Définition10.6. Dans ce cas aussi le second noeud de cette branche est un attaquant direct du sujet qui n'est pas défendu par un argument non attaqué.
- Supposons maintenant que toues les branches de $D_i^c$ soient de longueur paire, alors chaque feuille de l'arbre est acceptée puisque non attaquée dans $\mathsf{AS}_i^c$ (d'après le Théorème 1). En considérant itérativement chaque noeud pair depuis la feuille jusqu'à la racine, elles peuvent toutes être ajoutées à l'extension basique car la feuille défend strictement l'avant dernier noeud pair contre l'attaque du dernier noeud impair et ainsi de suite sachant que par construction pour chaque noeud impair attaquant un noeud pair il existe un noeud pair de niveau plus profond qui le défend strictement (d'après la Définition 10.5). Ainsi chaque noeud pair est dans l'extension basique, donc le sujet est accepté dans $\mathsf{AS}_i^c$.

■

**Démonstration**[Corollaire 1] D'après le Théorème 3, puisque $\texttt{Subject}(D)$ est accepté dans $\mathsf{AS}_i^c$, alors toutes ses branches sont de longueur paire. D'après le Théorème 1, les feuilles de chaque branche de longueur paire ne sont pas attaquées dans $D$. Donc toutes les feuilles de $D_i^c$ sont non attaquées dans $D$. ■

**Démonstration**[Théorème 4]

- Supposons qu'il existe $D_j^c$ avec $1 \leq j \leq m$ et que $\texttt{Status}(\texttt{Subject}(D), \mathsf{AS}_j^c)$ est accepté. D'après le Théorème 3, cela signifie que toutes les branches de $D_j^c$ sont de longueur pair. D'après le Corollaire 1, cela implique que les feuilles de $D_j^c$ sont non attaquées dans le graphe du dialogue initial $D$.
  Soit $2i$ la profondeur de $D_j^c$ (c.à.d la longueur maximale d'une branche de $D_j^c$). Définissons la hauteur d'un noeud $N$ dans un arbre comme la profondeur du sous-arbre de racine $N$. Montrons par induction sur $p$ que $\forall p$ tel que $0 \leq p \leq i$, l'ensemble $\{y | y$ est un argument d'indice pair et dans un noeud de hauteur $\leq 2p$ appartenant à $D_j^c\}$ est inclus dans l'extension basique de $\mathsf{AS}_D$).
  - Cas $p = 0$. Les feuilles de $D_j^c$ ne sont pas attaquées dans $D$ (d'après le Corollaire 1). Donc, elles appartiennent à l'extension basique de $\mathsf{AS}_D$.
  - Supposons que la propriété est vraie au rang $p$ et montrons qu'elle est vraie au rang $p+1$.
    On peut ne considérer que les arguments des niveaux pairs dans les noeuds de hauteur $2p+2$ de $D_j^c$. Soit $y$ un tel argument. Puisqu'$y$ apparaît à un niveau pair, alors tous les arguments $y'$ attaquant $y$ dans $\mathsf{AS}_D$ apparaissent dans $D_j^c$ comme fils de $y$ (sinon la branche ne serait pas maximale ou $D_j^c$ ne serait pas canonique), et chaque $y'$ est lui-même strictement attaqué dans $\mathsf{AS}_D$ par exactement un argument $z$ apparaissant dans $D_j^c$ comme fils de $y'$. Ainsi chaque $z$ est à un niveau pair dans $D_j^c$ et apparaît comme noeud de hauteur $2p$ dans $D_j^c$. Par hypothèse d'induction, chaque argument $z$ est dans l'extension basique de $\mathsf{AS}_D$. Puisque tous les attaquants de $y$ ont été considérés, l'extension basique de $\mathsf{AS}_D$ défend $y$. Donc $y$ appartient aussi à l'extension basique.
- Supposons que $\texttt{Status}(\texttt{Subject}(D), \mathsf{AS}_D)$ est accepté. Soit $i_0$ le plus petit indice $\geq 0$ tel que $\texttt{Subject}(D) \in \mathcal{F}^{i_0}(\mathcal{C}^4)$. Montrons par induction sur $i$ que si un argument $\alpha \in \texttt{Args}(D)$ est dans $\mathcal{F}^i(\mathcal{C})$ alors il existe un arbre canonique de racine $\alpha$ pour $D$[5] ayant

[4] L'ensemble $\mathcal{C}$ contient tous les arguments qui ne sont pas attaqués dans $D$.

[5] Ici, on considère un "arbre canonique de racine $\alpha$ pour un dialogue $D$". Sa définition est plus générale qu'un arbre canonique pour un dialogue $D$ puisque cela ne requiert pas que toutes les branches commencent par le sujet du dialogue (cela modifie simplement le point 2 de la Défi-

une profondeur $\leq 2i$ et n'ayant que des branches de longueur paire.
- Cas $i = 0$ : si $\alpha \in \mathcal{C}$, alors $\alpha$ lui-même est un arbre canonique de racine $\alpha$ et profondeur 0.
- Supposons la propriété vraie au rang $i$ et considérons le rang $i+1$. Considérons $\alpha \in \mathcal{F}^{i+1}(\mathcal{C})$ et $\alpha \notin \mathcal{F}^{k}(\mathcal{C})$ avec $k < i+1$. Soit $x_1, \ldots, x_n$ les attaquants de $\alpha$. Considérons un attaquant $x_j$. $x_j$ attaque $\alpha$, et $\alpha \in \mathcal{F}^{i+1}(\mathcal{C}) = \mathcal{F}(\mathcal{F}^{i}(\mathcal{C}))$. D'après la Proposition 4.1 dans [1], il existe $y$ dans l'extension basique de $\mathsf{AS}_D$ tel que $y$ attaque strictement $x_j$. Puisque $y$ défend $\alpha$ (par définition de $\mathcal{F}$) alors $y \in \mathcal{F}^{i}(\mathcal{C})$. Par hypothèse d'induction appliquée à $y$, il existe un arbre canonique de racine $y$ et de profondeur $\leq 2i$. La même construction peut se faire pour chaque $x_j$. Ainsi, nous obtenons un arbre canonique de racine $\alpha$ et de profondeur $\leq 2(i+1)$ et qui a toutes ses branches de longueur paire.

Finalement, du fait que $\texttt{Subject}(D) \in \mathcal{F}^{i_0}(\mathcal{C})$ on peut conclure qu'il existe un arbre canonique de racine $\texttt{Subject}(D)$ ayant toutes ses branches de longueur paire. D'après le Théorème 3, on obtient que $\texttt{Subject}(D)$ est accepté dans cet arbre canonique.

■

**Démonstration**[Théorème 5] Par construction de $D_i^c$, il existe un chemin dans cet arbre de la racine à chaque argument $\alpha$ de l'arbre canonique. D'après la propriété 6, nous obtenons qu'il existe un chemin correspondant dans $\mathsf{AS}_D$ de $\alpha$ à $\texttt{Subject}(D)$, par conséquent un coup contenant l'argument $\alpha$ est utile dans $D$. ■

**Démonstration**[Théorème 6] Si un coup $m$ est décisif alors, comme vu dans la preuve de la propriété 4,
- si le sujet est accepté dans $\mathsf{AS}_D$ alors cela signifie qu'il existe au moins un attaquant qui n'est plus indirectement défendu par un argument non attaqué dans $\mathsf{AS}_D \ominus \texttt{Content}(m)$. Le sujet étant accepté dans $\mathsf{AS}_D$ cela signifie qu'il existe un arbre canonique dont toutes les branches sont de longueur paire (d'après le Théorème 3). Par construction ce graphe canonique contient tous les attaquants directs du sujet. Si $\texttt{Content}(m)$ n'appartient pas à ce graphe canonique alors il existe un défenseur du sujet sur un chemin qui ne passe pas par $\texttt{Content}(m)$ dans $\mathsf{AS}_D$, si c'est le cas pour tous les attaquants directs du sujet alors le sujet aurait du être accepté dans $\mathsf{AS}_D \ominus \texttt{Content}(m)$. C'est impossible, donc $\texttt{Content}(m)$ appartient au graphe canonique qui accepte le sujet.
- si le sujet est rejeté dans $\mathsf{AS}_D$ mais accepté dans $\mathsf{AS}_D \ominus \texttt{Content}(m)$ alors il existe un arbre canonique dont toutes les branches sont de longueur pair dans $\mathsf{AS}_D \ominus \texttt{Content}(m)$. Puisque l'ajout de $\texttt{Content}(m)$ rend le sujet rejeté, cela signifie que $content(m)$ attaque au moins un défenseur direct ou indirect du sujet présent dans tout arbre canonique acceptant le sujet dans $\mathsf{AS}_D \ominus \texttt{Content}(m)$. La séquence contenant la branche partant du sujet jusqu'à ce défenseur peut être prolongée avec $\texttt{Content}(m)$ pour former une branche de longueur impaire dans $\mathcal{D}^t$. Donc pour que tous les arbres canoniques rejettent le sujet il faut que $\texttt{Content}(m)$ appartienne à une de leur branche.

■

**Démonstration**[Propriété 8] Si le sujet est accepté dans $D$ alors, par construction, un arbre canonique de $D$ contient chaque argument existant dans $D$ qui attaque directement le sujet puisqu'ils appartient à toutes les branches possibles de dialogue qui peuvent être construites à partir de $D$. Mais pour chacune d'elles il contient seulement un attaquant qui est en faveur du sujet (cet attaquant est un fils d'un noeud OU dans l'arbre de dialogue), pour chaque argument choisi en faveur du sujet, tous les attaquants sont présent dans l'arbre canonique (ils sont les fils d'un noeud "ET" dans l'arbre de dialogue). Par ailleurs, si le sujet est accepté alors chaque branche de l'arbre canonique est de longueur paire. Cela signifie que les feuilles sont en faveur du sujet et sont non attaquées dans le dialogue initial $D$. Cette propriété vaut pour n'importe quel arbre canonique. Donc, puisque le dialogue idéal correspond au plus petit arbre canonique, cela signifie que c'est le dialogue le plus court qui satisfait cette propriété. ■

nition 10) mais on requiert que toutes les branches commencent par le noeud $\alpha$.

# An axiomatic account of preference-based argumentation systems

Leila Amgoud* Srdjan Vesic*
amgoud@irit.fr vesic@irit.fr

*IRIT – CNRS
118, route de Narbonne
31062, Toulouse Cedex 09 – FRANCE

**Abstract:**
Argumentation is a reasoning model based on the construction and evaluation of arguments. Dung has proposed an abstract argumentation framework in which arguments are assumed to have the same strength. This assumption is unfortunately not realistic. Consequently, three main extensions of the framework have been proposed in the literature. The basic idea is that if an argument is stronger than its attacker, the attack fails.

The aim of the paper is twofold : First, it shows that the three extensions of Dung framework may lead to unintended results. Second, it proposes a new approach that takes into account the strengths of arguments, and that ensures sound results. We start by presenting two minimal requirements that any preference-based argumentation framework should satisfy, namely the conflict-freeness of arguments extensions and the generalization of Dung's framework. Inspired from works on handling inconsistency in knowledge bases, the proposed approach defines a binary relation on the powerset of arguments. The maximal elements of this relation represent the extensions of the new framework.

**Keywords:** Argumentation, Preferences

## 1 Introduction

*Preferences* are used in most models that have been developed to solve conflicts (e.g. [6, 8, 9, 11]). Conflicts have a better chance to be solved in presence of such information. Preferences have also been introduced into *argumentation* theory. Argumentation is a reasoning model based on the construction and the evaluation of arguments. An argument gives a reason to believe a statement, to perform an action, or to choose an option, etc. Due to its explanatory power, argumentation is gaining an increasing interest in Artificial Intelligence, namely for handling inconsistency (e.g. [1, 7, 14]), and decision making (e.g. [3]). Most of the models that treat the cited applications are instantiations of an abstract framework developed in [10]. This framework consists of a set of arguments and a binary relation that captures attacks among arguments. Arguments are assumed to have all the same strength. This assumption is unfortunately not realistic since it may be the case that an argument relies on certain information, while another argument is built from less certain ones. The former is clearly stronger than the latter. In [6, 9, 13, 14] different preference relations between arguments have been defined. A preference relation captures differences in arguments' strengths.

In [2], a first extension of Dung's framework has been proposed. It takes as input a set of arguments, an attack relation, and a preference relation between arguments. This relation is abstract and can be instantiated in different ways. This proposal has recently been generalized in [12] in order to reason even about preferences. Thus, arguments may support preferences about arguments. The last extension of Dung's framework has been proposed in [5]. It assumes that each argument promotes a value, and a preference between two arguments comes from the importance of the respective values that are promoted by the two arguments. Whatever the source of the preference relation is, the idea behind the three extensions is that an attack from an argument $a$ to an argument $b$ fails if $b$ is stronger than $a$.

The aim of the paper is twofold : First, it shows that while the above idea is interesting and seems meaningful, the three extensions return unintended results. Second, it proposes a new preference-based argumentation framework (PAF) that ensures sound results. We start by defining two basic requirements that any PAF should satisfy. Namely, the extensions should be conflict-free w.r.t. the attack relation. The second requirement consists of recovering Dung's acceptability extensions in case preferences are not available. We then propose a new approach which defines a binary relation on the powerset of arguments. The maximal elements of this relation are the extensions of the new framework. Three relations are particularly proposed in the paper. They capture respectively stable extensions, preferred extensions and grounded extensions of Dung's framework.

The paper is organized as follows : Section 2 recalls briefly Dung's framework as well as its extensions with preferences. Section 3 presents

their limits through a simple example. Section 4 develops the new approach, and Section 5 concludes.

## 2 Dung's Framework And Its Extensions

In the seminal paper [10], an *argumentation framework* is a pair $\mathcal{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{A}$ is a set of arguments and $\mathcal{R}$ is a binary relation between arguments, representing attacks among them ($\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$). The notation $(a, b) \in \mathcal{R}$ or $a\mathcal{R}b$ means that the argument $a$ attacks the argument $b$. Different acceptability semantics for evaluating arguments have been proposed. Before recalling them, let us first define the notions of conflict-free and defense.

**Definition 1 (Conflict-free, Defense)** *Let $\mathcal{B} \subseteq \mathcal{A}$ and $a \in \mathcal{A}$.*
- *$\mathcal{B}$ is* conflict-free *iff $\nexists\ a, b \in \mathcal{B}$ s.t. $a\mathcal{R}b$.*
- *$\mathcal{B}$* defends *$a$ iff $\forall b \in \mathcal{A}$ if $b\mathcal{R}a$, then $\exists c \in \mathcal{B}$ s.t. $c\mathcal{R}b$.*

The main semantics introduced by Dung are recalled in the following definition.

**Definition 2 (Acceptability semantics)** *Let $\mathcal{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework, and $\mathcal{B}$ be a conflict-free set of arguments.*
- *$\mathcal{B}$ is a* admissible *iff it defends all its elements.*
- *$\mathcal{B}$ is a* preferred extension *iff it is a maximal (w.r.t. set $\subseteq$) admissible set.*
- *$\mathcal{B}$ is a* stable extension *iff it is a preferred extension that attacks any argument in $\mathcal{A} \setminus \mathcal{B}$.*
- *$\mathcal{B}$ is a* grounded extension*, denoted* GE*, iff $\mathcal{B}$ is the least fixpoint of a function $\mathcal{F}$ where $\mathcal{F}(S) = \{a \in \mathcal{A} \mid S$ defends $a\}$, for $S \subseteq \mathcal{A}$.*

In [2], a first extension of Dung's framework has been proposed. It takes as input a set $\mathcal{A}$ of arguments, an attack relation $\mathcal{R}$, and a (partial or total) preorder[1] $\geq$ on $\mathcal{A}$. This preorder is a preference relation between arguments. The expression $(a, b) \in \geq$ or $a \geq b$ means that the argument $a$ is at least as strong as $b$. The symbol $>$ denotes the strict relation associated with $\geq$. Indeed, $a > b$ iff $a \geq b$ and not $(b \geq a)$. From the two relations $\mathcal{R}$ and $\geq$, a new binary relation, Def, is defined as follows : $a$ Def $b$ iff $a\mathcal{R}b$ and not $(b > a)$. This means that among all the attacks in $\mathcal{R}$, only the ones that hold between incomparable and indifferent arguments and the ones that agree with the preference relation are kept. In order to evaluate the acceptability of the arguments, Dung's acceptability semantics are applied to the framework $\langle \mathcal{A}, \texttt{Def} \rangle$.

[1] A binary relation is a *preorder* iff it is *reflexive* and *transitive*.

In [12], the preference relation $\geq$ is given by arguments.The idea is that an argument may support a preference between two other arguments. Two attack relations are assumed : a classical one denoted by $\mathcal{R}$, and another relation, $\mathcal{D}$, that ranges from an argument of $\mathcal{A}$ to an element of $\mathcal{R}$. An expression $(a, (b, c))$ means that the argument $a$ supports a preference of $c$ over $b$. This preference conflicts with the fact that $b$ attacks $c$. A new relation, Def, is defined as follows : $a$ $\texttt{Def}_{\mathcal{S}}$ $b$ iff $a\mathcal{R}b$ and $\exists c \in \mathcal{S}$ such that $(c, (a, b)) \in \mathcal{D}$, where $\mathcal{S} \subseteq \mathcal{A}$.

The extension proposed in [5], called value-based framework, assumes that a set $\mathcal{V}$ of values is available. Each argument in $\mathcal{A}$ promotes one value given by a function val (i.e. $\texttt{val} : \mathcal{A} \mapsto \mathcal{V}$). The values may not have the same importance and this is captured by a binary relation Pref. This latter is assumed to be irreflexive, asymmetric and transitive. Like in [1], a new relation, called defeats, is defined as follows : $(a, b) \in \texttt{defeats}$ iff $(a, b) \in \mathcal{R} \wedge (\texttt{val}(b), \texttt{val}(a)) \notin \texttt{Pref}$. Dung's acceptability semantics are applied to the framework $\langle \mathcal{A}, \texttt{defeats} \rangle$.

## 3 Critical Examples

This section shows through a simple example that the above two extensions may lead to counter-intuitive results.

Let us consider a case of an agent who wants to buy a given violin. An expert says that the violin in question is produced by Stradivari ($s$), that's why it is expensive ($s \rightarrow e$). This agent has thus an argument $a_1$ whose conclusion is "the violin is expensive". Suppose now that the 3-years old son of this agent says that the violin was not produced by Stradivari ($\neg s$). Thus, an argument $a_2$ which attacks $a_1$ is given. In sum, $\mathcal{A} = \{a_1, a_2\}$ and $\mathcal{R} = \{(a_2, a_1)\}$. According to Dung's framework, argument $a_2$ wins. This is inadmissible, especially since it is clear that an argument of an expert is stronger than an argument given by a 3-years old child. In the framework presented in [1], the fact that $a_1$ is

stronger than $a_2$ is taken into account. Thus, the relation $\geq = \{(a_1, a_1), (a_2, a_2), (a_1, a_2)\}$ is available. However, in this framework the relation Def is empty. Consequently, both arguments are in the unique preferred extension of the framework $\langle \mathcal{A}, \texttt{Def} \rangle$. This means that this extension is not conflict-free. Moreover, both $s$ and $\neg s$ are deduced.

According to [12], there are three arguments : $a_1$, $a_2$ and $a_3$ where $a_3$ expresses the fact that $a_1$ is strictly preferred to $a_2$. Thus, $\mathcal{A} = \{a_1, a_2, a_3\}$, $\mathcal{R} = \{(a_2, a_1)\}$, and $\mathcal{D} = \{(a_3, (a_2, a_1))\}$. The set $\{a_1, a_2, a_3\}$ is a preferred extension which is not conflict-free w.r.t. $\mathcal{R}$.

The same problem holds in the value-based framework of [5]. Assume that the set of values is $V = \{expert, child\}$ and, of course, $(expert, child) \in \texttt{Pref}$. The value of $a_1$ is expert while the value of $a_2$ is child. The new relation defeats is empty. So, like with the previous preference-based framework, the two arguments appear in the same extension which is not conflict-free.

## 4 A New Approach

The previous section has highlighted the limits of existing preference-based argumentation frameworks. Even if the idea pursued by these frameworks is intuitive and meaningful, their results are not satisfactory and violate a key property. This property concerns the conflict-freeness of their extensions w.r.t. the attack relation $\mathcal{R}$. In this section, we propose a new preference-based argumentation framework. It takes as input three elements : a set $\mathcal{A}$ of arguments, an attack relation $\mathcal{R}$, and a (partial or total) preorder $\geq$. It returns *extensions* that are subsets of $\mathcal{A}$. These extensions satisfy the two following basic requirements :

**Conflict-freeness :** If $\mathcal{E}$ is an extension of $(\mathcal{A}, \mathcal{R}, \geq)$, then $\mathcal{E}$ is conflict free w.r.t. $\mathcal{R}$.

**Generalization :** If $(\nexists a, b \in \mathcal{A})$ s.t. $(a, b) \in \mathcal{R}$ and $(b, a) \in >$, then any extension of $(\mathcal{A}, \mathcal{R}, \geq)$ is also an extension of Dung's framework $\langle \mathcal{A}, \mathcal{R} \rangle$ and vice versa.

The first requirement ensures that the extensions returned by the new framework are conflict-free. This is important since it ensures safe results in the sense that inconsistent conclusions are avoided. The second one captures the idea that an attack fails in case the attacker is weaker than its target. Moreover, it states that the proposed approach extends Dung's framework, i.e. it refines its acceptability semantics.

In what follows, we show how extensions of a PAF are computed. We follow the same reasoning as in some approaches developed for handling inconsistency in knowledge bases, namely the coherence-based ones. In [9], for instance, an inconsistent knowledge base $\Sigma$ is equipped with a partial or total preorder, meaning that formulas of $\Sigma$ have not the same priority. Then, preference relations among consistent sub-bases of $\Sigma$ are defined. The maximal elements w.r.t. those preference relations represent the preferred ones. We apply the same idea in an argumentation context. Indeed, by analogy, the inconsistent base represents our conflicting set of arguments, and the priorities between formulas of $\Sigma$ are the preferences between arguments. We define preference relations, denoted by $\succeq$, between the different conflict-free sets of arguments. Thus, $\succeq \subseteq 2^{\mathcal{A}} \times 2^{\mathcal{A}}$. The relation $\succ$ is the strict version of $\succeq$, that is for $\mathcal{E}, \mathcal{E}' \subseteq \mathcal{A}$, $\mathcal{E} \succ \mathcal{E}'$ iff $\mathcal{E} \succeq \mathcal{E}'$ and not $(\mathcal{E}' \succeq \mathcal{E})$. The maximal elements of $\succeq$ are the extensions of a PAF. This notion of maximality is defined as follows.

**Definition 3 (Maximal elements)** *Let $\mathcal{E}$ be a conflict-free set of arguments. $\mathcal{E}$ is* maximal *w.r.t. $\succeq$ iff :*

1. $(\forall \mathcal{E}' \subseteq \mathcal{A})$ $((\mathcal{E}'$ *is conflict-free)* $\Rightarrow (\mathcal{E} \succeq \mathcal{E}'))$
2. *No strict superset of $\mathcal{E}$ is conflict-free and verifies* $(1)$

*Let $\succeq_{max}$ denote the set of maximal sets w.r.t. $\succeq$.*

The above definition privileges maximal (for set inclusion) sets of arguments among the conflict-free ones. It is worth mentioning that different relations $\succeq$ can be defined, and may lead to different sets of extensions. Moreover, those extensions are not necessarily the ones got by Dung's acceptability semantics. The new preference-based argumentation framework is defined as follows.

**Definition 4 (PAF)** *A* PAF *is a tuple $(\mathcal{A}, \mathcal{R}, \geq)$, where $\mathcal{A}$ is a set of arguments, $\mathcal{R}$ is an attack relation, and $\geq$ is a (partial or total) preorder on $\mathcal{A}$. Extensions of $(\mathcal{A}, \mathcal{R}, \geq)$ are the maximal*

*elements of a relation $\succeq \subseteq 2^{\mathcal{A}} \times 2^{\mathcal{A}}$ that satisfies the two basic requirements.*

In what follows, we propose three relations which generalize respectively stable, preferred and grounded semantics. Note that, without loss of generality, we assume that there are no self-attacking arguments, i.e., $(\nexists x \in \mathcal{A})$ s.t. $(x, x) \in \mathcal{R}$. Moreover, the set $\mathcal{A}$ is assumed to be finite.

### 4.1 Generalization Of Stable Semantics

This section presents a relation $\succeq$ that generalizes stable semantics. The idea behind this relation is the following : given two conflict-free sets of arguments, $\mathcal{E}$ and $\mathcal{E}'$, we say that $\mathcal{E}'$ is better then $\mathcal{E}$ iff any argument in $\mathcal{E} \setminus \mathcal{E}'$ is weaker than at least one argument in $\mathcal{E}' \setminus \mathcal{E}$ or is attacked by it. Formally :

**Definition 5** *Let $\mathcal{E}, \mathcal{E}'$ be two conflict-free sets of arguments. $\mathcal{E}' \succeq \mathcal{E}$ iff $(\forall x \in \mathcal{E} \setminus \mathcal{E}')$ $(\exists x' \in \mathcal{E}' \setminus \mathcal{E})$ s.t. $(((x', x) \in \mathcal{R} \wedge (x, x') \notin >) \vee (x', x) \in >)$.*

Let us illustrate this definition through the following simple example.

**Example 1** *Let $\mathcal{A} = \{a, b, c\}$, $\geq = \{(a, a), (b, b), (a, b)\}$ and $\mathcal{R} = \{(a, b), (b, a), (b, c), (c, b)\}$. The conflict-free sets of arguments are : $\mathcal{E}_1 = \emptyset$, $\mathcal{E}_2 = \{a\}$, $\mathcal{E}_3 = \{b\}$, $\mathcal{E}_4 = \{c\}$, and $\mathcal{E}_5 = \{a, c\}$. It can be checked that the following relations hold : $\mathcal{E}_2 \succeq \mathcal{E}_1$, $\mathcal{E}_3 \succeq \mathcal{E}_1$, $\mathcal{E}_4 \succeq \mathcal{E}_1$, $\mathcal{E}_5 \succeq \mathcal{E}_1$, $\mathcal{E}_5 \succeq \mathcal{E}_4$, $\mathcal{E}_5 \succeq \mathcal{E}_2$, $\mathcal{E}_5 \succeq \mathcal{E}_3$, $\mathcal{E}_4 \succeq \mathcal{E}_3$, $\mathcal{E}_3 \succeq \mathcal{E}_4$, and $\mathcal{E}_2 \succeq \mathcal{E}_3$. It can be checked that $\succeq_{max} = \{\mathcal{E}_5\}$.*

Note that relation $\succeq$ is not transitive. However, the following property states that this relation privileges maximal for set inclusion elements.

**Property 1** *Let $\mathcal{E}, \mathcal{E}'$ be conflict-free sets of arguments. If $\mathcal{E} \subsetneq \mathcal{E}'$ then $\mathcal{E}' \succ \mathcal{E}$.*

**Proof** *Let us prove that $\mathcal{E}' \succeq \mathcal{E}$. We have that $\mathcal{E} \setminus \mathcal{E}' = \emptyset$, and, consequently, there are no arguments in $\mathcal{E} \setminus \mathcal{E}'$. Let us now see why $\neg(\mathcal{E} \succeq \mathcal{E}')$. Since $\mathcal{E} \subsetneq \mathcal{E}'$, then $\exists x' \in \mathcal{E}' \setminus \mathcal{E}$. But, from the fact that $\mathcal{E} \setminus \mathcal{E}'$ is empty, we conclude that $\nexists x \in \mathcal{E} \setminus \mathcal{E}'$ s.t. $(x, x') \in >$ or $((x, x') \in \mathcal{R} \wedge (x', x) \notin >)$.* ■

With the above relation, Definition 3 can be simplified.

**Property 2** *Let $\mathcal{E}'$ be a conflict-free set of arguments. It holds that $\mathcal{E}' \in \succeq_{max}$ iff $(\forall \mathcal{E} \subseteq \mathcal{A})$ ($\mathcal{E}$ is conflict-free) $\Rightarrow (\mathcal{E}' \succeq \mathcal{E})$.*

**Proof** $\Rightarrow$ *Trivial, according to Definition 3.*

$\Leftarrow$ *Let $(\forall \mathcal{E} \subseteq \mathcal{A})$ ($\mathcal{E}$ is conflict-free) $\Rightarrow (\mathcal{E}' \succeq \mathcal{E})$. We will prove that $(\nexists \mathcal{E}'' \subseteq \mathcal{A})$ s.t. $\mathcal{E}''$ is conflict-free $\wedge$ $\mathcal{E}' \subsetneq \mathcal{E}''$ $\wedge$ $(\forall \mathcal{E}''' \subseteq \mathcal{A})$ ($\mathcal{E}'''$ conflict-free $\Rightarrow \mathcal{E}'' \succeq \mathcal{E}'''$). Suppose the contrary. Since $\mathcal{E}' \subsetneq \mathcal{E}''$ then Property 1 implies that $\neg(\mathcal{E}' \succeq \mathcal{E}'')$. Contradiction.* ■

The following property shows that the maximal sets of arguments w.r.t. the relation $\succeq$ given in Definition 5 are maximal conflict-free subsets of $\mathcal{A}$.

**Property 3** *Let $\mathcal{E}$ be a conflict-free set of arguments. If $\mathcal{E} \in \succeq_{max}$, then $\mathcal{E}$ is a maximal conflict-free set.*

**Proof** *Suppose the contrary, i.e., that $\mathcal{E} \in \succeq_{max}$ and that $\mathcal{E}$ is not a maximal conflict-free set. This means that $(\exists x \in \mathcal{A})$ s.t. $x \notin \mathcal{E}$ and $\mathcal{E} \cup \{x\}$ is conflict-free. According to Property 1, $(\mathcal{E} \cup \{x\}) \succ \mathcal{E}$. Contradiction with the fact $\mathcal{E} \in \succeq_{max}$.* ■

The converse is not true, as illustrated by the next example.

**Example 2 (Ex. 1 Cont.)** *The set $\mathcal{E}_3$ is maximal conflict-free but does not belong to $\succeq_{max}$.*

We can show that the proposed framework handles correctly the example discussed in Section 3.

**Example 3** *Recall that $\mathcal{A} = \{a_1, a_2\}$, $\mathcal{R} = \{(a_2, a_1)\}$ and $\geq = \{(a_1, a_1), (a_2, a_2), (a_1, a_2)\}$. Conflict-free sets are : $\mathcal{E}_1 = \emptyset$, $\mathcal{E}_2 = \{a_1\}$, $\mathcal{E}_3 = \{a_2\}$. It can easily be checked that $\succeq_{max} = \{\mathcal{E}_2\}$. Thus, the new PAF has a unique extension which is $\{a_1\}$.*

The extensions of the new PAF are conflict-free w.r.t the attack relation $\mathcal{R}$.

**Property 4** *Let $(\mathcal{A}, \mathcal{R}, \geq)$ be a preference-based argumentation framework. The extensions of this framework w.r.t. the relation $\succeq$ given in Definition 5 are conflict-free w.r.t. $\mathcal{R}$.*

**Proof** *This follows from the definition of $\succeq$.* ■

Regarding the second requirement, the following theorem proves that the extensions of our preference-based argumentation framework coincide with stable extensions in case preferences are not available and when any attacked argument is not stronger than its attacker.

**Theorem 1** *Let $(\mathcal{A}, \mathcal{R}, \geq)$ be a preference-based argumentation framework, and $\mathcal{E}_1, \ldots, \mathcal{E}_n$ denote its extensions w.r.t. $\succeq$. If $(\nexists x, y \in \mathcal{A})$ s.t. $(x, y) \in \mathcal{R} \wedge (y, x) \in >$, then each $\mathcal{E}_i$ is a stable extension of $\langle \mathcal{A}, \mathcal{R} \rangle$ and vice versa.*

**Proof** $\Rightarrow$ *Let $\mathcal{E}' \in \succeq_{max}$.*
- *Since $\mathcal{E}' \in \succeq_{max}$ then it is conflict-free.*
- *We will now prove that $\mathcal{E}'$ defends all its elements. Let us suppose that $(\exists a \in \mathcal{E}')$ $(\exists x \in \mathcal{A})$ s.t. $(x, a) \in \mathcal{R} \wedge (\nexists y \in \mathcal{E}')$ $(y, x) \in \mathcal{R}$. Since $\mathcal{E}'$ is conflict-free, then $x \notin \mathcal{E}'$. Let $\mathcal{E} = \{x\} \cup \{t \in \mathcal{E}' \mid (x, t) \notin \mathcal{R} \wedge (t, x) \notin \mathcal{R}\}$. It is clear the $\mathcal{E}$ is conflict-free since $\mathcal{E}$ is union of two conflict-free sets which do not attack one another. Since $\mathcal{E}' \in \succeq_{max}$ then $\mathcal{E}' \succeq \mathcal{E}$. In particular, since $x \in \mathcal{E} \backslash \mathcal{E}'$, then $(\exists x' \in \mathcal{E}' \backslash \mathcal{E})$ s.t. $((x', x) \in \mathcal{R} \wedge (x, x') \notin >) \vee (x', x) \in >$. Since $(\nexists y \in \mathcal{E}')$ $(y, x) \in \mathcal{R}$, then it must be the case that $(x', x) \notin \mathcal{R}$ and $(x', x) \in >$. Since $x' \in \mathcal{E}'$ and $x' \notin \mathcal{E}$ then, with respect to definition of $\mathcal{E}$, from $x' \notin \mathcal{E}$ we have that $(x, x') \in \mathcal{R}$ or $(x', x) \in \mathcal{R}$. Since we have just seen that $(x', x) \notin \mathcal{R}$, it must be that $(x, x') \in \mathcal{R}$. Recall that we have $(x', x) \in >$. But we supposed that $(\nexists z, z' \in \mathcal{A})$ s.t. $(z, z') \in \mathcal{R}$ and $(z', z) \in >$. Contradiction. Thus, $\mathcal{E}'$ defends its arguments.*
- *We have just shown that $\mathcal{E}'$ is admissible, i.e., it is conflict-free and it defends all its arguments. We will now prove that $\mathcal{E}'$ attacks all arguments in $\mathcal{A} \setminus \mathcal{E}'$. Let $x \notin \mathcal{E}'$ be an argument and suppose that $(\nexists y \in \mathcal{E}')$ $(y, x) \in \mathcal{R}$. Either $x$ attacks some argument of $\mathcal{E}'$ or not. If it is the case, i.e., $(\exists a \in \mathcal{E}')$ s.t. $(x, a) \in \mathcal{R}$ then, since $\mathcal{E}'$ defends all its elements, it holds that $(\exists y \in \mathcal{E}')$ s.t. $(y, x) \in \mathcal{R}$. Contradiction. So, it must be that $(\nexists a \in \mathcal{E}')$ s.t. $(x, a) \in \mathcal{R}$. This means that $\mathcal{E} = \mathcal{E}' \cup \{x\}$ is conflict-free. According to Property 1, it holds that $\neg(\mathcal{E}' \succeq \mathcal{E})$. Contradiction with the fact that $\mathcal{E}' \in \succeq_{max}$.*
  *So, $\mathcal{E}$ is conflict-free and it attacks all arguments in $\mathcal{A} \setminus \mathcal{E}$. This means that $\mathcal{E}$ is a stable extension of framework $\mathcal{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$.*

$\Leftarrow$ *Let $\mathcal{E}'$ be a stable extension of the framework $\mathcal{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$ and let us prove that $\mathcal{E}' \in \succeq_{max}$.*
- *Since $\mathcal{E}'$ is stable then it is conflict-free.*
- *We will prove that for an arbitrary conflict-free set of arguments $\mathcal{E}$ it holds that $\mathcal{E}' \succeq \mathcal{E}$. Let $\mathcal{E} \subseteq \mathcal{A}$ be a conflict-free set. If $\mathcal{E} \setminus \mathcal{E}' = \emptyset$ the proof is over. If it is not the case, let $x \in \mathcal{E} \setminus \mathcal{E}'$. Since $x \notin \mathcal{E}'$ and $\mathcal{E}'$ is a stable extension, then $(\exists x' \in \mathcal{E}')$ s.t. $(x', x) \in \mathcal{R}$. We supposed that $(\nexists z, z' \in \mathcal{A})$ s.t. $(z, z') \in \mathcal{R}$ and $(z', z) \in >$. Thus, $(x, x') \notin >$. Since $x \in \mathcal{E} \setminus \mathcal{E}'$ was arbitrary, it holds that $\mathcal{E}' \succeq \mathcal{E}$.*
- *Using Property 2, we conclude that $\mathcal{E}' \in \succeq_{max}$.*

■

From this result it follows that when preferences are not available, stable extensions are retrieved.

**Corollary 1** *Let $(\mathcal{A}, \mathcal{R}, \geq)$ be a preference-based argumentation framework, and $\mathcal{E}_1, \ldots, \mathcal{E}_n$ denote its extensions w.r.t. $\succeq$. If $\geq = \{(x, x) \mid x \in \mathcal{A}\}$, then each $\mathcal{E}_i$ is a stable extension of $\langle \mathcal{A}, \mathcal{R} \rangle$ and vice versa.*

**Proof** *Since $\geq = \{(x, x) \mid x \in \mathcal{A}\}$ then $(\nexists x, y \in \mathcal{A})$ s.t. $x \neq y \wedge (x, y) \in \mathcal{R} \wedge (y, x) \in >$. Since we supposed that $(\nexists x \in \mathcal{A})$ s.t. $(x, x) \in \mathcal{R}$ then $(\nexists x, y \in \mathcal{A})$ s.t. $(x, y) \in \mathcal{R} \wedge (y, x) \in >$. Thus, Theorem 1 implies that extensions of $(\mathcal{A}, \mathcal{R}, \geq)$ are exactly the stable extensions of $\langle \mathcal{A}, \mathcal{R} \rangle$.* ■

Note that the relation $\succeq$ gives more information than Dung's acceptability semantics. Indeed, even when preferences are not available, the relation $\succeq$ compares conflict-free sets of arguments, as can be shown on the following example.

**Example 4** *Let $\mathcal{A} = \{a, b, c\}$, $\geq = \{(a, a), (b, b), (c, c)\}$ and $\mathcal{R} = \{(a, b), (b, c)\}$. Note that in this case, the preference relation $\geq$ is useless. Thus, the only stable extension of $\langle \mathcal{A}, \mathcal{R} \rangle$ is $\{a, c\}$. This is also the only maximal element of the relation $\succeq$. However, in the new PAF, it is also possible to compare the two sets : $\{a\}$ and $\{b\}$. It can be checked that $\{a\} \succeq \{b\}$.*

### 4.2 Generalization Of Preferred Semantics

In this section, we define a relation $\succeq$ that allows to retrieve preferred extensions in case preferences between arguments are not available or are not important. The basic idea behind this relation is that a set $\mathcal{E}'$ is better than $\mathcal{E}$ iff for every attack from $\mathcal{E}$ to $\mathcal{E}'$ which does not fail $\mathcal{E}'$ is capable to defend the attacked argument and that for every attack from $\mathcal{E}'$ to $\mathcal{E}$ which fails, there is another attack from $\mathcal{E}'$ which defends the argument which failed in its attack.

**Definition 6** *Let $\mathcal{E}, \mathcal{E}'$ be conflict-free sets of arguments. $\mathcal{E}' \succeq \mathcal{E}$ iff $(\forall x' \in \mathcal{E}')(\forall x \in \mathcal{E})$ if $(((x, x') \in \mathcal{R} \wedge (x', x) \notin >)$ or $((x', x) \in \mathcal{R} \wedge (x, x') \in >))$ then $((\exists y' \in \mathcal{E}')$ s.t. $((y', x) \in \mathcal{R} \wedge (x, y') \notin >))$.*

Let us illustrate this definition through the next example.

**Example 5 (Ex. 1 Cont.)** *One can easily see that it holds that $\mathcal{E}_2 \succ \mathcal{E}_3$, $\mathcal{E}_3 \succeq \mathcal{E}_4$, $\mathcal{E}_4 \succeq \mathcal{E}_3$, $\mathcal{E}_5 \succ \mathcal{E}_3, \ldots$ It can also be checked that $\succeq_{max} = \{\mathcal{E}_5\}$.*

Note that this relation is not transitive. It is also clear from the above definition that the corresponding framework satisfies the conflict-freeness requirement.

**Property 5** *Let $(\mathcal{A}, \mathcal{R}, \geq)$ be a preference-based argumentation framework. The extensions of this framework w.r.t. $\succeq$ given in Definition 6 are conflict-free w.r.t. $\mathcal{R}$.*

**Proof** *This follows from the definition of $\succeq$.* ■

Regarding the second requirement, the following theorem shows that the preference-based framework that uses this relation generalizes preferred extensions.

**Theorem 2** *Let $(\mathcal{A}, \mathcal{R}, \geq)$ be a preference-based argumentation framework, and $\mathcal{E}_1, \ldots, \mathcal{E}_n$ denote its extensions w.r.t. $\succeq$. If $(\nexists x, y \in \mathcal{A})$ s.t. $(x, y) \in \mathcal{R} \wedge (y, x) \in >$, then each $\mathcal{E}_i$ is a preferred extension of $\langle \mathcal{A}, \mathcal{R} \rangle$ and vice versa.*

**Proof** *Since we supposed that $(\nexists x, y \in \mathcal{A})$ s.t. $(x, y) \in \mathcal{R} \wedge (y, x) \in >$ then $\mathcal{E}' \succeq \mathcal{E}$ iff $(\forall x' \in \mathcal{E}')$ $(\forall x \in \mathcal{E})$ if $(x, x') \in \mathcal{R}$ then $(\exists y' \in \mathcal{E}')$ s.t. $(y, x) \in \mathcal{R}$.*

*$\Leftarrow$ Let $\mathcal{E}'$ be a preferred extension of $\mathcal{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$.*

- *Since $\mathcal{E}'$ is a preferred extension then it is conflict-free.*
- *Let us prove that $\mathcal{E}' \in \succeq_{max}$. Suppose the contrary. This means that one of the following is true :*
  1. *$(\exists \mathcal{E} \subseteq \mathcal{A})$ s.t. $\mathcal{E}$ is conflict-free and $\neg(\mathcal{E}' \succeq \mathcal{E})$*
  2. *$(\exists \mathcal{E} \subseteq \mathcal{A})$ s.t. $\mathcal{E}$ is conflict-free $\wedge\ \mathcal{E}' \subsetneq \mathcal{E} \wedge (\forall \mathcal{E}'' \subseteq \mathcal{A})$ if $\mathcal{E}''$ is conflict-free then $\mathcal{E} \succeq \mathcal{E}''$*

  *Let $(1)$ be the case. Since $\neg(\mathcal{E}' \succeq \mathcal{E})$ then $(\exists x' \in \mathcal{E}')(\exists x \in \mathcal{E})$ s.t. $(x, x') \in \mathcal{R} \wedge (\nexists y' \in \mathcal{E}')$ s.t. $(y', x) \in \mathcal{R}$. This leads to the conclusion that $\mathcal{E}'$ does not defend its arguments, thus it cannot be a preferred extension. Contradiction. So, it must be that $(2)$ holds. Since $\mathcal{E}'$ is preferred and $\mathcal{E}' \subsetneq \mathcal{E}$ then $\mathcal{E}$ is not admissible. From the fact that $\mathcal{E}$ is conflict-free, one concludes that it does not defend its arguments. Thus, $(\exists x'' \in \mathcal{E}'' \setminus \mathcal{E}')$ s.t. $(\exists y \in \mathcal{A})$ s.t. $(y, x'') \in \mathcal{R} \wedge (\nexists z'' \in \mathcal{E}'')$ s.t. $(z'', y) \in \mathcal{R}$. Hence, $\neg(\mathcal{E}'' \succeq \{y\})$. Contradiction.*

*$\Rightarrow$ Let $\mathcal{E}' \in \succeq_{max}$. We will prove that $\mathcal{E}'$ is a preferred extension of Dung's argumentation framework $\mathcal{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$.*

- *Since $\mathcal{E}' \in \succeq_{max}$ then it is conflict-free.*
- *Let us prove that $\mathcal{E}'$ defends all its arguments. Suppose not. This means that $(\exists y \in \mathcal{A})$ s.t. $(y, x') \in \mathcal{R} \wedge (\nexists z' \in \mathcal{E}')$ s.t. $(z', y) \in \mathcal{R}$. This means that $\neg(\mathcal{E}' \succeq \{y\})$. Contradiction.*
- *We have just seen that $\mathcal{E}'$ is admissible. Let us prove that $\mathcal{E}'$ is a preferred extension of $\mathcal{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$. Suppose the contrary, i.e., $(\exists \mathcal{E} \subseteq \mathcal{A})$ s.t. $\mathcal{E}$ is a preferred extension and $\mathcal{E}' \subsetneq \mathcal{E}$. Since $\mathcal{E}' \in \succeq_{max}$ then $\mathcal{E} \notin \succeq_{max}$. On the other hand, since $\mathcal{E}$ is a preferred extension, then $\mathcal{E} \in \succeq_{max}$, as we have proved in the first part of this theorem. Contradiction.*

■

When preferences are not available, the framework that uses the relation $\succeq$ retrieves preferred extensions.

**Corollary 2** *Let $(\mathcal{A}, \mathcal{R}, \geq)$ be a preference-based argumentation framework, and $\mathcal{E}_1, \ldots, \mathcal{E}_n$ denote its extensions w.r.t. $\succeq$. If*

$\geq = \{(x,x) \mid x \in \mathcal{A}\}$, *then each* $\mathcal{E}_i$ *is a preferred extension of* $\langle \mathcal{A}, \mathcal{R} \rangle$ *and vice versa.*

**Proof** *Since* $\geq = \{(x,x) \mid x \in \mathcal{A}\}$ *then* $(\nexists x, y \in \mathcal{A})$ *s.t.* $x \neq y \wedge (x,y) \in \mathcal{R} \wedge (y,x) \in >$. *Since we supposed that* $(\nexists x \in \mathcal{A})$ *s.t.* $(x,x) \in \mathcal{R}$ *then* $(\nexists x, y \in \mathcal{A})$ *s.t.* $(x,y) \in \mathcal{R} \wedge (y,x) \in >$. *Theorem 2 now implies that extensions of* $(\mathcal{A}, \mathcal{R}, \geq)$ *are exactly the preferred extensions of* $\langle \mathcal{A}, \mathcal{R} \rangle$. ■

### 4.3 Generalization Of Grounded Semantics

In this section, we define a relation $\succeq$ that allows to retrieve the grounded extension in case preferences between arguments are not available or are not important. The basic idea behind this relation is that a set is not worse than another if it can strongly defend all its arguments against all attacks that come from another set.

We first generalize the notion of strong defense by taking into account preferences between arguments. The idea is that an argument has either to be preferred to its attacker or has to be defended by arguments that themselves can be strongly defended without using the argument in question.

**Definition 7 (Strong defense)** *Let* $\mathcal{E}' \subseteq \mathcal{A}$. $\mathcal{E}'$ strongly defends *an argument* $x$ *from attacks of a set* $\mathcal{E}$, *denoted by* $sd(x, \mathcal{E}', \mathcal{E})$ *iff* $(\forall y \in \mathcal{E})$ *if* $(((y,x) \in \mathcal{R} \wedge (x,y) \notin >)$ *or* $((x,y) \in \mathcal{R} \wedge (y,x) \in >))$ *then* $((\exists z \in \mathcal{E}' \setminus \{x\})$ *s.t.* $((z,y) \in \mathcal{R} \wedge (y,z) \notin > \wedge sd(z, \mathcal{E}' \setminus \{x\}, \mathcal{E})))$.
*If the third argument of* $sd$ *is not specified, then* $sd(x, \mathcal{E}) \equiv sd(x, \mathcal{E}, \mathcal{A})$.

Let us illustrate this notion through the following example.

**Example 6 (Ex. 1 Cont.)** *It holds that* $sd(a, \{a\}, \{b\})$ *since* $a$ *is strictly preferred to* $b$ *thus it can defend itself. However, we have* $\neg sd(b, \{b\}, \{c\})$ *since* $b$ *cannot defend itself against* $c$. *On the other hand, it does hold that* $sd(c, \{a,c\}, \{b\})$ *since* $a$ *can defend* $c$ *against* $b$ *and* $a$ *is protected from* $b$ *since it is strictly preferred to it.*

This relation amounts to prefer the subsets that strongly defend all their arguments. In particular, $\mathcal{E}' \succeq \mathcal{E}$ iff $\mathcal{E}'$ strongly defends all its arguments against all attacks of $\mathcal{E}$.

**Definition 8** *Let* $\mathcal{E}, \mathcal{E}'$ *be conflict-free sets of arguments. We say that* $\mathcal{E}' \succeq \mathcal{E}$ *iff* $(\forall x' \in \mathcal{E}')$ $sd(x', \mathcal{E}', \mathcal{E})$.

**Example 7** *Let* $\mathcal{A} = \{a,b,c\}$, $\geq = \{(a,a),(b,b),(b,a)\}$ *and* $\mathcal{R} = \{(a,b),(b,a),(b,c),(c,b)\}$. *One can check that there is exactly one subset of* $\mathcal{A}$ *which is preferred to all other subsets of arguments. This set is the empty one. While we do have* $\{b\} \succeq \{a\}$, *we have* $\neg(\{b\} \succeq \{c\})$, *so* $\{b\}$ *is not an extension of* $(\mathcal{A}, \mathcal{R}, \geq)$. *We have also* $\neg(\{a\} \succeq \{b\})$, $\neg(\{c\} \succeq \{b\})$ *and* $\neg(\{a,c\} \succeq \{b\})$. *This is expected and a natural output since neither* $b$ *nor* $c$ *are capable to defend strongly themselves and, on the other hand, it can be said that* $a$ *is the worst argument in this framework, thus not strong enough to be better than* $b$.

**Theorem 3** *Let* $(\mathcal{A}, \mathcal{R}, \geq)$ *be a preference-based argumentation framework. If* $(\nexists x, y \in \mathcal{A})$ *s.t.* $(x,y) \in \mathcal{R} \wedge (y,x) \in >$, *then* $\succeq_{max}$ *contains exactly one element which coincides with the grounded extension of* $\langle \mathcal{A}, \mathcal{R} \rangle$.

**Proof** *Since we supposed that* $(\nexists x, y \in \mathcal{A})$ *s.t.* $(x,y) \in \mathcal{R} \wedge (y,x) \in >$ *then we can simplify Definition 7 which becomes :* $sd(x, \mathcal{E}', \mathcal{E})$ *iff* $(\forall y \in \mathcal{E})$ *(if* $(y,x) \in \mathcal{R}$ *then* $(\exists z \in \mathcal{E}' \setminus \{x\})$ *s.t.* $((z,y) \in \mathcal{R} \wedge sd(z, \mathcal{E}' \setminus \{x\}, \mathcal{E})))$. *In this particular case when no attacked argument is strictly preferred to its attacker, our definition of* $sd(x, \mathcal{E})$ *becomes exactly the same as Definition 13 in [4]. Thus, using Proposition 50 and Proposition 51 of the same paper, we conclude that* $x \in \mathtt{GE}$ *iff* $sd(x, \mathtt{GE})$, *where* GE *is the grounded extension of the framework* $\mathcal{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$. $\Leftarrow$ *Let* $\mathcal{E}'$ *be the grounded extension of* $\langle \mathcal{A}, \mathcal{R} \rangle$.

- *Since* $\mathcal{E}'$ *is the grounded extension then it is conflict-free.*
- *We will prove that for an arbitrary conflict-free set* $\mathcal{E} \subseteq \mathcal{A}$ *it holds that* $\mathcal{E}' \succeq \mathcal{E}$. *Let* $\mathcal{E} \subseteq \mathcal{A}$ *be conflict-free. Since* $\mathcal{E}'$ *is the grounded extension then* $x \in \mathcal{E}' \Rightarrow sd(x, \mathcal{E}')$. *On the other hand,* $(\forall x \in \mathcal{E}')$ $sd(x, \mathcal{E}')$ *implies that* $sd(x, \mathcal{E}', \mathcal{E})$. *Thus,* $\mathcal{E}' \succeq \mathcal{E}$. *Since* $\mathcal{E}$ *was arbitrary, then* $(\forall \mathcal{E} \subseteq \mathcal{A})$ $((\mathcal{E}$ *is conflict-free*$)$ $\Rightarrow (\mathcal{E}' \succeq \mathcal{E}))$.
- *We will now prove that* $(\nexists \mathcal{E} \subseteq \mathcal{A})$ *s.t.* $\mathcal{E}$ *is conflict-free and* $\mathcal{E}' \subsetneq \mathcal{E}$ *and* $((\forall \mathcal{E}'' \subseteq \mathcal{A})$ $(\mathcal{E}''$ *conflict-free*$)$ $\Rightarrow$ $(\mathcal{E} \succeq \mathcal{E}''))$. *Suppose the contrary. Suppose also that* $(\forall x \in \mathcal{E})$ $sd(x, \mathcal{E})$. *If this is the case, according to Proposition 51 in [4],* $\mathcal{E} \subseteq \mathtt{GE}$. *Contradiction. So,*

*it must be that* $(\exists x \in \mathcal{E})$ *s.t.* $\neg sd(x, \mathcal{E})$. *Thus,* $(\exists y \in \mathcal{A})$ *s.t.* $\neg sd(x, \mathcal{E}, \{y\})$. *Consequently,* $\neg(\mathcal{E} \succeq \{y\})$. *Contradiction. So, we have proved that* $\mathcal{E}' \in \succeq_{max}$.

$\Rightarrow$ *Let* $\mathcal{E}' \in \succeq_{max}$ *and let us prove that* $\mathcal{E}' =$ GE. *Since* $(\forall x \in \mathcal{A})$ $\mathcal{E}' \succeq \{x\}$ *then* $(\forall x' \in \mathcal{E}')$ $sd(x', \mathcal{E}')$. *From the fact that* $(\forall x' \in \mathcal{E}')$ $sd(x, \mathcal{E}')$ *and Proposition 51 of [4] we have that* $\mathcal{E}' \subseteq$ GE. *Let us now prove that* $\mathcal{E}' =$ GE. *Suppose not, i.e., suppose that* $\mathcal{E}' \subsetneq$ GE. *We have proved in the first part of this theorem that* GE $\in \succeq_{max}$. *Contradiction, since we have supposed that* $\mathcal{E}' \in \succeq_{max}$ *and we have* $\mathcal{E}' \subsetneq$ GE. ■

When preferences are not available, the framework that uses the relation $\succeq$ retrieves exactly grounded extension.

**Corollary 3** *Let* $(\mathcal{A}, \mathcal{R}, \geq)$ *be a preference-based argumentation framework. If* $\geq = \{(x, x) \mid x \in \mathcal{A}\}$, *then* $\succeq_{max}$ *contains exactly one element which coincides with the grounded extension of* $\langle \mathcal{A}, \mathcal{R} \rangle$.

**Proof** *Since* $\geq = \{(x, x) \mid x \in \mathcal{A}\}$ *then* $(\nexists x, y \in \mathcal{A})$ *s.t.* $x \neq y \wedge (x, y) \in \mathcal{R} \wedge (y, x) \in >$. *Since we supposed that* $(\nexists x \in \mathcal{A})$ *s.t.* $(x, x) \in \mathcal{R}$ *then* $(\nexists x, y \in \mathcal{A})$ *s.t.* $(x, y) \in \mathcal{R} \wedge (y, x) \in >$. *Theorem 3 now implies that* $(\mathcal{A}, \mathcal{R}, \geq)$ *has exactly one extension which is the grounded extension of* $\langle \mathcal{A}, \mathcal{R} \rangle$. ■

## 5 Conclusion

This paper has shown through a simple example that existing preference-based argumentation frameworks may lead to undesirable results. This means that the way preferences between arguments are taken into account is not appropriate. We have then proposed an alternative approach that satisfies two basic requirements : conflict-freeness of extensions, and recovering Dung's acceptability semantics when preferences are not available. The approach amounts to define a relation on the powerset of arguments. In other words, it compares pairs of conflict-free subsets of arguments. The best elements w.r.t. this relation are the extensions of the new framework. The approach has three main advantages : i) it is general since different relations can be defined, ii) it enforces the new framework to satisfy key properties, namely conflict-freeness of the extensions and recovering Dung's semantics, iii) it allows to compare any pair of subsets of arguments, contrary to Dung's approach in which there are only two categories of sets : the ones that are considered as extensions and all the remaining ones. The results presented in this paper show also how to characterize Dung's semantics in terms of a relation between subsets of arguments. To the best of our knowledge this is the first work in this direction. It allows to better understand the underpinning of those semantics.

## Références


[1] L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *J. of Automated Reasoning*, 29 (2) :125–169, 2002.

[2] L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34 :197–216, 2002.

[3] L. Amgoud and H. Prade. Using arguments for making and explaining decisions. *Artificial Intelligence J.*, 173 :413–436, 2009.

[4] P. Baroni and M. Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence J.*, 171 :675–700, 2007.

[5] T. J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3) :429–448, 2003.

[6] S. Benferhat, D. Dubois, and H. Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proc. of UAI'93*, pages 411–419, 1993.

[7] P. Besnard and A. Hunter. *Elements of Argumentation*. MIT Press, 2008.

[8] G. Brewka. Preferred subtheories : An extended logical framework for default reasoning. In *Proc. of IJCAI-89*, pages 1043–1048, 1989.

[9] C. Cayrol, V. Royer, and C. Saurel. Management of preferences in assumption-based reasoning. *Lecture Notes in Computer Science*, 682 :13–22, 1993.

[10] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and $n$-person games. *Artificial Intelligence J.*, 77 :321–357, 1995.

[11] M. Gelfond and T. Son. Prioritized default theory. *Lecture Notes in AI*, 1471, 1997.

[12] S. Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence J.*, 2009.

[13] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *J. of Applied Non-Classical Logics*, 7 :25–75, 1997.

[14] G. Simari and R. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence J.*, 53 :125–157, 1992.

# Résultats de complexité pour le problème de la composition d'agents

P. Balbiani
Philippe.Balbiani@irit.fr

F. Cheikh
Fahima.Cheikh@irit.fr

G. Feuillade
Guillaume.Feuillade@irit.fr

CNRS- Université de Toulouse
Institut de recherche en informatique de Toulouse

**Résumé :**
Pour que des agents puissent réaliser des tâches complexes, il est nécessaire de composer entre eux ces agents. La composition d'agents a été caractérisée comme étant un problème de synthèse de plan ou comme étant un problème de synthèse de logiciel : étant donnés un but et un ensemble d'agents, générer une composition des agents qui satisfait le but. Nous proposons des algorithmes, basés sur des résultats récents de synthèse automatique de contrôleurs, pour réaliser la composition automatique d'agents. Nous donnons également des résultats concernant la complexité de ce problème.

**Mots-clés :** Composition d'agents, synthèse de contrôleurs, complexité et automates.

**Abstract:**
In order to realize complex tasks by agents, it is sometimes necessary to compose them. The agent composition has been characterized either as a plan synthesis problem or as a software synthesis problem : given a goal and a set of agents, generate a composition of the agents that satisfies the goal. We propose algorithms, based on recent advances in automated synthesis of controllers, for performing automated composition of agents. We also give complexity result concerning this problem.

**Keywords:** Composition of agents, controller synthesis, complexity and automata.

## 1 Introduction

Considérons une communauté d'agents, chacun capable de réaliser une tâche simple et un objectif qui est une tâche complexe. Aucun agent n'étant capable de réaliser seul l'objectif, il est nécessaire de faire coopérer les agents pour le mener à bien.

Plusieurs approches permettent de fabriquer automatiquement une solution permettant à la communauté d'agents de réaliser l'objectif. L'approche que nous privilégions dans cet article consiste à considérer les agents comme des services [7, 10]. Nous cherchons alors à composer ces services de telle manière que la composition obtenue réalise l'objectif. La composition se fait alors en fabriquant un agent virtuel, nommé *médiateur*, dont le rôle est de communiquer avec les agents de la communauté pour les amener à se coordonner afin de réaliser l'objectif.

Différentes techniques automatisées ont été proposées pour résoudre le problème de la composition de services [12]. On distingue principalement deux types de méthode de résolution. Le premier consiste à réduire le problème de la composition à un problème de synthèse de plan [14, 15, 18]. Ces méthodes permettent de traiter des problématiques complexes, cependant, la complexité des algorithmes de résolution n'a pas été étudiée en détail. Le deuxième consiste à faire appel aux méthodes de synthèse de logiciel [3, 4, 5]. En pratique cette approche a été utilisée dans le cadre de la composition de services Web ; malheureusement, les solutions reposent sur des restrictions fortes, comme par exemple l'absence de communication entre agents.

Dans cet article, les agents sont abstraits sous la forme d'automates capables de communiquer entre eux. L'échange d'information se fait sous la forme de messages envoyés via des canaux de communication. Nous ne détaillons pas le contenu des messages et chaque canal contient au plus un unique message. Ces restrictions de communication sont nécessaires d'une part pour garantir l'existence d'un algorithme de résolution et d'autre part pour permettre l'étude de la complexité précise du problème.
La tâche à accomplir est donnée sous la forme d'un automate. L'objectif de la composition est de synthétiser un agent médiateur, qui peut uniquement communiquer avec les agents de la communauté afin de garantir que la communauté plus l'agent médiateur ait un comportement similaire ou équivalent à la tâche à accomplir. Nous étudions alors la complexité de ce problème pour les notions de similarité que sont la simulation d'automates ou l'inclusion de traces ainsi que pour les notions d'équivalence que sont la bisimulation d'automates ou l'égalité de traces. Dans la plupart des cas nous proposons un algorithme de résolution.

L'article se présente comme suit. Dans la section 2 un exemple de scénario est présenté. La section 3 définit les agents, leur mode de communication ainsi que les notions de similarité et d'équivalence. La section 4 introduit le problème de la composition : étant donnés un but

et un ensemble d'agents, générer une composition des agents qui satisfait le but. Des résultats de complexités et des algorithmes sont donnés dans la section 5. La section 6 conclut l'article ; nous y présentons certains problèmes qui restent ouverts.

## 2 Exemple

Pour cet article, nous proposons un exemple sous la forme d'un scénario dans lequel deux agents doivent gérer une situation où il faut sauver un individu de la noyade. Ces deux agents sont hétérogènes, de ce fait leurs canaux de communication ne sont pas forcément les mêmes. Le premier agent intitulé agent *Pilote* peut effectuer trois actions :

- *?Region* : recevoir un message indiquant la position approximative de l'individu en danger,
- *LocaliseVict* : déterminer la position exacte de la victime et
- *!Position* : envoyer un rapport pour confirmer sa position exacte.

Chacune de ces actions est complexe et représente en réalité une séquence d'actions conditionnées. Nous nous intéressons cependant ici à une abstraction de ces actions, que nous considérons comme atomiques.

Le second agent intitulé agent *Sauveteur* peut effectuer les actions suivantes :

- *?Données* : recevoir la position exacte de la victime,
- *RecupVict* : récupérer la victime et
- *!Rapport* : envoyer un rapport pour signaler la réussite ou l'échec de la mission.

Le problème que nous traitons dans cet article est le suivant : étant donné un agent appelé agent *but* qui représente la ou les séquences d'actions souhaitées, créer de façon automatique un agent médiateur, qui communique avec les agents du système afin que le système composé du médiateur et des deux agents soit équivalent à celui de l'agent but (au sens de l'inclusion de traces, l'équivalence de trace, la simulation et la bisimulation). Dans cet exemple, l'agent but devra à la fois déterminer la position de la victime et la récupérer. Notons que l'agent but n'est pas réel. Notons également que pour des raisons de clarté, cet exemple est restrictif par rapport aux problématiques traités dans cet article : il ne présente que 2 agents, pas de communication directe entre agents et les agents sont déterministes.

## 3 Automates finis et agents

Dans cette section, nous présentons notre modèle des agents basé sur des automates finis communiquant de manière asynchrone. Tout d'abord, nous donnons les définitions utiles concernant les automates finis. Ensuite, nous définissons formellement les agents.

### 3.1 Automate fini

Soit $\Sigma$ un ensemble fini d'actions. Un automate fini défini sur $\Sigma$ est une structure $\mathcal{A} = (S, \Delta, s^{in})$ où $S$ est un ensemble fini d'états, $\Delta$ est une fonction $\Delta : S \times \Sigma \to 2^S$ et $s^{in} \in S$ est un état initial. Pour tout $\Sigma' \subseteq \Sigma$, la relation $\to_{\mathcal{A}}^{\Sigma'} \subseteq S \times S$ décrit le changement d'état de l'automate suite à l'exécution d'une action de $\Sigma'$. Formellement, $s \to_{\mathcal{A}}^{\Sigma'} t$ ssi il existe $a \in \Sigma'$ tel que $t \in \Delta(s, a)$. On note $\to_{\mathcal{A}}^{\Sigma'\star}$ la fermeture reflexive transitive de $\to_{\mathcal{A}}^{\Sigma'}$. Pour tout $\Sigma' \subseteq \Sigma$, nous dirons que $\mathcal{A}$ *boucle* sur $\Sigma'$ ssi pour tout $a \in \Sigma'$, $\to_{\mathcal{A}}^{\{a\}} = Id_S$.

Soit $\Sigma'$ un ensemble fini d'actions considérées comme étant non-observables, un chemin pour $\mathcal{A}$ modulo $\Sigma'$ est une séquence finie de la forme $(s^{in}, a_1, s^1)$, $(s^1, a_2, s^2)$,..., $(sn-1, a_n, s^n)$ telle que $s^{in} \to_{\mathcal{A}}^{\Sigma'\star} \circ \to_{\mathcal{A}}^{\{a_1\}} \circ \to_{\mathcal{A}}^{\Sigma'\star} s^1$ et pour tout $i \in \{1, \ldots, n-1\}$, $s^i \to_{\mathcal{A}}^{\Sigma'\star} \circ \to_{\mathcal{A}}^{\{a_{i+1}\}} \circ \to_{\mathcal{A}}^{\Sigma'\star} s^{i+1}$. Le mot $a_1 \ldots a_n$ est sa *trace*. L'ensemble des traces de tous les chemins pour $\mathcal{A}$ modulo $\Sigma'$ est noté $Tr_{\Sigma'}(\mathcal{A})$.

### 3.2 Inclusion et équivalence de traces

Soient $\mathcal{A}_1 = (S_1, \Delta_1, s_1^{in})$ et $\mathcal{A}_2 = (S_2, \Delta_2, s_2^{in})$ des automates finis définis sur $\Sigma$. Pour tout $\Sigma' \subseteq \Sigma$ Nous dirons que $\mathcal{A}_1$ est *inclus pour la trace* dans $\mathcal{A}_2$ modulo $\Sigma'$, noté $\mathcal{A}_1 \subseteq_{tr} \mathcal{A}_2(\Sigma')$, lorsque $Tr_{\Sigma'}(\mathcal{A}_1) \subseteq Tr_{\Sigma'}(\mathcal{A}_2)$. Nous dirons que $\mathcal{A}_1$ et $\mathcal{A}_2$ sont *équivalents pour la trace* modulo $\Sigma'$, noté $\mathcal{A}_1 \equiv_{tr} \mathcal{A}_2(\Sigma')$, lorsque $Tr_{\Sigma'}(\mathcal{A}_1) = Tr_{\Sigma'}(\mathcal{A}_2)$. Considérons le cas où les automates finis $\mathcal{A}_1$ et $\mathcal{A}_2$ sont représentés par la Figure 1. Dans ce cas, les deux automates sont équivalents pour la trace modulo $\emptyset$. En effet, $Tr_\emptyset(\mathcal{A}_1) = Tr_\emptyset(\mathcal{A}_2) = \{\epsilon, a, ab\}$.

### 3.3 Simulation et Bisimulation

Soient $\mathcal{A}_1 = (S_1, \Delta_1, s_1^{in})$ et $\mathcal{A}_2 = (S_2, \Delta_2, s_2^{in})$ deux automates finis définis sur $\Sigma$. Pour tout

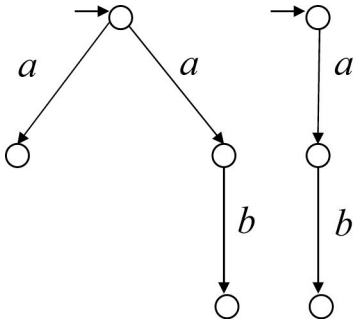



FIG. 1 – De gauche à droite, automates finis $\mathcal{A}_1$ et $\mathcal{A}_2$

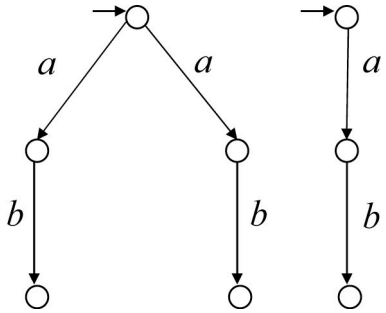



FIG. 2 – De gauche à droite, automates finis $\mathcal{A}_1$ et $\mathcal{A}_2$

$\Sigma' \subseteq \Sigma$, une relation $Z \subseteq S_1 \times S_2$ telle que $(s_1^{in}, s_2^{in}) \in Z$ est appelée *simulation* entre $\mathcal{A}_1$ et $\mathcal{A}_2$ modulo $\Sigma'$ ssi les conditions suivantes sont satisfaites pour tout $(s_1, s_2) \in Z$ et pour tout $a \in \Sigma \setminus \Sigma'$ :

– pour tout $t_1 \in S_1$, si $s_1 \rightarrow_{\mathcal{A}_1}^{\Sigma'\star} \circ \rightarrow_{\mathcal{A}_1}^{\{a\}} \circ \rightarrow_{\mathcal{A}_1}^{\Sigma'\star} t_1$ alors il existe $t_2 \in S_2$ tel que $s_2 \rightarrow_{\mathcal{A}_2}^{\Sigma'\star} \circ \rightarrow_{\mathcal{A}_2}^{\{a\}} \circ \rightarrow_{\mathcal{A}_2}^{\Sigma'\star} t_2$ et $(t_1, t_2) \in Z$,

De plus, pour tout $\Sigma' \subseteq \Sigma$, s'il y a une simulation entre $\mathcal{A}_1$ et $\mathcal{A}_2$ modulo $\Sigma'$ alors nous écrirons $\mathcal{A}_1 \leq_{si} \mathcal{A}_2(\Sigma')$ et nous dirons que $\mathcal{A}_1$ est *simulé* par $\mathcal{A}_2$.

Une relation $Z \subseteq S_1 \times S_2$ telle que $(s_1^{in}, s_2^{in}) \in Z$ est appelée *bisimulation* entre $\mathcal{A}_1$ et $\mathcal{A}_2$ modulo $\Sigma'$ ssi les conditions suivantes sont satisfaites pour tout $(s_1, s_2) \in Z$ et pour tout $a \in \Sigma \setminus \Sigma'$ :

– pour tout $t_1 \in S_1$, si $s_1 \rightarrow_{\mathcal{A}_1}^{\Sigma'\star} \circ \rightarrow_{\mathcal{A}_1}^{\{a\}} \circ \rightarrow_{\mathcal{A}_1}^{\Sigma'\star} t_1$ alors il existe $t_2 \in S_2$ tel que $s_2 \rightarrow_{\mathcal{A}_2}^{\Sigma'\star} \circ \rightarrow_{\mathcal{A}_2}^{\{a\}} \circ \rightarrow_{\mathcal{A}_2}^{\Sigma'\star} t_2$ et $(t_1, t_2) \in Z$,

– pour tout $t_2 \in S_2$, si $s_2 \rightarrow_{\mathcal{A}_2}^{\Sigma'\star} \circ \rightarrow_{\mathcal{A}_2}^{\{a\}} \circ \rightarrow_{\mathcal{A}_2}^{\Sigma'\star} t_2$ alors il existe $t_1 \in S_1$ tel que $s_1 \rightarrow_{\mathcal{A}_1}^{\Sigma'\star} \circ \rightarrow_{\mathcal{A}_1}^{\{a\}} \circ \rightarrow_{\mathcal{A}_1}^{\Sigma'\star} t_1$ et $(t_1, t_2) \in Z$.

De plus, pour tout $\Sigma' \subseteq \Sigma$, s'il y a une bisimulation entre $\mathcal{A}_1$ et $\mathcal{A}_2$ modulo $\Sigma'$ alors nous écrirons $\mathcal{A}_1 \longleftrightarrow_{bi} \mathcal{A}_2\ (\Sigma')$ et nous dirons que $\mathcal{A}_1$ et $\mathcal{A}_2$ sont *bisimilaires*.

Rappelons que dans le cas où les automates finis $\mathcal{A}_1$ et $\mathcal{A}_2$ sont déterministes, l'inclusion de trace (resp. équivalence de trace) et la simulation (resp. bisimulation) sont la même relation, ce n'est pas le cas pour les automates non déterministes. La relation de simulation (resp. bisimulation) est plus restrictive que celle de l'inclusion de trace (resp. l'équivalence de trace). Par exemple, dans la Figure 1, l'automate $\mathcal{A}_1$ simule l'automate $\mathcal{A}_2$ ; en revanche ces deux automates ne sont pas bisimilaires. En effet, après avoir effectué l'action $a$ l'automate $\mathcal{A}_1$ peut se retrouver dans un état où il ne peut plus effectuer l'action $b$ ; ce n'est pas le cas pour l'automate $\mathcal{A}_2$. Dans la Figure 2, les automates $\mathcal{A}_1$ et $\mathcal{A}_2$ sont bisimilaires.

### 3.4 Produit

Soient $\mathcal{A}_1 = (S_1, \Delta_1, s_1^{in})$ et $\mathcal{A}_2 = (S_2, \Delta_2, s_2^{in})$ deux automates finis sur $\Sigma$. Par $\mathcal{A}_1 \otimes \mathcal{A}_2$, nous notons le *produit asynchrone* de $\mathcal{A}_1$ et $\mathcal{A}_2$, c.-à-d. l'automate fini $\mathcal{A} = (S, \Delta, s^{in})$ sur $\Sigma$ tel que $S = S_1 \times S_2$, $\Delta$ est la fonction définie par $(t_1, t_2) \in \Delta((s_1, s_2), a)$ ssi $t_1 \in \Delta_1(s_1, a)$ et $t_2 = s_2$ ou $t_1 = s_1$ et $t_2 \in \Delta_2(s_2, a)$ et $s^{in} = (s_1^{in}, s_2^{in})$. Par $\mathcal{A}_1 \times \mathcal{A}_2$, nous notons le *produit synchrone* de $\mathcal{A}_1$ et $\mathcal{A}_2$, c.-à-d. l'automate fini $\mathcal{A} = (S, \Delta, s^{in})$ sur $\Sigma$ tel que $S = S_1 \times S_2$, $\Delta$ est la fonction définie par $(t_1, t_2) \in \Delta((s_1, s_2), a)$ ssi $t_1 \in \Delta_1(s_1, a)$ et $t_2 \in \Delta_2(s_2, a)$ et $s^{in} = (s_1^{in}, s_2^{in})$.

### 3.5 Modèle des agents

Dans les travaux concernant la composition de services Web qui ont été réalisé par [3, 4, 5], les services sont représentés par des automates finis à entrée/sortie. Les services communiquent par la transmission asynchrone de messages à travers des canaux de communication. Notre modèle pour les agents s'inspire de ces travaux. Comme leurs auteurs, nous supposons que la communication à travers les canaux est fiable (il n'y a pas de perte de messages durant leurs transmission). Pour simplifier, dans cet article nous considérons une abstraction du contenu des messages. De plus, nous supposons que les canaux ne peuvent pas contenir plus d'un message à la fois. Formellement, soit $\Pi$ un ensemble fini de canaux et $\Sigma$ un ensemble fini d'actions, un *agent* défini sur $\Pi$ et $\Sigma$ est un automate fini

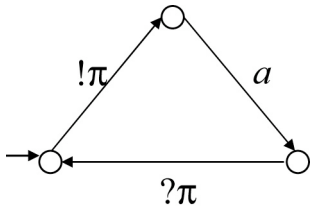



FIG. 3 –

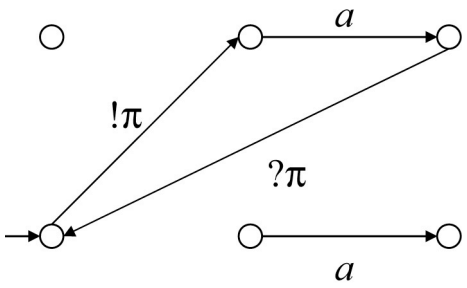



FIG. 4 –

défini sur $(\{!,?\} \times \Pi) \cup \Sigma$. Pour tout $\pi \in \Pi$, l'action de transmission $!\pi$ consiste à ajouter un message sur le canal $\pi$, tandis que l'action de réception $?\pi$ consiste à retirer un message du canal $\pi$. L'action $!\pi$ peut être exécutée à condition que le canal ne soit pas plein, c.-à-d. $\pi$ ne contient pas de message. L'action $?\pi$ peut être exécutée à condition que le canal ne soit pas vide, c.-à-d. $\pi$ contient un message.

Soit $\mathcal{A} = (S, \Delta, s^{in})$ un agent défini sur $(\{!,?\} \times \Pi) \cup \Sigma$. L'ensemble des *exécutions* de l'agent $\mathcal{A}$ est un automate fini $Exec(\mathcal{A}) = (S', \Delta', s^{in'})$ défini sur $(\{!,?\} \times \Pi) \cup \Sigma$ tel que $S' = S \times 2^{\Pi}$, $\Delta'$ est la fonction définie par
– $(t, Q) \in \Delta'((s, P), !\pi)$ ssi $t \in \Delta(s, !\pi)$, $\pi \notin P$, $Q = P \cup \{\pi\}$,
– $(t, Q) \in \Delta'((s, P), ?\pi)$ ssi $t \in \Delta(s, ?\pi)$, $\pi \in P$, $Q = P \setminus \{\pi\}$,
– $(t, Q) \in \Delta'((s, P), a)$ ssi $t \in \Delta(s, a)$, $Q = P$,
et $s^{in'} = (s^{in}, \emptyset)$ (initialement tous les canaux sont vides). Observons que $Exec(\mathcal{A})$ est de taille exponentielle par rapport à $\Pi$ et que nous pouvons le construire en un temps exponentiel. Considérons le cas où $\mathcal{A}$ est l'automate fini représenté dans la Figure 3. Dans ce cas $Exec(\mathcal{A})$ est l'automate fini représenté dans la Figure 4.

## 4 La composition d'agents

L'objectif de la composition est donné sous la forme d'un but. Le problème s'énonce de la manière suivante : étant donnés un but et un ensemble d'agents, générer une composition d'agents satisfaisant le but. Dans les travaux [14, 15, 18], les buts sont des conditions sur le comportement de la composition et ils peuvent être exprimés dans le langage *EaGLe*. Dans cette approche, les auteurs s'intéressent à la composition de services Web, problème qui consiste à combiner et coordonner les services disponibles en un processus complexe satisfaisant la condition donnée. Selon [3, 4, 5], les buts sont des automates finis à entrée/sortie, ils sont définis comme un agent : le but est l'agent complexe obtenu par la composition, il représente le comportement global souhaité pour la communauté d'agents. Dans notre cas, l'agent but partage la même définition que les agents de la communauté (voir Section 3.5). La composition consiste à combiner et coordonner les services disponibles en un processus complexe qui simule l'automate fini de l'agent but. Cette composition s'effectue en fabriquant un processus *médiateur* qui va organiser la communauté d'agents via des échanges de messages. Dans cet article, nous automatisons la composition définie dans la seconde approche avec pour objectif de vérifier l'une des quatre relations suivantes entre la composition obtenue et l'agent but : inclusion de traces, équivalence de traces, simulation et bisimulation. Cela nous amène au problème de décision suivant :

> $CP(\approx)$ : étant donnés un ensemble fini d'actions $\Sigma$, deux ensembles finis de canaux $\Pi$ et $\Pi' \subseteq \Pi$, des automates finis $\mathcal{A} = (S_{\mathcal{A}}, \Delta_{\mathcal{A}}, s^{in}_{\mathcal{A}})$, $\mathcal{B}_1 = (S_{\mathcal{B}_1}, \Delta_{\mathcal{B}_1}, s^{in}_{\mathcal{B}_1})$, …, $\mathcal{B}_n = (S_{\mathcal{B}_n}, \Delta_{\mathcal{B}_n}, s^{in}_{\mathcal{B}_n})$ définis sur $(\{!,?\} \times \Pi) \cup \Sigma$, déterminer s'il existe un automate fini $\mathcal{C} = (S_{\mathcal{C}}, \Delta_{\mathcal{C}}, s^{in}_{\mathcal{C}})$ défini sur $\{!,?\} \times \Pi$ tel que $Exec(\mathcal{A}) \approx Exec(\mathcal{B}_1 \otimes \ldots \otimes \mathcal{B}_n \otimes \mathcal{C})$ $(\{!,?\} \times \Pi')$.

Dans $CP(\approx)$, l'automate fini $\mathcal{A}$ joue le rôle du but et les automates $\mathcal{B}_1$, …, $\mathcal{B}_n$ jouent le rôle des agents disponibles. L'automate fini $\mathcal{C}$, quant à lui joue le rôle de médiateur entre les agents ; il va donc coordonner les agents de sorte à obtenir, en faisant abstraction des communications internes (les communications sur les canaux de $\Pi'$), un processus équivalent au but. En réalité le problème $CP(\approx)$ représente quatre problèmes différents suivant que $\approx$ est égale à $\subseteq_{tr}$, $\equiv_{tr}$, $\leq_{si}$ ou $\longleftrightarrow_{bi}$.
Intuitivement, vouloir que le comportement du but soit inclus dans celui des agents $\mathcal{B}_1$, …, $\mathcal{B}_n$ coordonnés par $\mathcal{C}$ signifie que toute séquence d'actions effectuée par le but peut également être effectuée par les agents coordonnés par le médiateur (en permettant éventuellement aux

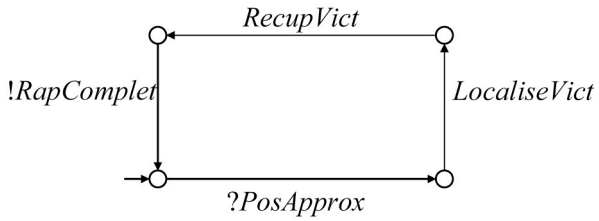



FIG. 5 – Agent *But*

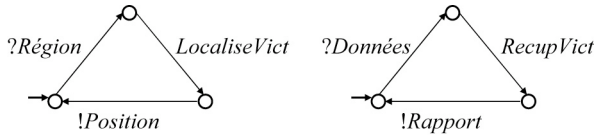



FIG. 6 – De gauche à droite agent *Pilote* et agent *Sauveteur*

agents d'effectuer des actions supplémentaires). Dans le cas où l'équivalence de trace est considérée, les agents coordonnés par le médiateur ne peuvent pas effectuer de séquence d'actions que le but ne peut pas effectuer. Par exemple, cela est utile pour garantir la sécurité du système en interdisant des séquences d'actions potentiellement dangeureuses. Les notions de simulation et de bisimulation, à la différence des notions d'inclusion de traces et d'équivalence de traces, se basent sur les états des systèmes à comparer. Par exemple, deux états sont similaires si on y accède avec la même séquence (passé identique) d'actions et que les séquences à partir de ces deux états sont identiques (futur identique). En particulier, la bisimulation garantit que les systèmes ont, pour un passé identique, les mêmes actions disponibles. Ainsi, ce sont des notions plus restrictives que l'inclusion et l'équivalence de traces.
Considérons notre scénario. Soit $\mathcal{A}$, $\mathcal{B}_1$, $\mathcal{B}_2$, les automates finis représentés respectivement dans les Figures 5 et 6. Dans ces Figures, *LocaliseVict* et *RecupVict* sont des actions et *Position*, *RapComplet*, *PosApprox*, *Données*, *Région* et *Rapport* sont des canaux.

Dans ce cas, l'automate fini $\mathcal{C}$ de la Figure 7 est tel que $Exec(\mathcal{A}) \longleftrightarrow_{bi} Exec(\mathcal{B}_1 \otimes \mathcal{B}_2 \otimes \mathcal{C})$ $(\{!, ?\} \times \{$*Position, Données, Rapport, Région*$\})$.

# 5 Complexité et algorithmes

Dans cette section, nous étudions la complexité des quatre problèmes de la composition définis dans la section précédente. Nous donnons les algorithmes effectifs pour l'inclusion de traces, la

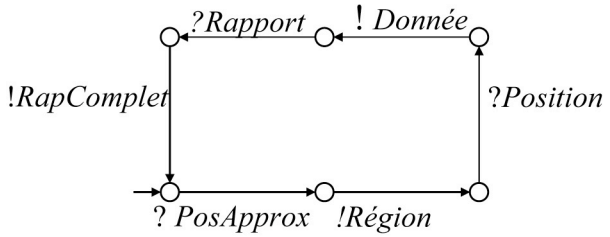



FIG. 7 – Agent *Médiateur*

simulation et la bisimulation.

## 5.1 Réductions en espace logarithmique

Dans un premier temps, nous présentons des réductions en espace logarithmique pour les problèmes $CP(\subseteq_{tr})$, $CP(\equiv_{tr})$, $CP(\leq_{si})$ et $CP(\longleftrightarrow_{bi})$.

**Théorème 1** *Le problème* $CP(\subseteq_{tr})$ *est EXPSPACE-difficile.*

**Preuve:** Cette preuve se base sur une réduction du problème d'universalité des langages réguliers avec carré. Soit $\Sigma$ un alphabet fini et $\Sigma^*$ l'ensemble des mots de longueur finie sur $\Sigma$. Nous noterons le mot vide (de longueur 0) par $\epsilon$. Une expression régulière avec carré $\alpha$ définie sur $\Sigma$ est construite comme suit

$\alpha := \epsilon \mid a \mid \alpha'.\alpha'' \mid \alpha' + \alpha'' \mid \alpha^+ \mid \alpha^2$, où $a \in \Sigma$.

Chaque expression $\alpha$ définit un langage rationnel sur $\Sigma$. Nous notons par $L(\alpha)$ le langage défini par $\alpha$. Ainsi, par exemple, $L(\alpha^2) = L(\alpha).L(\alpha)$.

Considérons le problème de décision suivant :
$ER$ : étant donnés un alphabet fini $\Sigma$ et une expression régulière avec carré $\alpha$ définie sur $\Sigma$, déterminer si $L(\alpha)$=$\Sigma^*$.
Sachant que $ER$ est ***EXPSPACE***-difficile [11], il suffit de réduire $ER$ à $CP(\subseteq_{tr})$ afin de montrer que $CP(\subseteq_{tr})$ est ***EXPSPACE***-difficile.
Étant donnée un alphabet fini $\Sigma$ et une expression régulière avec carré $\alpha$ définie sur $\Sigma$, nous devons déterminer si $L(\alpha)$=$\Sigma^*$. L'instance $\rho(\Sigma, \alpha)$ de $CP(\subseteq_{tr})$ que nous construisons est donnée par l'ensemble fini d'actions $\Sigma'$, l'ensemble fini de canaux $\Pi_\alpha$, les automates finis $\mathcal{A} = (S_\mathcal{A}, \delta_\mathcal{A}, s^{in}_\mathcal{A})$ et $\mathcal{B}_\alpha = (S_\mathcal{B}, \delta_\mathcal{B}, s^{in}_\mathcal{B})$ sur $(\{!, ?\} \times \Pi_\alpha) \cup \Sigma'$ et $\Pi'_\alpha \subseteq \Pi$ sont définis par
– $\Sigma' = \Sigma$,
– $\Pi_\alpha$ est construit par induction à partir de $\alpha$,

– $\Pi'_\alpha = \Pi_\alpha$,
– $\mathcal{A}$ est l'automate fini de la Figure 8,
– $\mathcal{B}_\alpha$ est construit par induction à partir de $\alpha$.

L'automate $\mathcal{A}$ est construit de telle sorte que le langage reconnu par $\mathcal{A}$, noté $L(\mathcal{A})$ soit égal à $\Sigma^*$. Maintenant, nous donnons la construction de $\mathcal{B}_\alpha$ et $\Pi_\alpha$ en fonction de $\alpha$. La construction de $\mathcal{B}_\alpha$ est telle que $L(\mathcal{B}_\alpha) = L(\alpha)$.

– Cas où $\alpha = \epsilon$, $\Pi_\epsilon = \{\pi\}$ et $\mathcal{B}_\epsilon$ est l'automate fini de la Figure 9.
– Cas où $\alpha = a$, $\Pi_a = \emptyset$ et $\mathcal{B}_a$ est l'automate fini de la Figure 10.
– Cas où $\alpha = \gamma.\sigma$, $\Pi_{\gamma.\sigma} = \Pi_\gamma \cup \Pi_\sigma \cup \{\pi\}$ où $\pi$ est un nouveau canal et $\mathcal{B}_{\gamma.\sigma}$ est l'automate fini de la Figure 11.
– Cas où $\alpha = \gamma + \sigma$, $\Pi_{\gamma+\sigma} = \Pi_\gamma \cup \Pi_\sigma \cup \{\pi\}$ où $\pi$ est un nouveau canal et $\mathcal{B}_{\gamma+\sigma}$ est l'automate fini de la Figure 12.
– Cas où $\alpha = \gamma^+$, $\Pi_{\gamma^+} = \Pi_\gamma \cup \{\pi\}$ où $\pi$ est un nouveau canal et $\mathcal{B}_{\gamma^+}$ est l'automate fini de la Figure 13.
– Cas où $\alpha = \gamma^2$, $\Pi_{\gamma^2} = \Pi_\gamma \cup \{\pi, \pi'\}$ où $\pi$ et $\pi'$ sont de nouveaux canaux et $\mathcal{B}_{\gamma^2}$ est l'automate fini de la Figure 14.

Cela complète la construction. Il est évident que $\rho$ peut être calculé en espace logarithmique. De plus, $L(\alpha)$=$\Sigma^*$ ssi il existe un automate fini $\mathcal{C} = (S_\mathcal{C}, \Delta_\mathcal{C}, s^{in}_\mathcal{C})$ défini sur $\{!, ?\} \times \Pi_\alpha$ tel que $Exec(\mathcal{A}) \subseteq_{tr} Exec(\mathcal{B}_\alpha \otimes \mathcal{C})$ $(\{!, ?\} \times \Pi'_\alpha)$. Ainsi, $CP(\subseteq_{tr})$ est ***EXPSPACE***-difficile. ⊣

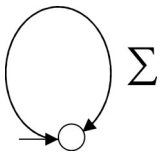



FIG. 8 –

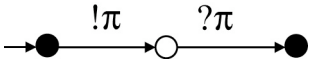



FIG. 9 –

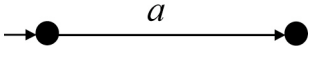



FIG. 10 –

**Théorème 2** *Le problème $CP(\equiv_{tr})$ est EXPSPACE-difficile.*

**Preuve:** Cette preuve se base sur la réduction du problème d'équivalence de trace de réseaux de Petri saufs. Soit $\Sigma$ un ensemble fini d'actions. Un réseau de Petri sauf défini sur $\Sigma$ est une structure de la forme $\mathcal{N} = (P, T, F, l)$ où $P$ est un ensemble fini de places, $T$ est un ensemble fini de transitions, $F \subseteq (P \times T) \cup (T \times P)$ est une relation et $l : T \to \Sigma$ est une fonction Pour toute transition $t \in T$, soit ${}^\bullet t$ l'ensemble de toutes les places $p \in P$ telles que $p\ F\ t$ et $t^\bullet$ l'ensemble de toutes les places $p \in P$ telles que $t\ F\ p$. Soit $Exec(\mathcal{N})$=$(S', \Delta', u^{in'})$, l'automate fini défini sur $\Sigma$ tel que $S' = 2^P$, $\Delta'$ est une fonction définie par : $v' \in \Delta'(u', a)$ ssi il existe $t \in T$ tel que $l(t) = a$, ${}^\bullet t \subseteq u'$ et $v' = (u' \setminus {}^\bullet t) \cup t^\bullet$, $u^{in'} = \emptyset$. Considérons le problème de décision suivant :

$RP(\equiv_{tr})$ : étant donnés un ensemble fini d'actions $\Sigma$ et des réseaux de Petri saufs $\mathcal{N} = (P_\mathcal{N}, T_\mathcal{N}, F_\mathcal{N}, l_\mathcal{N})$, $\mathcal{O} = (P_\mathcal{O}, T_\mathcal{O}, F_\mathcal{O}, l_\mathcal{O})$ définis sur $\Sigma$, Déterminer si $Exec(\mathcal{N}) \equiv_{tr} Exec(\mathcal{O})$ $(\emptyset)$.

Sachant que $RP(\equiv_{tr})$ est ***EXPSPACE***-difficile [9], il suffit de réduire $RP(\equiv_{tr})$ à $CP(\equiv_{tr})$ afin de montrer que $CP(\equiv_{tr})$ est ***EXPSPACE***-difficile. Etant donnés un ensemble fini d'actions $\Sigma$ et des réseaux de Petri saufs $\mathcal{N} = (P_\mathcal{N}, T_\mathcal{N}, F_\mathcal{N}, l_\mathcal{N})$, $\mathcal{O} = (P_\mathcal{O}, T_\mathcal{O}, F_\mathcal{O}, l_\mathcal{O})$ définis sur $\Sigma$, nous devons déterminer si $Exec(\mathcal{N}) \equiv_{tr} Exec(\mathcal{O})$ $(\emptyset)$. L'instance $\rho(\Sigma, \mathcal{N}, \mathcal{O})$ de $CP(\equiv_{tr})$ que nous construisons est donnée par l'ensemble fini d'actions $\Sigma^e$, l'ensemble fini de canaux $\Pi$, les automates finis $\mathcal{A} = (S_\mathcal{A}, \Delta_\mathcal{A}, s^{in}_\mathcal{A})$ et $\mathcal{B} = (S_\mathcal{B}, \Delta_\mathcal{B}, s^{in}_\mathcal{B})$ définis sur $(\{!, ?\} \times \Pi) \cup \Sigma^e$ et $\Pi' \subseteq \Pi$ définis par

– $\Sigma^e = \Sigma \cup \{a^e_1, a^e_2, a^e_3, a^e_4\}$ où $a^e_1$, $a^e_2$, $a^e_3$ et $a^e_4$ sont de nouvelles actions,
– $\Pi = \Pi' = P_\mathcal{N} \cup P_\mathcal{O}$,
– $\mathcal{A}$ est l'automate fini de la Figure 15,

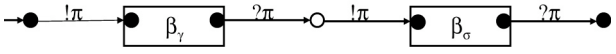



FIG. 11 –

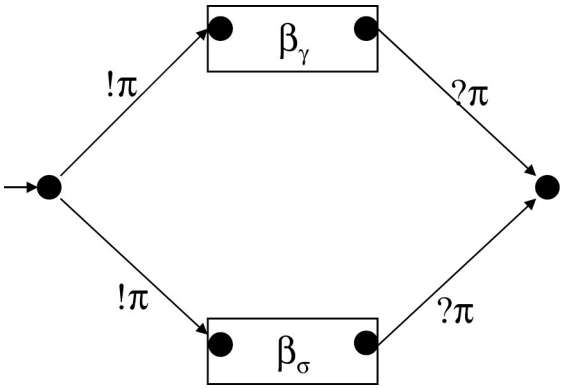



FIG. 12 –

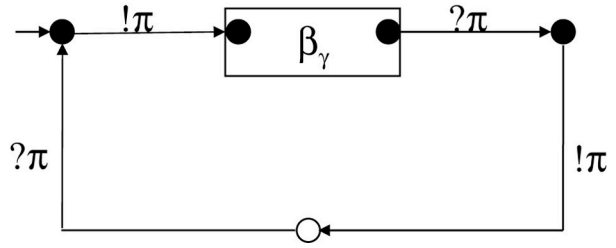



FIG. 13 –

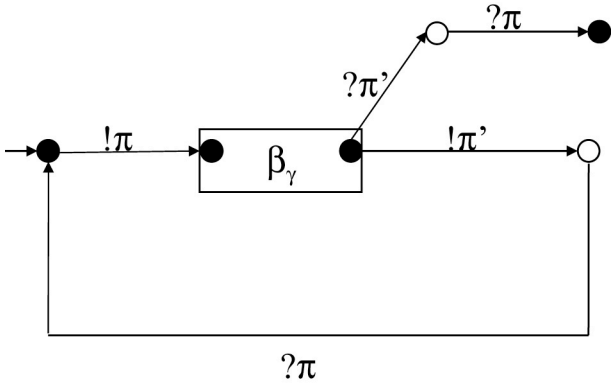



FIG. 14 –

– $\mathcal{B}$ est l'automate fini de la Figure 16.

Pour comprendre comment les parties en forme de fleur de $\mathcal{A}$ et $\mathcal{B}$ sont définies, le lecteur est invité à consulter la Figure 17. Cette figure montre que le tir de la transition $a$ vide les places $P_1, \ldots, P_{|\bullet t|}$ qui sont en entrée de la transition et remplit les places $P'_1, \ldots, P'_{|t\bullet|}$ qui sont en sortie de cette transition. Cette transition est reproduite par les automates $\mathcal{A}$ et $\mathcal{B}$ par la séquence d'actions suivantes : lecture sur les canaux $P_1, \ldots, P_{|\bullet t|}$, correspondant aux places en entrée de la transition, ce qui a pour effet de vider ces canaux, puis la transition $a$, suivi d'écriture sur les canaux $P'_1, \ldots, P'_{|t\bullet|}$, correspondant aux places en sortie de la transition, ce qui a pour effet de remplir ces canaux. Ainsi le contenu des canaux reflète correctement le contenu des places du réseaux. Les séquences de transitions des automates $\mathcal{A}$ et $\mathcal{B}$ contenant les actions $a_1^e$, $a_2^e$, $a_3^e$ ou $a_4^e$ servent à garantir qu'aucun médiateur ne peut venir interférer avec la simulation du réseau de Petri. Cela complète la construction. Il est évident que $\rho$ peut être calculé en un espace logarithmique. De plus, $Exec(\mathcal{N}) \equiv_{tr} Exec(\mathcal{O})$ $(\emptyset)$ ssi il existe un automate fini $\mathcal{C} = (S_\mathcal{C}, \Delta_\mathcal{C}, s_\mathcal{C}^{in})$ défini sur $\{!, ?\} \times \Pi$ tel que $Exec(\mathcal{A}) \equiv_{tr} Exec(\mathcal{B} \otimes \mathcal{C})$ $(\{!, ?\} \times \Pi')$. Ainsi, $CP(\equiv_{tr})$ est *EXPSPACE*-difficile.

⊣

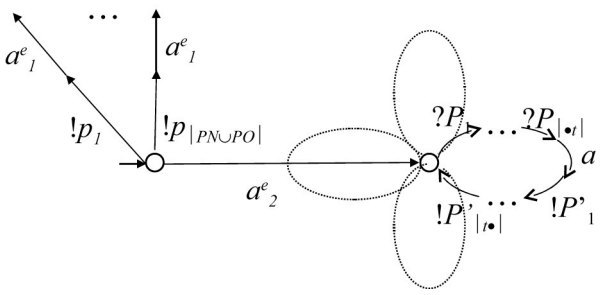



FIG. 15 –

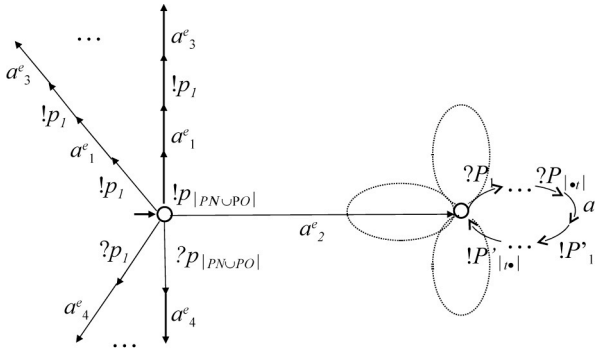



FIG. 16 –

**Théorème 3** *Le problème $CP(\leq_{si})$ est EXPTIME-difficile.*

**Preuve:** Considérons le problème de décision suivant :

Étant donnés un ensemble fini d'actions $\Sigma$ et des automates finis déterministes $\mathcal{A}, \mathcal{B}_1, \ldots, \mathcal{B}_n$ définis sur $\Sigma$, déterminer si $\mathcal{A} \leq_{si} \mathcal{B}_1 \otimes \ldots \otimes \mathcal{B}_n(\emptyset)$.

Ce problème est *EXPTIME*-difficile [13]. Il est clair qu'il existe une réduction en espace logarithmique de ce problème au problème $CP(\leq_{si})$. Si on considère les mêmes automates avec $\Pi = \emptyset$ alors $\mathcal{A} \leq_{si} \mathcal{B}_1 \otimes \ldots \otimes \mathcal{B}_n(\emptyset)$ ssi il existe $\mathcal{C} = (\{s_\mathcal{C}^{in}\}, \Delta_\mathcal{C}, s_\mathcal{C}^{in})$ défini sur $\{!, ?\} \times \emptyset$ tel que $Exec(\mathcal{A}) \leq_{si} Exec(\mathcal{B}_1 \otimes \ldots \otimes \mathcal{B}_n \otimes \mathcal{C})$ $(\{!, ?\} \times \emptyset)$. ⊣

**Théorème 4** *Le problème $CP(\longleftrightarrow_{bi})$ est EXPTIME-difficile.*

**Preuve:** On utilise la réduction de la preuve du théorème 2. Sachant que le problème de la bisimulation entre deux réseaux de Petri saufs est *EXPTIME*-difficile [9], on obtient que $CP(\longleftrightarrow_{bi})$ est également *EXPTIME*-difficile. ⊣

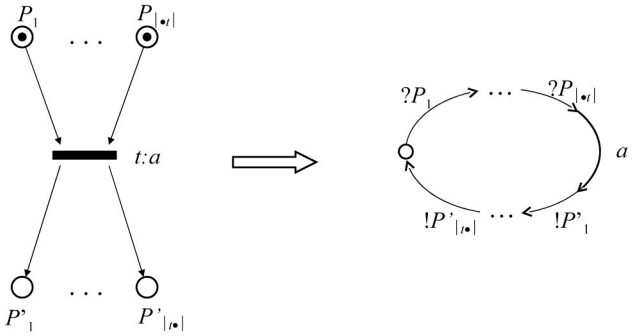



FIG. 17 –

### 5.2 Classes de complexité

Dans ce qui suit, nous présentons des algorithmes, basés sur les techniques de la synthèse de contrôleurs, pour résoudre les problèmes de la composition $CP(\subseteq_{tr})$, $CP(\equiv_{tr})$, $CP(\leq_{si})$ et $CP(\longleftrightarrow_{bi})$. Nous donnons également leur complexité.

Soit $\Sigma$ un ensemble fini d'actions et $\Pi$ un ensemble fini de canaux. Il est commode de considérer un ensemble fini $\Pi^\circ$ de canaux tel que $(\Sigma \cup \Pi) \cap \Pi^\circ = \emptyset$ et $Card(\Pi) = Card(\Pi^\circ)$ et d'utiliser une bijection $\pi \mapsto \pi^\circ$ de $\Pi$ vers $\Pi^\circ$. Par $L^\circ$, nous faisons référence à un automate fini $\mathcal{A}' = (S', \Delta', s^{in'})$ défini sur $\{!,?\} \times \Pi^\circ$ tel que $S' = \{0\}$, $\Delta'$ est la fonction définie par $\Delta'(0, !\pi^\circ) = \{0\}$ et $\Delta'(0, ?\pi^\circ) = \{0\}$, et $s^{in'} = 0$. Soit $\mathcal{A} = (S, \Delta, s^{in})$ un automate fini défini sur $(\{!,?\} \times (\Pi \cup \Pi^\circ)) \cup \Sigma$. Par $Del^\circ(\mathcal{A})$, nous notons l'automate fini $\mathcal{A}' = (S', \Delta', s^{in'})$ défini sur $(\{!,?\} \times \Pi) \cup \Sigma$ tel que $S' = S$, $\Delta'$ est la fonction définie par

– $\Delta'(s, !\pi) = \Delta(s, !\pi) \cup \Delta(s, !\pi^\circ)$ et $\Delta'(s, ?\pi) = \Delta(s, ?\pi) \cup \Delta(s, ?\pi^\circ)$,
– $\Delta'(s, a) = \Delta(s, a)$,

et $s^{in'} = s^{in}$. Par $Exec^\circ(\mathcal{A})$, nous notons l'automate fini $\mathcal{A}' = (S', \Delta', s^{in'})$ défini sur $(\{!,?\} \times (\Pi \cup \Pi^\circ)) \cup \Sigma$ de taille exponentielle tel que $S' = S \times 2^\Pi$, $\Delta'$ est la fonction définie par

– $(t, Q) \in \Delta'((s, P), !\pi)$ ssi $t \in \Delta(s, !\pi)$, $Q = P \cup \{\pi\}$, $\pi \notin P$ et $(t, Q) \in \Delta'((s, P), ?\pi)$ ssi $t \in \Delta(s, ?\pi)$, $\pi \in P$, $Q = P \setminus \{\pi\}$,
– $(t, Q) \in \Delta'((s, P), !\pi^\circ)$ ssi $t \in \Delta(s, !\pi^\circ)$, $Q = P \cup \{\pi\}$, $\pi \notin P$ et $(t, Q) \in \Delta'((s, P), ?\pi^\circ)$ ssi $t \in \Delta(s, ?\pi^\circ)$, $\pi \in P$, $Q = P \setminus \{\pi\}$,
– $(t, Q) \in \Delta'((s, P), a)$ ssi $t \in \Delta(s, a)$, $Q = P$,

et $s^{in'} = (s^{in}, \emptyset)$. Observons que nous pouvons construire $Exec^\circ(\mathcal{A})$ en un temps exponentiel.

Considérons le problème de décision suivant :

$SC(\approx)$ : étant donnés un ensemble fini d'actions $\Sigma$, deux ensembles finis de canaux $\Pi$ et $\Pi' \subseteq \Pi$, des automates finis $\mathcal{A}$, $\mathcal{B}_1, \ldots, \mathcal{B}_n$, déterminer s'il existe un automate fini $\mathcal{C}$ défini sur $(\{!,?\} \times (\Pi \cup \Pi^\circ)) \cup \Sigma$ qui boucle sur les actions dans $\Sigma \cup (\{!,?\} \times \Pi)$ et tel que $Exec(\mathcal{A}) \approx Del^\circ(Exec^\circ(\mathcal{B}_1 \otimes \ldots \otimes \mathcal{B}_n \otimes L^\circ) \times C)(\{!,?\} \times \Pi')$

Nous avons montré dans l'article [2] que le problème de synthèse de contrôleur $SC(\approx)$ et le problème de la composition $CP(\approx)$ sont équivalents.

**Théorème 5** *Le problème $CP(\subseteq_{tr})$ est dans EXPSPACE.*

**Preuve:** Considérons le problème de décision suivant :

$VP(\subseteq_{tr})$ : étant donnés un ensemble fini d'actions $\Sigma$, deux ensembles finis de canaux $\Pi$ et $\Pi' \subseteq \Pi$, des automates finis $\mathcal{A}$, $\mathcal{B}_1, \ldots, \mathcal{B}_n$, déterminer si $Exec(\mathcal{A}) \subseteq_{tr} Del^\circ(Exec^\circ(\mathcal{B}_1 \otimes \ldots \otimes \mathcal{B}_n \otimes L^\circ))(\{!,?\} \times \Pi')$

Il est facile de vérifier que le problème $VP(\subseteq_{tr})$ est équivalent au problème $SC(\subseteq_{tr})$. Considérons l'algorithme suivant :

1. Construire l'automate fini $\mathcal{A}'$=$Exec(\mathcal{A})$ ;
2. Construire l'automate fini $\mathcal{B}$=$Exec^\circ(\mathcal{B}_1 \otimes \ldots \otimes \mathcal{B}_n \otimes L^\circ)$ ;
3. Construire l'automate fini $\mathcal{B}' = Del^\circ(\mathcal{B})$
4. Si $\mathcal{A}' \leq_{si} \mathcal{B}'(\{!,?\} \times \Pi')$ alors retourner $vrai$ sinon retourner $faux$ ;

Sachant que la taille de $\mathcal{A}'$ est exponentielle par rapport à la taille de $\Pi$, que la taille de $\mathcal{B}$ est exponentielle par rapport à la taille de $\Pi$ et de $\mathcal{B}_1, \ldots, \mathcal{B}_n$, que la taille de $\mathcal{B}'$ est constante par rapport à la taille de $\mathcal{B}$ et qu'un espace polynomial est suffisant pour déterminer si $\mathcal{A}' \leq_{si} \mathcal{B}'(\{!,?\} \times \Pi')$ [6], il en résulte que $CP(\subseteq_{tr})$ est dans *EXPSPACE.* $\dashv$

**Théorème 6** *Le problème $CP(\leq_{si})$ est dans EXPTIME.*

**Preuve:** L'argument est le même que celui donné pour $CP(\subseteq_{tr})$. Rappelons que le problème suivant est dans $P$ [8] : étant donnés deux ensembles finis d'actions $\Sigma$ et $\Sigma' \subseteq \Sigma$ et deux automates finis $\mathcal{A}$ et $\mathcal{B}$ définis sur $\Sigma$, de plus déterminer si $\mathcal{A} \leq_{si} \mathcal{B}(\Sigma')$. $\dashv$

**Théorème 7** *Le problème $CP(\longleftrightarrow_{bi})$ est dans 2-EXPTIME.*

**Preuve:** Soit $\Sigma$ un ensemble fini d'actions. Cet ensemble est partitionné en un ensemble d'actions observables $\Sigma_{ob}$ et un ensemble d'actions non observables $\Sigma_{nob}$. L'ensemble $\Sigma$ est aussi partitionné en un ensemble d'actions contôlables $\Sigma_{ct}$ et un ensemble d'actions non contrôlables $\Sigma_{nct}$. Soit $\mathcal{C}$ un automate fini défini sur $\Sigma$. Les contraintes d'observabilité $(O)$ et de contrôlabilité $(C)$ pour l'automate $\mathcal{C}$ sont définies comme suit :

$(O)$ pour tout état $s$ dans $\mathcal{C}$ et pour toute action non observable $a \in \Sigma_{nob}$, s'il existe une transition de $s$ étiquetée par $a$ alors cette transition boucle sur $s$.

$(C)$ pour tout état $s$ dans $\mathcal{C}$ et pour toute action non contrôlable $a \in \Sigma_{nct}$, il existe une transition de $s$ étiquetée par $a$.

Considérons le problème de décision suivant :

Étant donnés des ensembles finis d'actions $\Sigma$ et $\Sigma'$, des automates finis $\mathcal{A}$ et $\mathcal{G}$ définis sur $\Sigma$ et des contraintes $(O)$ et $(C)$, déterminer s'il existe un automate fini $\mathcal{C}$ qui satisfait $(O)$ et $(C)$ et tel que $\mathcal{A} \longleftrightarrow_{bi} Del^{\circ}(\mathcal{G} \times C)(\Sigma')$

Ce problème peut être résolu en temps exponentiel [2]. Considérons les entrées du problème $CP(\longleftrightarrow_{bi})$. Soit un ensemble fini d'actions $\Sigma$, des ensembles finis de canaux $\Pi$ et $\Pi' \subseteq \Pi$ et des automates finis $\mathcal{A}, \mathcal{B}_1, \ldots, \mathcal{B}_n$ définis sur $(\{!, ?\} \times \Pi) \cup \Sigma$. Considérons l'algorithme suivant :

1. Construire l'automate fini $\mathcal{A}'$=$Exec(\mathcal{A})$ ;
2. Construire l'automate fini $\mathcal{B}$=$Exec^{\circ}(\mathcal{B}_1 \otimes \ldots \otimes \mathcal{B}_n \otimes L^{\circ})$ ;
3. Utiliser une procédure pour déterminer s'il existe $\mathcal{C}$ défini sur $(\{!, ?\} \times (\Pi \cup \Pi^{\circ})) \cup \Sigma$ qui boucle sur $(\{!, ?\} \times \Pi) \cup \Sigma$ et tel que $\mathcal{A} \longleftrightarrow_{bi} Del^{\circ}(\mathcal{B} \times C)(\{!, ?\} \times \Pi')$ ;

Cet algorithme est dans *2-EXPTIME*. ⊣

Concernant le problème $CP(\equiv_{tr})$, nous n'avons pas d'algorithme pour le résoudre. Dans la littérature il existe des résultats concernant la difficulté algorithmique d'un problème similaire [17]. Il existe également des algorithme pour résoudre le problème $SC(\equiv_{tr})$ sans la fonction $Del^{\circ}$ [16]. Cependant, à notre connaissance il n'existe pas d'algorithme pour résoudre le problème $SC(\equiv_{tr})$.

## 6 Conclusion et travaux futurs

Nous avons présenté un modèle dans lequel les agents sont représentés par des automates finis qui peuvent envoyer et recevoir des messages.

| Problème | Difficulté algorithmique | Complexité |
|---|---|---|
| $CP(\subseteq_{tr})$ | *EXPSPACE* | *EXPSPACE* |
| $CP(\equiv_{tr})$ | *EXPSPACE* | ? |
| $CP(\leq_{si})$ | *EXPTIME* | *EXPTIME* |
| $CP(\longleftrightarrow_{bi})$ | *EXPTIME* | *2-EXPTIME* |

TAB. 1 – Tableau récapitulatif

Nous avons défini le problème de la composition pour quatre types d'équivalence, l'inclusion de traces, l'équivalence de traces, la simulation et la bisimulation. Les résultats de la complexité des problèmes sont récapitulés dans le tableau 1. Concernant le problème de la composition en considérant l'inclusion de trace et la simulation, nous avons donné la difficulté algorithmique ainsi que la classe de complexité du problème. Plus précisément, nous avons montré que le problème de la composition est *EXPSPACE*-complet pour l'inclusion de traces et *EXPTIME*-complet pour la simulation. Nous avons également prouvé que le problème de la composition est *EXPSPACE*-difficile pour l'équivalence de trace et *EXPTIME*-difficile pour la bisimulation. De plus nous avons proposé un algorithme qui résout le problème de la composition pour la bisimulation en un temps doublement exponentiel par rapport à la taille du problème. Les algorithmes proposés dans cet article nécessitent tous de calculer explicitement la composition de l'ensemble des agents, ce qui est une étape particulièrement coûteuse lorsque le nombre d'agents est élevé. Cependant, les réductions en espace logarithmique proposées dans la section 5.1 montrent qu'il n'existe pas d'algorithme de moindre complexité permettant de résoudre le problème, du moins dans le cas de la simulation et de l'inclusion de traces. Plusieurs approches sont possibles en vue d'obtenir une complexité moindre. La première consiste à proposer une procédure qui sélectionne les agents pertinents par rapport au but à atteindre, soit via une heuristique, soit en donnant a priori des spécifications sur les agents. Une autre approche consiste à restreindre les comportements des agents. D'un autre côté, le modèle proposé dans cet article est déjà restrictif au niveau des communications. Il est possible de l'enrichir en autorisant des canaux de communication qui contiennent des files de messages de longueurs bornées, ce qui reviendrait dans notre formalisme à remplacer un canal par un nombre suffisant de canaux pour simuler la file de messages. Nous savons par contre qu'il n'est pas possible de considérer des files de messages de longueur non bornées ; en effet, la preuve du

théorème 2 serait valide pour des réseaux de Petri généraux pour lequel le problème est indécidable. Pour enrichir le processus de communication, il conviendrait alors de considérer par exemple des messages qui sont décrits par des termes de logique du premier ordre qui indiqueraient le type du message échangé.

## 7 Remerciements

Pour la préparation de cet article nous avons bénéficié d'un soutien financier dans le cadre des projets COPS (ANR) et ROSACE (RTRA STAE).

## Références

[1] A. Arnold, A. Vincent, I. Walukiewicz, 'Games for synthesis of controllers with partial observation'. Theoretical computer science **303** (2003) 7–34.

[2] P. Balbiani, F. Cheikh, G. Feuillade, 'Composition of interactive Web services based on controller synthesis'. In : *2nd International Workshop on Web Service Composition and Adaptation*. IEEE (2008) 521–528.

[3] D. Berardi, D. Calvanese, G. De Giacomo, R. Hull, M. Mecella, 'Automated composition of transition-based semantic Web services with messaging'. In : *Proceedings of the 31st International Conference on Very Large Data Bases.* VLDB Endowment (2005) 613–624.

[4] D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, M. Mecella, 'Automatic composition of e-services that export their behavior'. In : *Service-Oriented Computing — ICSOC 2003.* Springer (2003) 43–58.

[5] F. Cheikh, G. De Giacomo, M. Mecella, 'Automatic Web services composition in trust-aware communities'. In : *Proceedings of the 3rd ACM Workshop on Secure Web Services.* Association for Computing Machinery (2006) 43–52.

[6] M. Garey, D. Johnson, 'Computers and Intractability : A Guide to the Theory of NP-Completeness'. H. Freeman (1979).

[7] M. Huhns, 'Agents as Web services'. IEEE Internet Computing **6** (2002) 93–95.

[8] H. Hüttel, S. Shukla. 'On the Complexity of Deciding Behavioural Equivalences and Preorders, A Survey'. Rapport dans la série BRICS (1996).

[9] L. Jategaonkar, A. Meyer, 'Deciding true concurrency equivalences on safe, finite nets'. Theoretical computer science **154** (1996) 107–143.

[10] T. Melliti, S. Haddad, A. Suna, A. El Fallah Seghrouchni, '$Web - MASI$ : multi-agent systems interoperability using a Web services based approach'. In : *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology 2005.* IEEE (2005) 739–742.

[11] A. Meyer, L. Stockmeyer, 'The equivalence problem for regular expressions with squaring requires exponential space'. In : *Proceedings of the Annual Symposium on Switching and Automata Theory.* IEEE (1972) 125–129.

[12] N. Milanovic, M. Malek, 'Current solutions for Web service composition'. IEEE Internet Computing **8** (2004) 51–59.

[13] A. Muscholl, I. Walukiewicz, 'A Lower bound on Web services composition'. In : *Proceedings of the International Conference on Foundations of Software Science and Computation Structures.* Springer (2007) 274–286.

[14] M. Pistore, F. Barbon, P. Bertoli, D. Shaparau, P. Traverso, 'Planning and monitoring Web service composition'. In : *Artificial Intelligence : Methodology, Systems, and Applications.* Springer (2004) 106–115.

[15] M. Pistore, A. Marconi, P. Bertoli, P. Traverso, 'Automated composition of Web services by planning at the knowledge level'. In : *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence.* International Joint Conferences on Artificial Intelligence (2005) 1252–1259.

[16] P. Ramadge, Wonham, 'The control of descrete event systems'. Proceedings of the IEEE **77** (1989) 81–98.

[17] J. Tsitsiklis, 'On the control of discrete event dynamical systems'. Mathematics of Control, Signals and Systems **2**(1989) 95–107.

[18] P. Traverso, M. Pistore, 'Automated composition of semantic Web services into executable processes'. In : *The Semantic Web — ISWC 2004.* Springer (2004) 380–394.

# Raisonnements sur la permission de dire

P. Balbiani* H. van Ditmarsch† P. Seban*
balbiani@irit.fr hans@cs.otago.ac.nz seban@irit.fr

* Université de Toulouse - CNRS et UPS
Institut de recherche en informatique de Toulouse

† Computing Science, University of Aberdeen, UK
& Computer Science, University of Otago, New Zealand

**Abstract:**
Our goal is to formalize what "having the permission to say" means. We adapt the dynamic logic of permission considered by van der Meyden in the particular case where atomic actions are public and truthful announcements. Hence, we extend the logic of public announcements introduced by Plaza, that enables us to express the ways agents update their knowledge after public and truthful announcements, by introducing an operator of permission. We present the syntax and the semantics of our logic and we address the issues of axiomatizing and deciding the set of its valid formulas.

**Keywords:** Epistemic reasoning, deontic reasoning, public announcements, modal logic, axiomatisation and decidability

**Résumé :**
Notre objectif est de formaliser ce que signifie "avoir le droit de dire". Nous adaptons la logique dynamique de la permission considérée par van der Meyden dans le cas particulier où les actions atomiques sont des annonces publiques vraies. Nous étendons ainsi la logique des annonces publiques introduite par Plaza, qui nous permet d'exprimer comment les agents mettent à jour leurs connaissances après une annonce publique vraie, en introduisant un opérateur de permission. Nous présentons la syntaxe et la sémantique de notre logique et nous répondons aux problèmes de l'axiomatisation et de la décidabilité de l'ensemble de ses formules valides.

**Mots-clés :** Raisonnement épistémique, raisonnement déontique, annonces publiques, logique modale, axiomatisation et décidabilité

## 1 Introduction

Quatre joueurs jouent à la belote. Les trente-deux cartes du paquet sont distribuées, chaque joueur commence donc avec huit cartes qu'il connaît, et tout ceci fait partie de la connaissance commune des quatre joueurs. Nous pouvons représenter cette situation en utilisant la logique épistémique (voir [8] pour plus de détails) : l'ensemble des atomes propositionnels de notre langage est $\{(VC)_i | V \in \{7, 8, 9, 10, V, D, R, A\}, C \in \{\clubsuit, \heartsuit, \diamondsuit, \spadesuit\}, i \in \{1, 2, 3, 4\}\}$. Un modèle du jeu de belote a quatre relations (d'équivalence) épistémiques, un monde correspond à une distribution, et on dira que $\mathcal{M}, s \models (10\spadesuit)_3$ si le joueur 3 a le dix de pique dans la distribution $s$. Nous pouvons alors exprimer des phrases telles que "le joueur 2 sait que le joueur 3 a la dame de coeur" ($K_2(D\heartsuit)_3$). Que se passe-t-il lorsque le jeu commence ? Le fait que le joueur 1 joue le 7 de carreau peut être vu comme l'annonce publique que le joueur 1 a le 7 de carreau. Mais il peut arriver qu'il lui soit impossible de jouer cette carte bien qu'il l'ait dans son jeu, car la règle du jeu ne le lui autorise pas. Quels sont alors les annonces qui sont permises dans ce jeu, en fonction de la situation donnée ?

Pour formaliser la permission d'agir, van der Meyden propose la logique dynamique de la permission [16]. Il s'agit essentiellement d'une adaptation de la logique propositionnelle dynamique ($PDL$) [9] pour laquelle les modèles contiennent un ensemble $\mathcal{P} \subseteq S \times \mathbf{Act} \times S$ qui relie chaque action de **Act** à l'ensemble des transitions entre états de $S$ qui lui sont permises. Autrement dit, $\mathcal{P}$ est un ensemble de transitions permises. La syntaxe de ce langage contient les constructions suivantes :

$$\varphi ::= \bot | p | \neg\varphi | \varphi \vee \varphi | [\alpha]\varphi | \Diamond(\alpha, \varphi)$$

où $[\alpha]\varphi$ signifie "*après chaque exécution de l'action $\alpha$, $\varphi$ est vrai*", et $\Diamond(\alpha, \varphi)$ signifie "*il existe une exécution de l'action $\alpha$ qui est permise et après laquelle $\varphi$ devient vraie*".

Notre objectif est de formaliser le concept de "permission de dire". Pour cela, nous choisissons d'adapter la sémantique considérée par [16] dans le cas particulier où les actions sont remplacées par des annonces publiques. L'équivalent de $\alpha$ dans la formule $\Diamond(\alpha, \varphi)$ de van der Meyden devient pour nous l'annonce $\psi!$ de telle sorte que $\Diamond(\psi!, \varphi)$ signifie maintenant 'il est permis d'annoncer $\psi$, et ensuite $\varphi$ devient vraie'. La primitive de notre langage est en fait un peu différente car il nous faut prendre en compte la possibilité qu'une séquence d'annonces soit faite, mais nous serons en mesure de définir un tel $\Diamond(\psi!, \varphi)$ comme un cas particulier.

La logique des annonces publiques ($PAL$), qui

est une extension, proposée par Plaza dans [13], de la logique épistémique, pemet d'exprimer la façon dont des agents mettent à jour leurs connaissances après l'annonce publique d'une proposition vraie. Nous pouvons par exemple écrire dans ce langage $\langle\psi\rangle\varphi$, ce qui signifie qu'après l'annonce publique de la proposition vraie $\psi$, $\varphi$ devient vraie. Cette logique a été largement étudiée (voir [7]) et étendue (voir par exemple [2] et [3]).

Pour parler de "la permission de dire", nous allons étendre $PAL$ en introduisant un opérateur $P$ de permission, $P\varphi$ exprimant le fait qu'il est permis de dire $\varphi$. En fait, pour pouvoir définir la mise à jour de la permission après une annonce publique et obtenir la complétude de notre logique, nous définirons un opérateur binaire plus général $P(\psi,\varphi)$ exprimant le fait que "après l'annonce de $\psi$ il est permis d'annoncer $\varphi$". Nous présenterons d'abord la syntaxe et la sémantique de notre logique, puis nous prouverons quelques résultats techinques, en particulier la conmplétude de l'axiomatique que nous définissons et la décidabilité du problème de satisfiabilité.

# 2 Logique des annonces pulbiques et de la permission

## 2.1 Syntaxe de $\mathcal{L}_{ppal}$

Le langage $\mathcal{L}_{ppal}$, sur un ensemble dénombrable $N$ d'agents et un ensemble dénombrable $\Theta$ d'atomes propositionnels, est défini de la façon suivante :

$$\varphi ::= \bot|p|\neg\varphi|\psi\vee\varphi|K_i\varphi|[\psi]\varphi|P(\psi,\varphi)$$

où $i\in N$ et $p\in\Theta$.

Notre lecture de $K_i\varphi$ est "*l'agent i sait que* $\varphi$ *est vraie*" alors que nous interprétons $[\psi]\varphi$ par "*après toute annonce publique possible de* $\psi$*, il est vrai que* $\varphi$". $P(\psi,\varphi)$ se lit "*après que* $\psi$ *ait été publiquement annoncé, il est permis de dire* $\varphi$". Les abréviations classiques pour les autres opérateurs booléens sont utilisées. Par ailleurs, nous définissons un opérateur $\langle\psi\rangle$ par $\langle\psi\rangle\varphi = \neg[\psi]\neg\varphi$.

En outre, appelons $\mathcal{L}_{pel}$ le fragment de notre langage ne contenant pas d'opérateur d'annonces et $\mathcal{L}_{el}$ celui restreint aux opérateurs épistémiques et booléens.

Définissons d'autres notations qui s'avèreront utiles :

- $F(\psi,\varphi) := \neg P(\psi,\varphi)$
- $P\varphi := P(\top,\varphi)$
- $F\varphi := F(\top,\varphi)$

$P\varphi$ signifie "*il est permis de dire* $\varphi$", $F(\psi,\varphi)$ signifie "*après que* $\psi$ *ait été annoncé publiquement, il est interdit de dire* $\varphi$" et $F\varphi$ signifie "*il est interdit de dire* $\varphi$".

Enfin, définissons le degré $deg$ d'une formule.

**Définition 1** (Degré)**.** *Le degré d'une formule* $\varphi\in\mathcal{L}_{ppal}$ *est défini par récurrence sur la forme de* $\varphi$ *de la façon suivante :* $deg(p)=0$*;* $deg(\bot)=0$*;* $deg(\neg\psi)=deg(\psi)$*;* $deg(\psi_1\vee\varphi_2)=max(deg(\psi_1),deg(\psi_2))$*;* $deg(K_i\psi)=deg(\psi)$*;* $deg([\psi]\varphi)=deg(\psi)+deg(\varphi)$*;* $deg(P(\psi,\varphi))=deg(\psi)+deg(\varphi)+1$.

Notons que pour toute formule $\varphi\in\mathcal{L}_{ppal}$, $deg(\varphi)=0$ ssi $\varphi$ ne contient aucune occurence de $P$.

## 2.2 Sémantique

Les modèles de notre logique sont des structures de la forme $\mathcal{M}=(S,\{\sim_i\}_{i\in N},V,\mathcal{P})$ où $S$ est un ensemble non-vide d'états, $\sim_i$ est une relation d'équivalence sur les états de $S$, $V$ est une fonction attribuant un sous ensemble de $S$ à chaque atome propositionnel $p\in\Theta$ et $\mathcal{P}\subseteq S\times 2^S\times 2^S$. Le modèle $\mathcal{M}_\psi$ défini ci-dessous est la mise à jour d'un modèle $\mathcal{M}$ après l'annonce publique de $\psi$ :

**Définition 2** (Modèle restreint par une annonce)**.** *Pour tout modèle* $\mathcal{M}$ *et toute formule* $\psi\in\mathcal{L}_{ppal}$*, on définit la* restriction $\mathcal{M}_\psi = (S_\psi,\sim_i^\psi,V_\psi,\mathcal{P}_\psi)$ *où :*
– $S_\psi = [\![\psi]\!]_{\mathcal{M}} = \{s\in S|\mathcal{M},s\models\psi\}$
– *pour tout* $p\in\Theta$, $V_\psi(p)=V(p)\cap S_\psi$
– *pour tout* $i$, $\sim_i^\psi = \sim_i\cap(S_\psi\times S_\psi)$
– $\mathcal{P}_\psi = \{(s,S',S'')\in\mathcal{P}\mid s\in S_\psi,\ S'\subseteq S_\psi, S''\subseteq S_\psi\}$

Il est maintenant possible de définir la relation de satisfaisabilité $\models$.

**Définition 3** (Relation de satisfaisabilité)**.** *Soit* $\mathcal{M}$ *un modèle et* $s$ *un état* $S$*. La relation de satisfaisabilité* $\mathcal{M},s\models\varphi$ *est définie par récurrence sur la structure de* $\varphi$ *de la façon suivante :*
$\mathcal{M},s\models p$ *ssi* $s\in V(p)$
$\mathcal{M},s\not\models\bot$
$\mathcal{M},s\models\neg\psi$ *ssi* $\mathcal{M},s\not\models\psi$

$\mathcal{M}, s \models \psi_1 \vee \psi_2$ *ssi* ($\mathcal{M}, s \models \psi_1$ *ou* $\mathcal{M}, s \models \psi_2$)
$\mathcal{M}, s \models K_i\psi$ *ssi pour tout* $t \sim_i s$, $\mathcal{M}, t \models \psi$
$\mathcal{M}, s \models [\psi]\chi$ *ssi* ($\mathcal{M}, s \models \psi \Rightarrow \mathcal{M}_\psi, s \models \chi$)
$\mathcal{M}, s \models P(\psi, \chi)$ *ssi* $(s, [\![\psi]\!]_\mathcal{M}, [\![\langle\psi\rangle\chi]\!]_\mathcal{M}) \in \mathcal{P}$

*Si deux états* $s, t \in S$, *sont liés par la relation d'équivalence* $\sim_i$, *cela signifie que pour l'agent* $i$, $s$ *et* $t$ *sont indistingables.*

*L'appartenance de* $(s, S', S'')$ *à* $\mathcal{P}$ *peut être interprétée de la façon suivante : dans l'état* $s$, *toute annonce qui restreint l'ensemble de tous les mondes possibles à* $S'$ *le fera de telle sorte que toute annonce ultérieure qui restreindrait l'ensemble de tous les mondes possibles à* $S''$ *sera permise.*

*Pour tout* $\varphi \in \mathcal{L}_{ppal}$, *l'on note* $\mathcal{M} \models \varphi$ *ssi pour tout* $s \in S$, $\mathcal{M}, s \models \varphi$ *et* $\models \varphi$ *ssi pour tout modèle* $\mathcal{M}$ *l'on a* $\mathcal{M} \models \varphi$.

Il faut noter que nous n'imposons pas que $S'$ et $S''$ correspondent à des formules du langage pour que $(s, S', S'')$ soit dans $\mathcal{P}$. Cette sémantique est donc plus générale que ce que l'on entend intuitivement lorsque l'on dit "avoir le droit de dire". En effet, si $S'$ ou $S''$ ne correspondent pas à la restriction de $S$ par l'annonce publique d'une formule donnée, alors le fait que $(s, S', S'')$ soit dans $\mathcal{P}$ ne correspond pas au fait qu'une annonce est permise.

Notons que $\models \langle\psi\rangle\varphi \leftrightarrow \psi \wedge [\psi]\varphi$. Voyons quelques autres formules qui sont valides dans notre logique :

**Proposition 4.**
- $\models [\psi]p \longleftrightarrow (\psi \rightarrow p)$
- $\models [\psi]\bot \longleftrightarrow \neg\psi$
- $\models [\psi]\neg\varphi \longleftrightarrow (\psi \rightarrow \neg[\psi]\varphi)$
- $\models [\psi](\varphi_1 \vee \varphi_2) \longleftrightarrow ([\psi]\varphi_1 \vee [\psi]\varphi_2)$
- $\models [\psi]K_i\varphi \longleftrightarrow (\psi \rightarrow K_i[\psi]\varphi)$
- $\models [\psi_1][\psi_2]\varphi \longleftrightarrow [\langle\psi_1\rangle\psi_2]\varphi$
- $\models [\psi]P(\psi', \varphi) \longleftrightarrow (\psi \rightarrow P(\langle\psi\rangle\psi', \varphi))$

Ces équivalences appellent une explication, regardons en détail la première. Elle dit que $p$ est vraie après toute annonce possible de $\psi$ ssi si $\psi$ est vraie (et donc peut être annoncée) alors $p$ est vraie. En d'autres termes, une annonce ne peut pas changer la valuation.

**Démo**. La preuve de la validité des six premières formules est facile et laissée au lecteur (voir [13] pour plus de détails). Démontrons la validité de la septième. Pour tout $\mathcal{M}$, tout $x \in S$ et tout $\psi, \psi', \varphi \in \mathcal{L}_{ppal}$ on a :

($\Rightarrow$) $\mathcal{M}, x \models [\psi]P(\psi', \varphi) \wedge \psi$ seulement si ($x \in [\![\psi]\!]_\mathcal{M}$ et $(x, [\![\psi']\!]_{\mathcal{M}_\psi}, [\![\langle\psi'\rangle\varphi]\!]_{\mathcal{M}_\psi}) \in \mathcal{P}_\psi$) seulement si $(x, [\![\langle\psi\rangle\psi']\!]_\mathcal{M}, [\![\langle\psi\rangle\langle\psi'\rangle\varphi]\!]_\mathcal{M}) \in \mathcal{P}$ seulement si $(x, [\![\langle\psi\rangle\psi']\!]_\mathcal{M}, [\![\langle\langle\psi\rangle\psi'\rangle\varphi]\!]_\mathcal{M}) \in \mathcal{P}$ seulement si $\mathcal{M}, x \models P(\langle\psi\rangle\psi', \varphi)$.

($\Leftarrow$) $\mathcal{M}, x \models (\psi \rightarrow P(\langle\psi\rangle\psi', \varphi)$ seulement si ($x \in [\![\psi]\!]_\mathcal{M}$ implique $(x, [\![\langle\psi\rangle\psi']\!]_\mathcal{M}, [\![\langle\langle\psi\rangle\psi'\rangle\varphi]\!]_\mathcal{M}) \in \mathcal{P}$) seulement si ($x \in [\![\psi]\!]_\mathcal{M}$ implique $(x, [\![\psi']\!]_{\mathcal{M}_\psi}, [\![\langle\psi'\rangle\varphi]\!]_{\mathcal{M}_\psi}) \in \mathcal{P}_\psi$) seulement si $\mathcal{M}, x \models [\psi]P(\psi', \varphi)$. □

Regardons de plus près la septième formule de la proposition 4. Elle dit la chose suivante : "*après l'annonce possible de* $\psi$, *nous obtenons qu'après l'annonce de* $\psi'$ *il est permis de dire* $\varphi$ *ssi ou bien* $\psi$ *ne peut pas être annoncé, ou bien après l'annonce de* $\psi$ *suivie de l'annonce de* $\psi'$ *il est permis de dire* $\varphi$". Ca ressemble à une tautologie, mais cela souligne précisément que la définition de la mise à jour d'un modèle après une annonce vraie correspond à ce que l'on attend.

Une autre propriété intéressante que revêt notre sémantique est que, sans conditions supplémentaires, la proposition suivante est vraie :

**Proposition 5.** *Pour tout modèle* $\mathcal{M}$ *et toutes formules* $\psi, \psi', \varphi, \varphi' \in \mathcal{L}_{ppal}$ *nous avons que si* $\mathcal{M} \models (\psi \leftrightarrow \psi') \wedge (\langle\psi\rangle\varphi \leftrightarrow \langle\psi'\rangle\varphi')$ *alors* $\mathcal{M} \models P(\psi, \varphi) \leftrightarrow P(\psi', \varphi')$

Cela vient de la définition de la restriction $\mathcal{P}_\psi$ et correspond aux intuitions suivantes :

1. après deux annonces équivalentes les mêmes formules sont permises d'être dites, et
2. si deux propositions sont équivalentes, elles sont permises d'être dites de la même manière.

Plus précisément :

**Démo**. Pour tous $\psi, \psi', \varphi, \varphi' \in \mathcal{L}_{ppal}$, si $\mathcal{M} \models (\psi \leftrightarrow \psi')$ et $\mathcal{M} \models \langle\psi\rangle\varphi \leftrightarrow \langle\psi'\rangle\varphi'$, alors $[\![\psi]\!]_\mathcal{M} = [\![\psi']\!]_\mathcal{M}$ et $[\![\langle\psi\rangle\varphi]\!]_\mathcal{M} = [\![\langle\psi'\rangle\varphi']\!]_\mathcal{M}$. Cela implique que pour tout $s \in S$, $(s, [\![\psi]\!]_\mathcal{M}, [\![\langle\psi\rangle\varphi]\!]_\mathcal{M}) = (s, [\![\psi']\!]_\mathcal{M}, [\![\langle\psi'\rangle\varphi']\!]_\mathcal{M})$. □

Les propositions 4 et 5 nous permettent de définir la traduction suivante $tr : \mathcal{L}_{ppal} \rightarrow \mathcal{L}_{pel}$ :

**Définition 6** (la traduction $tr$)**.**
*On définit $tr(\varphi)$ par récurrence sur la complexité de $\varphi$ de la façon suivante :*
- $tr(p) = p$
- $tr(\bot) = \bot$
- $tr(\neg\varphi) = \neg tr(\varphi)$
- $tr(\psi \vee \varphi) = tr(\psi) \vee tr(\varphi)$
- $tr(K_i\varphi) = K_i tr(\varphi)$
- $tr(P(\psi, \varphi)) = P(tr(\psi), tr(\varphi))$
- $tr([\psi]p) = tr(\psi) \rightarrow p$
- $tr([\psi]\bot) = \neg tr(\psi)$
- $tr([\psi]\neg\varphi) = tr(\psi) \rightarrow \neg tr([\psi]\varphi)$
- $tr([\psi](\varphi_1 \vee \varphi_2)) = tr([\psi]\varphi_1) \vee tr([\psi]\varphi_2)$
- $tr([\psi]K_i\varphi) = tr(\psi) \rightarrow K_i tr([\psi]\varphi)$
- $tr([\psi_1][\psi_2]\varphi) = tr([\langle\psi_1\rangle\psi_2]\varphi)$
- $tr([\psi_1]P(\psi_2, \varphi)) = tr(\psi_1) \rightarrow P(tr(\langle\psi_1\rangle\psi_2), tr(\varphi))$

D'après les propositions 4 et 5, on obtient

**Proposition 7.** *Pour tout $\varphi \in \mathcal{L}_{ppal}$, $\models \varphi \leftrightarrow tr(\varphi)$.*

Finalement, présentons une propriété très utile du degré d'une formule :

**Proposition 8.** *Pour toute formule $\varphi \in \mathcal{L}_{ppal}$, $deg(tr(\varphi)) = deg(\varphi)$.*

Les preuves des propositions 7 et 8 peuvent être faites aisément par récurrence sur la complexité de $\varphi$.

# 3 Axiomatique

## 3.1 Correction et complétude

Soit $PPAL$ la plus petite logique normale sur notre langage qui satisfasse les axiomes inclus dans la table 1 et qui soit close pour la règle d'inférence suivante :
- De $(\psi \leftrightarrow \psi') \wedge (\langle\psi\rangle\varphi \leftrightarrow \langle\psi'\rangle\varphi')$ déduire $P(\psi, \varphi) \leftrightarrow P(\psi', \varphi')$

**Proposition 9.** *$PPAL$ est une aximoatique correcte pour la classe des modèles de Kripke pour lesquels $\sim_i$ est une relation d'équivalence pour tout $i \in N$.*

**Démo**. Par les propositions 4 et 5. □

On obtient en particulier la proposition suivante :

**Proposition 10.** *Pour tout $\varphi \in \mathcal{L}_{ppal}$, $\vdash_{PPAL} \varphi \leftrightarrow tr(\varphi)$.*

Pour démontrer le résultat de complétude, définissons le modèle canonique pour $PPAL$ :

**Définition 11** (Modèle canonique)**.** *Le modèle canonique $\mathcal{M}^c = (S^c, \sim_i^c, V^c, \mathcal{P}^c)$ est défini de la façon suivante : $S^c$ est l'ensemble de tous les ensembles $\vdash_{PPAL}$-maximaux consistants ; pour tout $p \in \Theta$, $V^c(p) = \{x \in S^c \mid p \in x\}$; $x \sim_i^c y$ ssi $K_i x = K_i y$, où $K_i x = \{\varphi | K_i\varphi \in x\}$ et $\mathcal{P}^c = \{(x, S', S'') : \exists P(\psi, \varphi) \in x |\ S' = \{y \in S^c : \psi \in y\}\ ,\ S'' = \{y \in S^c : \langle\psi\rangle\varphi \in y\}\}$.*

Dans le modèle canonique, un état est donc un ensemble de formules. Le lien entre le fait qu'une formule $\varphi$ soit dans un ensemble $x$ et le fait que $\mathcal{M}^c, x \models \varphi$ est explicité par la proposition suivante :

**Proposition 12** (Truth Lemma pour $\mathcal{L}_{pel}$)**.** *Pour tout $\varphi \in \mathcal{L}_{pel}$,*

$\Pi(\varphi)$ : *pour tout $x \in S^c, \mathcal{M}^c, x \models \varphi$ ssi $\varphi \in x$*

**Démo**. Démontrons la proposition par récurrence sur le degré de $\varphi$.

**cas initial** Si $deg(\varphi) = 0$ alors $\varphi \in \mathcal{L}_{el}$ et $\Pi(\varphi)$ est un résultat connu (voir [4] ou [8] pour plus de détails).

**pas de récurrence** Soit $k \in \mathbb{N}$, supposons que $\Pi(\psi)$ est vrai pour toute formule $\psi \in \mathcal{L}_{pel}$ telle que $deg(\psi) \leqslant k$. Soit $\varphi$ telle que $deg(\varphi) \leq k+1$. Raisonnons maintenant par récurrence sur la structure de $\varphi$.

- $\varphi = p; \bot; \neg\psi; \varphi_1 \vee \varphi_2; K_i\psi$ **:** Voir la preuve du truth lemma pour $\mathcal{L}_{el}$ dans [4] ou [8].
- $\varphi = P(\psi, \chi)$ **:**
  ($\Leftarrow$) Si $P(\psi, \chi) \in x$ alors $(x, S', S'') \in \mathcal{P}^c$ où $S' = \{y \in S^c : \psi \in y\}$ et $S'' = \{y \in S^c : \langle\psi\rangle\chi \in y\}$. Notons que $deg(P(\psi, \chi)) \leqslant k+1$ implique $deg(\psi) + deg(\chi) \leqslant k$, et en particulier $deg(\psi) \leqslant k$ et $deg(\langle\psi\rangle\chi) \leqslant k$. Nous obtenons donc que $S' = \{y \in S^c : \mathcal{M}^c, y \models \psi\}$ (par HR) et $S'' = \{y \in S^c : \mathcal{M}^c, y \models \langle\psi\rangle\chi\}$. En effet, du fait que $\vdash_{PPAL} \langle\psi\rangle\chi \leftrightarrow tr(\langle\psi\rangle\chi)$ et $deg(\langle\psi\rangle\chi) = deg(tr(\langle\psi\rangle\chi))$, pour tout $y \in S^c$, $\langle\psi\rangle\chi \in y$ ssi $tr(\langle\psi\rangle\chi) \in y$ ssi $\mathcal{M}^c, y \models tr(\langle\psi\rangle\chi)$ (par HR) ssi $\mathcal{M}^c, y \models \langle\psi\rangle\chi$. Finalement, on obtient que $(x, [\![\psi]\!]_{\mathcal{M}^c}, [\![\langle\psi\rangle\chi]\!]_{\mathcal{M}^c}) \in \mathcal{P}^c$ et donc $\mathcal{M}^c, x \models P(\psi, \chi)$.
  ($\Rightarrow$) Si $\mathcal{M}^c, x \models P(\psi, \chi)$ alors $(x, [\![\psi]\!]_{\mathcal{M}^c}, [\![\langle\psi\rangle\chi]\!]_{\mathcal{M}^c}) \in \mathcal{P}^c$. Nous sommes toujours dans le cas où $deg(\psi) \leqslant k$ et

TAB. 1 – L'axiomatique $PPAL$

| | |
|---|---|
| $K_i\varphi \rightarrow \varphi$ | vérité |
| $K_i\varphi \rightarrow K_iK_i\varphi$ | introspection positive |
| $\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$ | introspection negative |
| $[\psi]p \longleftrightarrow (\psi \rightarrow p)$ | stabilité pour les atomes |
| $[\psi]\bot \longleftrightarrow \neg\psi$ | ann. et faux |
| $[\psi]\neg\varphi \longleftrightarrow (\psi \rightarrow \neg[\psi]\varphi)$ | ann. et négation |
| $[\psi](\varphi_1 \vee \varphi_2) \leftrightarrow ([\psi]\varphi_1 \vee [\psi]\varphi_2)$ | ann. et disjonction |
| $[\psi]K_i\varphi \longleftrightarrow (\psi \rightarrow K_i[\psi]\varphi)$ | ann. et connaissance |
| $[\psi_1][\psi_2]\varphi \longleftrightarrow [\langle\psi\rangle_1\psi_2]\varphi$ | composition d'annonces |
| $[\psi]P(\psi',\varphi) \longleftrightarrow (\psi \rightarrow P(\langle\psi\rangle\psi',\varphi))$ | ann. et permission |

$deg(\langle\psi\rangle\chi) \leqslant k$. On obtient donc que $[\![\psi]\!]_{\mathcal{M}^c} = \{y \in S^c : \psi \in y\}$ (par HR) et $[\![\langle\psi\rangle\chi]\!]_{\mathcal{M}^c} = \{y \in S^c : \langle\psi\rangle\chi \in y\}$ (argument identique). Donc, il existe $P(\psi',\chi') \in x$ tel que $\{y \in S^c : \psi \in y\} = \{y \in S^c : \psi' \in y\}$ et $\{y \in S^c : \langle\psi\rangle\chi \in y\} = \{y \in S^c : \langle\psi'\rangle\chi' \in y\}$. Par conséquent, $(\psi \leftrightarrow \psi') \wedge (\langle\psi\rangle\varphi \leftrightarrow \langle\psi'\rangle\varphi') \in PPAL$ et $P(\psi,\chi) \leftrightarrow P(\psi',\chi') \in PPAL$. Comme $P(\psi',\chi') \in x$, alors $P(\psi,\chi) \in x$. □

**Proposition 13.** *$PPAL$ est correcte et complète pour la classe de modèles pour lesquels $\sim_i$ est une relation d'équivalence pour tout $i \in N$.*

**Démo**. La correction a été démontrée dans la proposition 9. Nous montrons la complétude pour cette classe de modèles à l'aide de la proposition 12 . En effet, pour toute formule $\varphi \in \mathcal{L}_{ppal}$ : $\models \varphi \Rightarrow \models tr(\varphi) \Rightarrow \mathcal{M}^c \models tr(\varphi) \Rightarrow \vdash_{PPAL} tr(\varphi) \Rightarrow \vdash_{PPAL} \varphi$. □

### 3.2 Axiomes supplémentaires

Il est possible d'exiger que soient vérifiées les propriétés suivantes, pour toutes formules $\varphi, \psi \in \mathcal{L}_{ppal}$ :

1. $\vdash P(\psi,\varphi) \rightarrow \psi$ : S'il est permis de dire quelquechose après l'annonce de $\psi$, alors $\psi$ est vrai (et peut donc être annoncé).
2. $\vdash P(\psi,\varphi) \rightarrow [\psi]\varphi$ : si après avoir annoncé $\psi$ il est permis de dire $\varphi$, alors $\varphi$ est vrai après l'annonce de $\psi$.
3. $\vdash P(\psi,\varphi) \rightarrow \langle\psi\rangle\varphi$

Notons que 1 & 2 est équivalent à 3 (car $\vdash \psi \wedge [\psi]\varphi \leftrightarrow \langle\psi\rangle\varphi$). Que se passe-t-il donc si nous ajoutons ces axiomes à $PPAL$ ? La réponse est que l'axiomatique que nous obtenons est correcte et complète pour la classe de modèles restreinte correspondante.

**Proposition 14.**

**PPAL + (1)** *est correcte et complète pour la classe de modèles pour lesquels $(s,S',S'') \in \mathcal{P} \Rightarrow s \in S'$*

**PPAL + (2)** *est correcte et complète pour la classe de modèles pour lesquels $(s,S',S'') \in \mathcal{P} \Rightarrow (s \in S' \Rightarrow s \in S'')$.*

**Démo**. A l'aide d'un argument similaire à celui de la preuve de la proposition 13, il suffit de prouver que ces axiomes supplémentaires sont corrects dans la classe de modèles correspondante, et que chaque nouveau modèle canonique appartient à la classe correspondante. Cette preuve est simple est laissée au soin du lecteur. □

## 4 Decidabilité et complexité

Nous allons démontrer dans cette section (théorème 23) que $PPAL$ est décidable en démontrant une propriété dite *du modèle fini*. Pour cela, nous allons utiliser une filtration.

**Définition 15** (Ensemble fermé)**.** *Un ensemble $X \subseteq \mathcal{L}_{pel}$ est dit* fermé *s'il est fermé pour la sous-formule et s'il vérifie $P(\psi,\varphi) \in X \Rightarrow tr(\langle\psi\rangle\varphi) \in X$.*

**Définition 16** (Filtration)**.** *Soit* $\mathcal{M} = (S, \sim_i, V, \mathcal{P})$ *un modèle et* $\Gamma$ *un ensemble fermé de formules. Nous définissons la relation* $\leftrightsquigarrow_\Gamma$ *entre états de* $S$ *de la façon suivante : pour tous* $s, t \in S$,

$s \leftrightsquigarrow_\Gamma t$ *ssi* $\forall \varphi \in \Gamma : (\mathcal{M}, s \models \varphi$ *ssi* $\mathcal{M}, t \models \varphi)$

*Notons que* $\leftrightsquigarrow_\Gamma$ *est une relation d'équivalence. Pour tout* $s \in S$*, on note* $|s|_\Gamma$ *(ou simplement* $|s|$*) la classe d'équivalence de* $s$ *pour la relation* $\leftrightsquigarrow_\Gamma$.

*On appelle* filtration de $\mathcal{M}$ suivant $\Gamma$ *(ou simplement* filtration de $\mathcal{M}$*) le modèle* $\mathcal{M}^f_\Gamma = (S^f, \sim^f_i, V^f, \mathcal{P}^f)$ *où :*
- $S^f = S/_{\leftrightsquigarrow_\Gamma}$
- $|s| \sim^f_i |t|$ *ssi pour tout* $K_i\varphi \in \Gamma$, $(\mathcal{M}, s \models K_i\varphi$ *ssi* $\mathcal{M}, t \models K_i\varphi)$
- $V^f(p) = \begin{cases} \emptyset \text{ si } p \notin \Gamma \\ V(p)/_{\leftrightsquigarrow_\Gamma} \text{ si } p \in \Gamma) \end{cases}$
- $\mathcal{P}^f = \{(|s|, S'/_{\leftrightsquigarrow_\Gamma}, S''/_{\leftrightsquigarrow_\Gamma})$: $(s, S', S'') \in \mathcal{P}$, $\leftrightsquigarrow_\Gamma(S') = S'$ *et* $\leftrightsquigarrow_\Gamma(S'') = S''\}$

Voici maintenant un lemme utile :

**Lemma 17.** *Soit* $\Gamma \subset \mathcal{L}_{pel}$ *un ensemble fermé fini . Pour tout modèle* $\mathcal{M}$*, sa filtration* $\mathcal{M}^f_\Gamma$ *contient au plus* $2^m$ *états, avec* $m = Card(\Gamma)$.

**Démo**. Soit $\mathcal{M}$ un modèle. Soit une fontion $g : S^f \to \mathcal{P}(\Gamma)$ définie par $g(|s|) = \{\psi \in \Gamma | \mathcal{M}, s \models \psi\}$. Il découle de la définition de $\leftrightsquigarrow_\Gamma$ que $g$ est bien définie et injective. On en conclut que la taille de $S^f$ est au plus $2^m$. □

Les relations épistémiques d'un modèle et leur filtration suivant un ensemble $\Gamma$ sont liées par la propriété suivante :

**Proposition 18.** *Soit* $\mathcal{M}$ *un modèle et* $\Gamma$ *un ensemble fermé de formules. Alors pour tous* $s, t \in S$*, pour tout* $\varphi \in \Gamma$ *:*
1. $s \sim_i t \Rightarrow |s| \sim^f_i |t|$.
2. $|s| \sim^f_i |t|$ *et* $K_i\varphi \in \Gamma$ *et* $\mathcal{M}, s \models K_i\varphi \Rightarrow \mathcal{M}, t \models \varphi$.

**Démo**.

1. Soient $s, t \in S$ tels que $s \sim_i t$, et soit $K_i\varphi \in \Gamma$. On a alors $\mathcal{M}, s \models K_i\varphi$ ssi pour tout $u \sim_i s$, $\mathcal{M}, u \models \varphi$ ssi pour tout $u \sim_i t$, $\mathcal{M}, u \models \varphi$ ssi $\mathcal{M}, t \models K_i\varphi$. Alors, par définition de $\sim^f_i$, on obtient que $|s| \sim^f_i |t|$.
2. Supposons la première partie de l'implication. Comme $|s| \sim^f_i |t|$, $K_i\varphi \in \Gamma$ et $\mathcal{M}, s \models K_i\varphi$ alors $\mathcal{M}, t \models K_i\varphi$. Comme $\sim_i$ est réflexive $\mathcal{M}, t \models \varphi$. □

Cette proposition est suffisante pour prouver la proposition importante qui suit :

**Proposition 19** (Lemme de Filtration)**.** *Soit* $\mathcal{M}$ *un modèle et* $\Gamma$ *un ensemble fermé de formules. Pour tout* $\varphi \in \Gamma$ *on a :*

$$(F\varphi) \quad \forall s \in S, (\mathcal{M}, s \models \varphi ssi \mathcal{M}^f_\Gamma, |s| \models \varphi).$$

**Démo**. Démontrons-le par récurrence sur le degré de $\varphi$.

**cas initial** Si $deg(\varphi) = 0$ alors $\varphi \in \mathcal{L}_{el}$ et la preuve de $(F\varphi)$ s'obtient par récurrence sur la complexité de $\varphi$ (voir [4] ou [8] pour plus de détails. Il faut noter aue $\Gamma$ est en particulier fermé pour la sous-formule).

**pas de récurrence** Soit $k \in \mathbb{N}$, supposons que $(F\psi)$ est vraie pour tout $\psi \in \mathcal{L}_{pel}$ tel que $deg(\psi) \leqslant k$. Soit $\varphi$ tel que $deg(\varphi) \leq k + 1$ et raisonnons par récurrence sur la structure de $\varphi$.

- $\varphi = p; \bot; \neg\psi; \varphi_1 \vee \varphi_2, K_i\varphi$ **:** Voir la preuve du lemme de filtration dans [4] ou [8].
- $\varphi = P(\psi, \chi)$ **:** Soit $s \in S$, on a les équivalence suivantes :
  $\mathcal{M}, s \models P(\psi, \chi)$ ssi $(s, [\![\psi]\!]_\mathcal{M}, [\![\langle\psi\rangle\chi]\!]_\mathcal{M}) \in \mathcal{P}$ ssi $(s, [\![\psi]\!]_\mathcal{M}, [\![tr(\langle\psi\rangle\chi]\!]_\mathcal{M}) \in \mathcal{P}$ ssi $(|s|, [\![\psi]\!]_\mathcal{M}/_{\leftrightsquigarrow_\Gamma}, [\![tr(\langle\psi\rangle\chi)]\!]_\mathcal{M}/_{\leftrightsquigarrow_\Gamma}) \in \mathcal{P}^f$.
  Mais, par hypothèse de récurrence, $t \in [\![\psi]\!]_\mathcal{M}$ ssi $|t| \in [\![\psi]\!]_{\mathcal{M}^f_\Gamma}$ (par $(F\psi)$) et $t \in [\![tr(\langle\psi\rangle\chi)]\!]_\mathcal{M}$ ssi $|t| \in [\![tr(\langle\psi\rangle\chi)]\!]_{\mathcal{M}^f_\Gamma}$ (par $(F\ tr(\langle\psi\rangle\chi))$ avec $tr(\langle\psi\rangle\chi) \in \Gamma$). On obtient $(|s|, [\![\psi]\!]_\mathcal{M}/_{\leftrightsquigarrow_\Gamma}, [\![tr(\langle\psi\rangle\chi)]\!]_\mathcal{M}/_{\leftrightsquigarrow_\Gamma}) \in \mathcal{P}^f$ ssi $(|s|, [\![\psi]\!]_{\mathcal{M}^f_\Gamma}, [\![tr(\langle\psi\rangle\chi)]\!]_{\mathcal{M}^f_\Gamma}) \in \mathcal{P}^f$ ce qui est vrai ssi $\mathcal{M}^f_\Gamma, |s| \models P(\psi, \chi)$ (d'après la sémantique).

  □

**Définition 20** (Fermeture)**.** *Pour tout* $\varphi \in \mathcal{L}_{pel}$*, on construit la fermeture de* $\varphi$*, notée* $Cl(\varphi)$*, par récurrence sur la structure de* $\varphi$ *:*
- $Cl(p) = \{p\}$
- $Cl(\bot) = \{\bot\}$
- $Cl(\neg\varphi) = \{\neg\varphi\} \cup Cl(\varphi)$
- $Cl(\psi \vee \varphi) = \{\psi \vee \varphi\} \cup Cl(\psi) \cup Cl(\varphi)$
- $Cl(K_i\varphi) = \{K_i\varphi\} \cup Cl(\varphi)$
- $Cl(P(\psi, \varphi)) = \{P(\psi, \varphi)\} \cup Cl(\psi) \cup Cl(\varphi) \cup Cl(tr(\langle\psi\rangle\varphi))$.

**Proposition 21.** *Pour toute formule $\varphi \in \mathcal{L}_{pel}, Cl(\varphi)$ est bien définie, et est un ensemble fini et fermé.*

**Démo**. Démontrons-le par récurrence sur le degré de $\varphi$.

**cas initial** Si $deg(\varphi) = 0$ alors $\varphi \in \mathcal{L}_{el}$ et il nous suffit de prouver que $Cl(\varphi)$ est un ensemble bien défini de formules, fini et fermé pour la sous-formule, ce qui est direct et laissé au lecteur.

**pas de récurrence** Soit $k \in \mathbb{N}$, supposons que $Cl(\psi)$ est un ensemble bien défini de formules, fini et fermé, pour toute formule $\psi$ telle que $deg(\psi) \leq k$. Soit $\varphi$ telle que $deg(\varphi) \leq k + 1$, et raisonnons par récurrence sur la structure de $\varphi$

- $\varphi = p; \bot; \neg\psi; \varphi_1 \vee \varphi_2; K_i\psi$ **:** Trivial.
- $\varphi = P(\psi, \chi)$ **:** $Cl(\psi), Cl(\chi)$ et $Cl(tr(\langle\psi\rangle\chi))$ sont bien définis, fermés et finis, alors $Cl(P(\psi, \chi))$ est un ensemble fini bien défini. Il ne reste qu'à démontrer qu'il est fermé, ce qui est direct et laissé aux soins du lecteur.

□

**Proposition 22** (Propriété du modèle fini)**.** *Soit $\varphi \in \mathcal{L}_{pel}$, si $\varphi$ est satisfaisable alors $\varphi$ est satisfaisable dans un modèle contenant au plus $2^m$ états, avec $m = Card(Cl(\varphi))$.*

**Démo**. Supposons que $\mathcal{M}$ et $s$ sont tels que $\mathcal{M}, s \models \varphi$. Soit $\Gamma = Cl(\varphi)$. Alors, par la proposition 19, $\mathcal{M}^f_\Gamma, |s| \models \varphi$. Par le lemme 17, $\mathcal{M}^f_\Gamma$ contient au plus $2^m$ états. □

**Théorème 23.** *$PPAL$ is decidable.*

**Démo**. Soit $\varphi \in \mathcal{L}_{ppal}$ une formule, la procédure suivante décide si oui ou non $\varphi$ est satisfaisable :

1. Calculer $\Phi = tr(\varphi)$
2. Calculer $\Gamma = Cl(\Phi)$
3. Pour tout modèle $\mathcal{M}$ de taille $\leq 2^{Card(\Gamma)}$ vérifier si $\exists s \in \mathcal{M}$ tel que $\mathcal{M}, s \models \Phi$

□

# 5 Etude de cas

Prenons l'exemple du jeu de belote. Après que l'on ait distribué les cartes, et après le choix d'une couleur d'atout, le premier joueur joue n'importe quelle carte de son jeu, puis c'est au tour de chacun des joueurs successivement, dans le sens des aiguilles d'une montre. Le joueur ayant joué la plus forte carte d'atout ou la plus forte carte de la même couleur que celle du premier joueur remporte la main et entame la main suivante avec n'importe quelle carte de son jeu. A l'exception du premier joueur de chaque main, il est obligatoire de suivre la couleur demandée ou, à défaut, de jouer de l'atout. De plus, lorsqu'un atout est joué, il est interdit de jouer un atout plus bas. Pour plus de détails sur les règles de la belote voir [1].

Comme nous l'avons indiqué précédemment, nous pouvons considérer l'action de jouer une carte comme l'annonce publique que la carte en question appartenait au joueur correspondant. Dans le langage que nous construisons, l'ensemble des atomes propositionnels $\Theta$ est donc $\{(VC)_i | V \in \{7, 8, 9, 10, V, D, R, A\}, C \in \{\clubsuit, \heartsuit, \diamondsuit, \spadesuit\}, i \in \{1, 2, 3, 4\}\}$. Chaque atome $(VC)_i$ a une valeur $V$, une couleur $C$ et appartient à un joueur $i$. Rappelons que $\mathcal{M}, s \models (VC)_i$ signifie que pour la distribution de cartes $s$, le joueur $i$ possède le $V$ de $C$. La couleur d'atout a été choisie avant que le jeu ne commence, et nous allons supposer ici que l'atout est trèfle. L'ensemble d'atomes est partiellement ordonné de la façon suivante, qui correspond à l'ordre des cartes dans le jeu ("$*$" peut être remplacé par n'importe quel nom de joueur).

Pour les non-atouts (i.e. pour tout $C \neq \clubsuit$) :

$$7C_* < 8C_* < 9C_* < JC_* < QC_* < KC_* < 10C_* < AC_*$$

Pour les atouts :

$$7\clubsuit_* < 8\clubsuit_* < Q\clubsuit_* < K\clubsuit_* < 10\clubsuit_* < A\clubsuit_* < 9\clubsuit_* < J\clubsuit_*$$

La formalisation complète du jeu serait trop longue et inutile pour notre propos, mais formalisons tout de même, à titre d'exemple, quelques unes des règles du jeu qui sont valides au début de chaque main. Cette condition implique que les modèles $\mathcal{M}$ que nous considérons sont des modèles pour lesquels tous les joueurs ont le même nombre de cartes dans leur main.

**1-Un joueur à la fois :**
Pour tout $\psi \in \mathcal{L}_{ppal}$, tous joueurs $i \neq j$, tous $p_i, q_j \in \Theta$, $\mathcal{M} \models P(\psi, p_i) \rightarrow \neg P(\psi, q_j)$.

**2-Chaque carte n'est jouée qu'une fois** :
Pour tout $p \in \Theta$, et tout $\psi \in \mathcal{L}_{ppal}$, $\mathcal{M} \models \neg P(p \wedge [p]\psi, p)$

**3-Obligation de suivre la couleur demandée :** Pour tous $V_1, V_2, V'$, tous $C \neq C'$, et tous $i \neq j$,
$\mathcal{M} \models (V_1C)_j \rightarrow \neg P((V_2C)_i, (V'C')_j)$

**4-Sinon obligation de jouer atout :** Pour tous $V_1, V, V'$, tous $C \neq C'$ avec $C, C' \neq \clubsuit$, et tous $i \neq j$,
$\mathcal{M} \models (V_1\clubsuit)_j \rightarrow \neg P((VC)_i, (V'C')_j)$

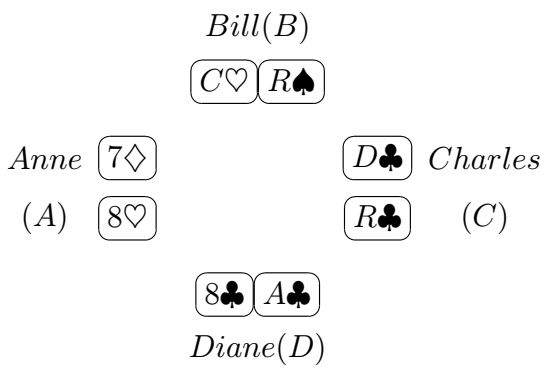
**5-Droit de dire"belote et rebelote" :** (Seule exception à la règle d'après) Pour tout $i$,
$\mathcal{M} \models (K\clubsuit)_i \wedge (Q\clubsuit)_i \wedge P(\psi, (Q\clubsuit)_i) \rightarrow P(\psi, (Q\clubsuit)_i \wedge (K\clubsuit)_i)$

**6-"On joue pas à la parlante" :** Pour tous $\psi, \varphi \in \mathcal{L}_{ppal}$ et tous $s$, $\mathcal{M}, s \models P(\psi, \varphi)$ implique que $\varphi \in \Theta$.

**7-Obligation de monter à l'atout :** pour tout $\psi \in \mathcal{L}_{ppal}$, tous $i, j$, et tous $V, V_1, V_2$ tels que $(V_1\clubsuit)_j < (V\clubsuit)_j < (V_2\clubsuit)_j$, $\mathcal{M} \models (V_2\clubsuit)_j \rightarrow \neg P(\langle\psi\rangle(V\clubsuit)_i, (V_1\clubsuit)_j)$

**etc...**

Voyons les conséquences de ces règles conditionnelles sur la permission de dire de l'état (i.e. la distribution) suivant(e) $s$, où chaque joueur possède 2 cartes.




C'est à Anne de jouer. D'après la règle du jeu, notre modèle vérifie les propriétés suvantes :

$\mathcal{M}, s \models P(8\heartsuit)_A \wedge P(7\diamondsuit)_A \wedge \neg P((8\heartsuit)_A \wedge (7\diamondsuit)_A)$ : Anne a la permission de jouer n'importe laquelle de ses deux cartes, mais pas les deux (règle (6))

$\mathcal{M}, s \models \neg P((8\heartsuit)_A, (R\spadesuit)_B)$ : Si Anne joue le $8\heartsuit$, Bill n'a pas la permission de jouer une carte d'une autre couleur (règle (3))

$\mathcal{M}, s \models P(\langle(8\heartsuit)_A\rangle(D\heartsuit)_B, (D\clubsuit)_C \wedge (R\clubsuit)_C)$ : Lorsque c'est à son tour de jouer, Charles a la permission d'annoncer qu'il possède les deux cartes qui constituent la belote. (règle (5))

$\mathcal{M}, s \models \neg P(\langle\langle(8\heartsuit_A)\rangle(D\heartsuit_B)\rangle(D\clubsuit)_C, (8\clubsuit)_D)$ : Après que Charles ait joué atout, Diane peut jouer un atout plus grand donc elle n'a pas la permission d'en jouer un plus petit que celui de Charles (rule (7)).

# 6 Comparaison et recherches ultérieures

## 6.1 Comparaison avec la littérature

Notre logique permettant de raisonner sur les annonces permises peut être vue comme une continuation des efforts entamés par van der Meyden dans [16] et Pucella et Weismann dans [14]. La notion de permission de van der Meyden est définissable à partir de notre notion. Il considère un ensemble de variables d'actions $a$ dans sa logique. Dans notre cas, ces actions sont toutes les annonces $\varphi!$. En employant cette terminologie, nous obtenons la correspondance suivante :

**Proposition 24.** *$\Diamond(\varphi!, \theta)$ est équivalent à $P(\varphi) \wedge \langle\varphi\rangle\theta$*

**Démo**. Nous pouvons voir l'annonce $\varphi!$ comme une action atomique qui relie chaque état $s \in [\![\varphi]\!]_{\mathcal{M}}$ au même état $s \in \mathcal{M}|\varphi$. Clairement cette action est *verte* dans la sémantique de van der Meyden (i.e. $\varphi!$ est une action permise) si et seulement si $\mathcal{M}, s \models P(\varphi)$. La proposition devient donc simplement une traduction entre deux langages. □

Bien entendu, le $\Diamond(\varphi!, \theta)$ de van der Meyden ayant un correspondant linguistique $Perm(\varphi)\theta$ dans [14], la relation entre $P(\varphi)$ (le nôtre) et $Perm(\varphi)$ (de Pucella est Weismann) est plus proche encore.

Dans 'Merging Frameworks for Interaction', van Benthem et al. [15, 11] proposent une logique des protocoles qui possède comme cas particulier une logique pour laquelle seuls certaines annonce vraies peuvent effectivement être annoncées. Ce qui peut ou non être annoncé est fixé par des traces ou des séquences de formules. Un ensemble de ces séquences est appelé "protocole". Leur logique semble reliée à notre proposition d'annonces permises – notez que dans notre cas les séquences d'annonces sont induites par la sémantique de $\mathcal{P}$ et la restriction $\mathcal{P}_\psi$ de cette fonction de permission dans le modèle restreint après l'annonce de $\psi$.

### 6.2 Dynamiques de la permission

La dynamique de la permission a été étudiée par Pucella et Weismann dans [14]. Ils proposent d'ajouter de nouvelles permissions en reliant les états qui satisfont une proposition $\rho_1$ aux états satisfaisant une proposition $\rho_2$, en utilisant un opérateur noté $grant(\rho_1, \rho_2)$.

Une adaptation naturelle de leur proposition, dans le contexte où les actions sont des annonces publiques, est obtenu en définissant l'opérateur modal $grant(\rho_1, \rho_2)$ de la façon suivante : pour tout modèle $\mathcal{M}$ et tout $s \in \mathcal{M}$ l'on a : $\mathcal{M}, s \models grant(\rho_1, \rho_2)\theta$ ssi $\mathcal{M}^g, s \models \theta$ où $\mathcal{M}^g$ est identique à $\mathcal{M}$ sauf pour la relation de permission qui est étendue de la façon suivante : $\mathcal{P}^g = \mathcal{P} \cup \{(s', S_1, S_2) | s' \in S_2 \subseteq S_1, \mathcal{M}_{|S_1}, s' \models \rho_1$ and $\mathcal{M}_{|S_2}, s' \models \rho_2\}$. L'opérateur $revoke$ correspondant serait alors tel que $\mathcal{P}$ devienne $\mathcal{P} \setminus \{(s', S_1, S_2) | s' \in S_2 \subseteq S_1, \mathcal{M}_{|S_1}, s' \models \rho_1$ et $\mathcal{M}_{|S_2}, s' \models \rho_2\}$.

Une telle définition est particulièrement intéressante si l'on pense qu'avoir ou non la permission de dire quelque chose dépend des conséquences d'une telle annonce.

Mais cette variante naturelle n'est pas forcément appropriée à notre étude de cas. En effet, nous aimerions être à même d'ajouter une nouvelle règle du jeu (ou d'en enlever une). Or pour ces règles, la permission de dire quelque chose dépend des caractéristiques de l'annonce elle-même, et non des états qui sont reliés par l'annonce en question.

Pour exprimer ce type de mise à jour des permissions, on peut définir les opérateurs modaux $GRANT$ and $REVOKE$.

Pour tous $\chi, \psi, \varphi \in \mathcal{L}_{ppal}, GRANT(\chi, P(\psi, \varphi))$ est tel que que pour tout modèle $\mathcal{M}$ et tout état $s \in \mathcal{M}$ l'on a : $\mathcal{M}, s \models GRANT(\chi, P(\psi, \varphi))\theta$ ssi $\mathcal{M}^G, s \models \theta$ où $\mathcal{M}^G$ est identique à $\mathcal{M}$ sauf pour la relation de permission qui est étendue de la façon suivante : $\mathcal{P}^G = \mathcal{P} \cup \{(s, [\![\psi]\!]_{\mathcal{M}}, [\![\langle\psi\rangle\varphi]\!]_{\mathcal{M}}) \mid s \in [\![\chi]\!]_{\mathcal{M}}\}$. Intuitivement, après application de $GRANT(\chi, P(\psi, \varphi))$ la proposition suivante fait partie de la règle du jeu : "si $\chi$ est vrai, alors après l'annone de $\psi$ il est permis de dire $\varphi$"

**Proposition 25.** *Pour tous $\psi, \varphi \in \mathcal{L}_{ppal}$, la formule suivante est valide :*
$\models GRANT(\chi, P(\psi, \varphi))(\chi \rightarrow P(\psi, \varphi))$

**Démo**. Cette preuve est aisée et laissée aux soins du lecteur. □

L'on peut introduire d'une façon très similaire l'opérateur $REVOKE(\chi, P(\psi, \varphi))$ : par son application, $\mathcal{P}$ devient $\mathcal{P}^R = \mathcal{P} \setminus \{(s, [\![\psi]\!]_{\mathcal{M}}, [\![\langle\psi\rangle\varphi]\!]_{\mathcal{M}}) \mid s \in [\![\chi]\!]_{\mathcal{M}}\}$.

Après introduction de l'une ou l'autre des définitions des opérateurs de la forme "grant et revoke", les problèmes de l'axiomatisation et de la décidabilité restent des problèmes ouverts.

En plus des annonces, nous pourrions étendre notre langage pour inclure des affectations, ou d'autres événements informatifs. Un exemple typique apparaît lorsque un Maire (ou l'officiel correspondant, en fonction du pays, de la législation, ...) déclare un couple "mari et femme". En "changeant le monde", pour ainsi dire, le couple est maintenant soumis à toutes les obligations et les permissions dûes au mariage. Cela ne résulte pourtant pas d'un changement de la loi, mais bien de l'application de la même loi à une nouvelle instance, résultat du changement de l'état du monde. Des changements dans le monde peuvent être obtenus par des affectations publiques, et les logiques pour les annonces publiques et les affectations publiques [6] (ou d'autres, comme souligné précédemment) peuvent donc être étendues de façon évidente à l'aide d'opérateurs de permission

### 6.3 Quantification sur la permission

Supposons que je sois en train de jouer à la belote. Je peux me demander, dans une situation donnée, s'il existe une carte qu'il m'est permis de jouer et telle que, si je la joue, il soit certain que je remporte la main. Une telle quantification sur les annonces permises peut être exprimée en introduisant l'opérateur $\Diamond(\psi, !)\theta$ exprimant le fait que $P(\psi, \varphi) \wedge [\psi]\langle\varphi\rangle\theta$ pour un certain $\varphi$. En d'autres termes, après l'annonce de $\psi$ il existe une annonce permise telle qu'après l'avoir annoncée $\theta$ devienne vrai. Il est possible d'adapter aisément les travaux de [2] et [3] pour définir des axiomatiques correctes et complètes pour ce nouveau langage, en considérant la même classe de modèles. Pour le problème de décidabilité, notons qu'en restreignant les modèles à la classe de ceux pour lesquels il n'est rien permis d'annoncer nous obtenons une logique équivalente à la logique $PAL$ dont on sait qu'elle est $PSPACE$-complète (voir [12]). Au contraire, si l'on restreint les modèles à la classe

de ceux pour lesquels toute annonce est permise l'on obtient une logique équivalente à la logique $APAL$ dont on sait qu'elle est indécidable (voir [10]). Le problème de la décidabilité lorsqu'on n'opère aucune restriction sur les modèles reste un problème ouvert.

Une autre type de quantification intéressante serait celle contenue dans $P(!, \varphi)$, qui exprime que $P(\psi, \varphi)$ est vrai pour tout $\psi$. En d'autres termes, quelles que soient les annonces ultérieures, il sera *toujours* permis de dire $\varphi$. On pourrait faire entrer les droits humains basiques dans cette catégorie, dans la mesure où rien ne pourrait les remettre en cause. Et on trouverait dans la catégorie "interdit quoi qu'il arrive" des choses du type "tu ne diras point ..."

### 6.4 Obligation

Pour définir l'obligation de dire quelque chose, on pourrait ajouter un nouvel opérateur modal $O(\psi, \varphi)$ exprimant "*après l'annonce de $\psi$, il est obligatoire de dire $\varphi$*" et une sémantique analogue à la précédente , *id est* un ensemble $\mathcal{O}$ tel que $\mathcal{M}, s \models O(\psi, \varphi)$ ssi $(s, [\![\psi]\!]_{\mathcal{M}}, [\![\langle\psi\rangle\varphi]\!]_{\mathcal{M}}) \in \mathcal{O}$. Comme précédemment, $O(\varphi)$ serait une abbréviation de $O(\top, \varphi)$.

Supposons que nous avons cette définition, comment exprimer l'obligation, pour un agent $i$, de jouer le roi ou la dame de trèfle, ce qui correspond à une situation dans laquelle il est obligatoire de jouer atout ? Notons que $O((K\clubsuit)i \vee (Q\clubsuit)_i)$ n'exprime pas cela, ni d'ailleurs $O(K\clubsuit)_i \vee O(Q\clubsuit)_i$. Il serait donc impossible d'exprimer l'obligation de dire $\varphi_1$ ou de dire $\varphi_2$. Ce n'est pas surprenant : la définition de l'obligation en logique déontique est un problème réputé plus difficile que celui de la définition de la permission (voir [5] pour plus de détails).

Nous tentons une proposition d'approche. Définissons l'annonce $\varphi!$ comme une action épistémique, et considérons alors le choix non-déterministe comme habituellement. L'obligation de dire $\varphi_1$ ou de dire$\varphi_2$ s'exprimerait alors par $O(\top, \varphi_1! \cup \varphi_2!)$. Les modèles de notre logique seraient ainsi des tuples $\mathcal{M} = (S, V, \sim_i, \mathcal{P}, \mathcal{O})$ avec $\mathcal{O} \subseteq S \times 2^S \times 2^{2^S}$. La relation de satisfaisabilité pourrait alors être définie récursivement comme précédemment, avec pour tout $\mathcal{M}$, tout $s \in S$, et tous $\psi, \varphi_1, \ldots \varphi_n \in \mathcal{L}_{ppal}$ : $\mathcal{M}, s \models O(\psi, \varphi_1! \cup \ldots \cup \varphi_n!)$ ssi $(s, [\![\psi]\!]_{\mathcal{M}}, \{[\![\langle\psi\rangle\varphi_1]\!], \ldots, [\![\langle\psi\rangle\varphi_n]\!]\}) \in \mathcal{O}$.

Nous n'avons pas considéré une telle sémantique dans la définition de la permission car nous assumions implicitement que le libre choix s'appliquait (id est que la permission de dire $\varphi_1$ ou dire $\varphi_2$ correspond à la permission de dire $\varphi_1$ et la permission de dire $\varphi_2$). Mais à vrai dire, la question de l'alternative entre la sémantique du libre choix et la sémantique du choix imposé reste ouverte (pour plus de détails sur ces notions, voir [5]).

## Références


[1] *http ://fr.wikipedia.org/wiki/Belote*.

[2] T. Ågotnes, P. Balbiani, H. van Ditmarsch, and P. Seban. Group announcement logic. *Submitted to the Journal of Applied Logic*, 2009.

[3] P. Balbiani, A. Baltag, H.P. van Ditmarsch, A. Herzig, T. Hoshi, and T.de Lima. 'knowable' as 'known after an announcement'. *Review of Symbolic Logic*, 1(03) :305–344, 2008.

[4] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[5] Jan Broersen. *Modal Action Logics for Reasoning About Reactive Systems*. PhD thesis, Vrije Universiteit Amsterdam, 2003.

[6] H.van Ditmarsch, W. van der Hoek, and B. Kooi. Dynamic epistemic logic with assignment. In *Proceedings of the fourth AAMAS*, pages 141–148. ACM, 2005.

[7] H.van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Synthese library. 2007.

[8] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[9] M. Fischer and R. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2) :194–211, 1979.

[10] T. French and H. van Ditmarsch. Undecidability for arbitrary public announcement logic. Proceedings of the seventh conference "Advances in Modal Logic", 2008.

[11] T. Hoshi. Logics of public announcements with announcement protocols. Manuscript. Philosophy Department, Stanford University, 2008.

[12] C. Lutz. Complexity and succinctness of public announcement logic. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 06)*, pages 137–144. Springer, 2006.

[13] J. Plaza. Logics of public communications. In *Proceedings of the 4th ISMIS : Poster Session Program*, pages 201–216. Oak Ridge National Laboratory, 1989.

[14] R. Pucella and V. Weissman. Reasoning about dynamic policies. *Lecture Notes in Computer Science*, 2987 :453–467, 2004.

[15] J. van Benthem, J. Gerbrandy, T. Hoshi, and E. Pacuit. Merging frameworks for interaction. *Jounal of Philosophical Logic*, forthcoming.

[16] R. van der Meyden. The dynamic logic of permission. *Journal of Logic and Computation*, 6(3) :465–479, 1996.

# Analyse du comportement d'opérateurs de fusion basés sur la somme : du cadre commensurable au cadre incommensurable

Salem Benferhat benferhat@cril.univ-artois.fr
Sylvain Lagrue lagrue@cril.univ-artois.fr
Julien Rossit rossit@cril.univ-artois.fr

Université Lille-Nord de France
CRIL-CNRS UMR 8081
Faculté des Sciences Jean Perrin
rue Jean Souvraz, SP18
62307 Lens France

**Résumé :**

Différentes méthodes ont été proposées dans la littérature pour fusionner les croyances potentiellement conflictuelles d'agents. En particulier, lorsque les échelles utilisées par les agents (ou les sources) pour évaluer la pertinence de leurs informations sont commensurables, l'opérateur de fusion basé sur la somme jouit d'une certaine popularité. En effet, cet opérateur satisfait de nombreuses propriétés intuitives, en particulier celle de la majorité, et il adopte un attitude discriminante au moment de dériver ses conclusions.

Cet article étudie en profondeur une méthode également basée sur l'opérateur somme lorsque les agents ne possèdent pas d'échelle commune pour évaluer la fiabilité des différentes informations dont ils disposent. Cet opérateur peut être caractérisé au moyen d'un ensemble infini d'échelles compatibles, ce qui permet de restaurer la commensurabilité, ou en terme d'ordre de Pareto sur l'ensemble des mondes possibles. Puis nous étudions plus particulièrement des familles d'échelles compatibles afin de caractériser des opérateurs moins prudents. Enfin nous fournissons une analyse complète en terme de postulats de rationalité et de prudence des inférences de tous les opérateurs de fusions présentés dans cet article.

**Mots-clés :** Fusion de croyances, incommensurabilité, majorité, consensus

**Abstract:**

Different methods have been proposed in the literature for merging multiple and potentially conflicting informations. Sum-based operators offer a natural method for merging commensurable prioritized belief bases. Their popularity is due to the fact that they satisfy the majority property and they adopt a non cautious attitude in deriving plausible conclusions.

This paper analyses the sum-based merging operator when sources to merge are incommensurable, namely they do not share the same meaning of uncertainty scales. We first show that the obtained merging operator can be equivalently characterized either in terms of an infinite set of compatible scales, or by a well-known Pareto ordering on a set of models. We then study different families of compatible scales useful for merging process. This paper also provides a postulates-based analysis of our merging operators.

**Keywords:** Belief Merging, incommensurability, majority, consensus

## 1 Introduction

Confronter différents points de vue, malgré leurs incohérences et leurs incertitudes, afin de produire une image la plus complète possible du monde est l'objet même de la fusion de croyances. Ce problème de fusion d'informations survient dès qu'il faut prendre en compte des bases de données distribuées, des systèmes multi-agents ou tous les cas d'informations distribuées en général (le web restant l'exemple le plus frappant). Parmi les nombreux opérateurs qui ont été proposés dans la littérature pour la fusion d'informations incertaines (voir par exemple [9, 13, 14] ou encore [5, 10, 16]), toutes font l'hypothèse que les échelles d'évaluation de l'incertitude ou de la pertinence des informations est commune à toutes les sources. En d'autres termes, ces méthodes supposent que tous les poids, utilisés pour quantifier l'incertitude ou l'importance des formules, sont commensurables. Cette hypothèse de commensurabilité fait réellement sens lorsque, par exemple, les pondérations représentent une ressource monétaire, lorsqu'elles sont issues de protocole de notation ou encore lorsque les valeurs que prennent les poids ne sont pas trop éloignées les unes des autres. Cette hypothèse est d'autant plus nécessaire que les méthodes précitées s'appuient généralement sur des opérateurs d'agrégation. En particulier, l'opérateur de fusion basé sur la somme jouit d'une certaine popularité. En effet, cet opérateur satisfait de nombreuses propriétés intuitives, par exemple celle de la majorité, et il adopte un attitude discriminante au moment de dériver ses conclusions.

En revanche, l'hypothèse de commensurabilité est beaucoup trop forte lorsque les informations proviennent de sources réellement hétérogènes et qu'elles ne partagent pas d'échelle commune. On ne trouve que peu de travaux qui traitent

de ces problèmes dans le cadre de la fusion de croyance [2, 8], mais aucun d'entre eux ne traite du cas d'opérateurs de fusion basés sur la somme, opérateur qui servira de point de départ et de fil conducteur à l'ensemble de cet article.

Ainsi, nous présentons tout d'abord en section 3 une extension naturelle de l'opérateur de fusion basé sur la somme permettant de traiter des bases de croyance incommensurables. Cette extension utilise pour cela la notion d'échelles compatibles. Nous montrons également dans cette même section que ce type de fusion peut être caractérisé par un critère de Pareto sur les mondes possibles. Nous analysons ensuite dans la section 4 le comportement logique de ces opérateurs, que se soit dans le cas commensurable ou dans le cas incommensurable. À titre d'illustration, dans le cas incommensurable, l'opérateur basé sur la somme perd son caractère majoritaire et devient même indépendant à la majorité. En section 5, nous présentons différentes méthodes d'inférences basée sur des restrictions sur les ensembles d'échelles compatibles (par exemple en ne traitant que des transformations linéaires ou en bornant le plus haut poids possible). Nous analysons en particulier l'impact de ces différentes restrictions d'un point de vue de postulats de rationalité et de prudence pour les opérateurs de fusion induits. Nous montrons par exemple que limiter le poids le plus élevé permet de retrouver la propriété de majorité, tandis qu'il est impossible de satisfaire des propriétés de consensus et d'équité si l'on ne sélectionne qu'une seule échelle compatible. Enfin, la section 6 fournit une comparaison en terme de prudence et de productivité des relations d'inférence des différentes approches considérées tout au long de cet article.

Mais avant de développer l'ensemble de ces résultats en détail, nous rappelons dans la section suivante une méthode bien connue de fusion de base de croyance basée sur la somme, dans le cas commensurable, sur laquelle nous nous appuierons dans la suite de l'article.

## 2 Fusion de croyances

Nous considérons dans cet article un langage propositionnel $\mathcal{L}$ possédant un nombre *fini* de variables propositionnelles. L'ensemble $\Omega$ représente l'ensembles des interprétations associées au langage considéré et $\omega$ représente un élement de $\Omega$. Les lettres grecques $\varphi$ et $\psi$ sont utilisées pour représenter des formules propositionnelles. La conséquence logique est notée $\models$, tandis que $Mod(\varphi)$ représente l'ensemble des interprétations satisfiant $\varphi$ (c'est à dire l'ensemble des interprétation $\omega \in \Omega$ telles que $\omega \models \varphi$). Les symboles $\top$ et $\bot$ représentent respectivement la tautologie et la contradiction.

### 2.1 Bases de croyance pondérées

Dans cet article, nous considérerons que chaque agent représente ses croyances, ainsi que la confiance qu'il a en chacune d'elles, au moyen d'un ensemble de formules pondérées. Les formules du langage $\mathcal{L}$ sont l'expression logique des différentes informations qu'il possède et le poids associé représente la fiabilité, la certitude ou encore la vraisemblance de ces informations. Ces bases de croyance permettent ainsi à un agent de modéliser des informations incertaines et ordonnées.

**Définition 1 (Base de croyance)** *Une base de croyance $B_i$ est un ensemble de formules pondérées, $B_i = \{(\varphi_{ij}, R_{B_i}(\varphi_{ij})),\ j \in \{1, ..., m_i\}\}$, avec $\varphi_{ij} \in \mathcal{L}$, et $R_{B_i}(\varphi_{ij}) \in \mathbb{N}^*$.*

Un couple $(\varphi_{ij}, R_{B_i}(\varphi_{ij}))$ signifie que pour l'agent $i$, la formule $\varphi_{ij}$ a une priorité de $R_{B_i}(\varphi_{ij})$. Intuitivement, les formules associées aux poids les plus élevés sont les formules les plus crues pour un agent donné. Nous ne considérons ici que les entiers positifs. Cette représentation est intimement liée aux approches développées en théorie des possibilités [4] qui utilise l'intervalle [0,1] au lieu des entiers, et dans les fonctions conditionnelles ordinales (OCF) qui quant à elles utilisent des ordinaux pour évaluer l'incertitude des informations [14, 17, 18].

De plus, nous notons $B_i^*$ l'ensemble des formules contenues dans $B_i$, sans considérer les poids. Cet ensemble représente l'ensemble des croyances de l'agent, sans priorité. Si cet ensemble peut potentiellement être contradictoire, nous considérerons dans cet article que $B_i^*$ est cohérent pour chaque agent.

Il est possible à partir d'une base de croyance de définir une relation de plausibilité sur l'ensemble des interprétations. On associera à une interprétation $\omega$ un poids $\kappa_{B_i}(\omega)$ tel que ce poids soit égal au poids de la formule la plus importante falsifiée par $\omega$.

**Définition 2 ($\kappa$-functions)** *Une $\kappa$-fonction $\kappa_{B_i}$ associée à une base de croyance $B_i$ est une fonction qui associe à chaque interprétation $\omega \in \Omega$ un entier $\kappa_{B_i}(\omega)$ tel que :*

$$\kappa_B(\omega) = \begin{cases} 0 \text{ si } \forall(\varphi_{ij},\ R_{B_i}(\varphi_{ij})) \in B_i,\ \omega \models \varphi_{ij}; \\ max\{R_{B_i}(\varphi_{ij}) : \omega \not\models \varphi_{ij}, \\ \qquad (\varphi_{ij}, R_{B_i}(\varphi_{ij})) \in B_i\} \text{ sinon.} \end{cases}$$

Contrairement aux poids sur les formules, plus le poids associé à une interprétation $\omega$ est petit, plus l'état du monde représenté par $\omega$ est vraisemblable pour l'agent. En particulier, les croyances courantes de l'agent sont représentées par l'ensemble des interprétations possédant le plus petit poids. C'est sur cet ensemble que l'agent s'appuiera pour fonder ses décisions.

**Exemple 1** *Considérons la base de croyance suivante : $B = \{(\neg a \vee b, 8), (a \vee b, 5), (a, 2)\}$. La Table 1 présente la $\kappa$-fonction $\kappa_B$ associée à $B$. $\omega_3$ est l'interprétation préférée. Elle représente donc les croyances courantes de l'agent.*

| $\omega_i \in \Omega$ | a | b | $\kappa_B(\omega_i)$ |
|---|---|---|---|
| $\omega_0$ | 0 | 0 | 5 |
| $\omega_1$ | 0 | 1 | 2 |
| $\omega_2$ | 1 | 0 | 8 |
| $\omega_3$ | 1 | 1 | **0** |

TABLE 1 – Un exemple de $\kappa$-fonction

## 2.2 Fusion de croyances basée sur la somme

Lorsque plusieurs agents mettent en commun leurs croyances, celles-ci sont rarement cohérentes. Aussi l'objet de la fusion de croyances est de résoudre les conflits dans les croyances des agents afin d'obtenir une vision du monde plus précise que si l'on ne considère que les croyances d'un agent. Cette section présente une méthode de fusion, basée sur l'opérateur d'agrégation *somme*, lorsque les agents partagent une même échelle pour évaluer la pertinence de leurs croyances.

Considérons un multi-ensemble $E = \{B_1, ..., B_n\}$ contenant $n$ bases de croyance et $\mu$ une formule propositionnelle représentant les contraintes d'intégrité à satisfaire. Le but d'un opérateur de fusion sera, à partir de cet ensemble $E$, d'ordonner les différentes interprétations de $\Omega$ en fonction de toutes les informations données par les agents. Nous noterons $\lhd^E$ un tel ordre. À partir de cet ordre, nous pouvons représenter les croyances collectives issues de $E$, notées $\triangle^E$, comme étant les interprétations *minimales* de $\lhd^E$. Différentes méthodes ont été proposées dans la littérature pour générer $\lhd^E$, la plupart basée sur des opérateurs d'agrégations telle la somme ou le maximum [10, 12, 13].

Afin de calculer l'ordre $\lhd^E$, résultat de la fusion, nous associons à chaque interprétation un *profil*, noté $\nu_E(\omega)$, et défini par :

$$\nu_E(\omega) = \langle \kappa_{B_1}(\omega), ..., \kappa_{B_n}(\omega) \rangle$$

Ce profil représente la cohérence de l'interprétation $\omega$ vis-à-vis de chacune des bases de croyance. Le point de départ de cet article est un opérateur basé sur la somme et dont nous notons le résultat $\lhd^E_\Sigma$. Le calcul de cet ordre repose sur l'agrégation des profils au moyen de l'opérateur somme. Puis les éléments de $\Omega$ sont ordonnés en fonction de cette agrégation : plus la somme est petite, plus elle est représentative de $E$. Plus formellement :

**Définition 3 ($\lhd^E_\Sigma$)** *Soient $\omega$ et $\omega'$ deux interprétations de $\Omega$. Soient $\nu_E(\omega)$ et $\nu_E(\omega')$ leur profil respectif. Alors :*

$$\omega \lhd^E_\Sigma \omega' \ \textit{ssi}\ \ \Sigma(\nu_E(\omega)) < \Sigma(\nu_E(\omega')).\ ^1$$

Les modèles du résultat de la fusion, $\triangle^\mu_\Sigma(E)$, sont les modèles de $\mu$ qui sont minimaux relativement à $\lhd^E_\Sigma$. Plus formellement :

$$Mod(\triangle^\mu_\Sigma(E)) = Min(Mod(\mu), \lhd^E_\Sigma)$$

**Exemple 2** *Considérons $E = \{B_1,\ B_2\ B_3\}$ avec $B_1 = \{(a, 6), (b, 3)\}$, $B_2 = \{(a \vee b, 3), (\neg b, 1)\}$ et $B_3 = \{(\neg a, 5)\}$. Supposons que $\mu \equiv \neg a \vee \neg b$. Chaque profil associé à une interprétation est donnée par la Table 2, d'où l'on peut déduire que le résultat de la fusion est $Mod(\triangle^\mu_\Sigma(E)) = \{\omega_1\}$.*

| $\omega \in \Omega$ | a | b | $\nu_E(\omega)$ | $\Sigma(\nu_E(\omega))$ |
|---|---|---|---|---|
| $\omega_0$ | 0 | 0 | ⟨6,3,0⟩ | 9 |
| $\omega_1$ | **0** | **1** | **⟨6,1,0⟩** | **7** |
| $\omega_2$ | 1 | 0 | ⟨3,0,5⟩ | 8 |
| $\omega_3$ | 1 | 1 | ⟨0,1,5⟩ | 6 |

TABLE 2 – Profils associés à chaque interprétation

1. avec $\Sigma(\nu_E(\omega)) = \sum_{i=1,..,n} \kappa_{B_i}(\omega)$

L'opérateur de fusion basé sur la somme est dépendant de la majorité. En particulier, la répétition d'une information influe sur le résultat de la fusion. Ce comportement est particulièrement adapté lorsque les croyances des agents sont indépendantes et que ceux-ci sont de fiabilité équivalente.

# 3 Extension de l'opérateur basé sur la somme au cas incommensurable

L'opérateur de fusion basé sur la somme défini précédemment repose sur une hypothèse forte : l'agrégation des différents poids par la somme impose que les poids soient commensurables. En d'autres termes, les agents doivent posséder les mêmes échelles d'évaluation de la pertinence de leurs informations. Lorsque ces poids représentent des utilités ou des gains, cette hypothèse se justifie totalement. Mais lorsque ces poids proviennent de sources hétérogènes, dont les qualités intrinsèques diffèrent et sont inconnues (on pourrait prendre l'exemple d'informations tirées du web, des témoignages de personnes dont on ne sait pas si elles ont assisté à la scène ou encore des domaines demandant une forte expertise), cette hypothèse n'a pas de sens. C'est pourquoi, nous présentons dans cette section une approche basée sur les échelles compatibles permettant de traiter ce cadre de fusion incommensurable.

## 3.1 Un approche basée sur les échelles compatibles

Une approche naturelle et prudente repose sur l'utilisation de l'ensemble des échelles compatibles [2]. Cette méthode affecte de nouveaux poids aux différentes informations contenues dans les différentes bases de croyance, tout en maintenant l'ordre à l'intérieur de chaque base. Plus formellement :

**Définition 4 (Échelles compatibles)** *Une échelle compatible $\mathcal{S}$ est une fonction qui associe à $E = \{B_1, ..., B_n\}$ un nouveau multi-ensemble $E^{\mathcal{S}} = \{B_1^{\mathcal{S}}, ..., B_n^{\mathcal{S}}\}$tel que pour tout $B_i \in E$ :*

*(i)* $B_i^{\mathcal{S}} = \{(\varphi_{ij}, \mathcal{S}(\varphi_{ij})) : (\varphi_{ij}, R_{B_i}(\varphi_{ij})) \in B_i\}$

*(ii)* $\forall B_i \in E$, $\forall (\varphi_{ij}, R_{B_i}(\varphi_{ij})) \in B_i$, $\forall (\varphi_{ij'}, R_{B_i}(\varphi_{ij'})) \in B_i$, $R_{B_i}(\varphi_{ij}) \leq R_{B_i}(\varphi_{ij'})$ *ssi* $\mathcal{S}(\varphi_{ij}) \leq \mathcal{S}(\varphi_{ij'})$.

Nous illustrons ce concept par l'exemple suivant.

**Exemple 3** *Considérons l'ensemble des bases de croyance $E = \{B_1, B_2, B_3\}$ traitées précédemment dans l'exemple 2. Le tableau 3 fournit trois nouvelles échelles : $\mathcal{S}_1$, $\mathcal{S}_2$ et $\mathcal{S}_3$.*

| | $\varphi_{ij}$ | $R_{B_i}(\varphi_{ij})$ | $\mathcal{S}_1(\varphi_{ij})$ | $\mathcal{S}_2(\varphi_{ij})$ | $\mathcal{S}_3(\varphi_{ij})$ |
|---|---|---|---|---|---|
| $B_1$ | $a$ | 6 | 2 | 6 | 1 |
| | $b$ | 3 | 1 | 4 | 5 |
| $B_2$ | $a \vee b$ | 3 | 2 | 3 | 3 |
| | $\neg b$ | 1 | 1 | 2 | 1 |
| $B_3$ | $\neg a$ | 5 | 1 | 4 | 5 |

TABLE 3 – Exemples d'échelles compatibles

*Les échelles $\mathcal{S}_1$ et $\mathcal{S}_2$ sont compatibles, car elles préservent l'ordre initial des informations contenues dans les bases. À l'opposé, $\mathcal{S}_3$ n'est pas compatible car elle inverse les priorités de $B_1$.*

L'ensemble des échelles compatibles avec $E$ est noté $\mathbb{S}_E$. Notons que $\mathbb{S}_E$ n'est jamais vide, le plus simple étant de considérer l'échelle qui ne change pas les poids initiaux. Si l'on considère une échelle compatible $\mathcal{S} \in \mathbb{S}_E$, la base de croyance obtenue à partir de $B_i$ en utilisant $\mathcal{S}$ est notée $B_i^{\mathcal{S}}$. Plus formellement, $B_i^{\mathcal{S}}$ est obtenue en remplaçant chaque paire $(\varphi_{ij}, R_{B_i}(\varphi_{ij}))$ par $(\varphi_{ij}, \mathcal{S}(\varphi_{ij}))$. Enfin, $E^{\mathcal{S}}$ représente le multi-ensemble obtenu en appliquant $\mathcal{S}$ à chacune des $B_i$ appartenant à $B$.

**Définition 5 ($\blacktriangleleft_\Sigma^E$)** *Soient $\omega$ et $\omega'$ deux interprétations de $\Omega$. Alors :*

$$\omega \blacktriangleleft_\Sigma^E \omega' \text{ ssi } \forall \mathcal{S} \in \mathbb{S}_E,\ \omega \lhd_\Sigma^{E^{\mathcal{S}}} \omega'$$

*avec $\lhd_\Sigma^{E^{\mathcal{S}}}$ le résultat de l'application de la définition 3 sur $E^{\mathcal{S}}$.*

Comme définis précédemment, les modèles de $\blacktriangle_\Sigma^\mu(E)$ sont :

$$Mod(\blacktriangle_\Sigma^\mu(E)) = Min(Mod(\mu), \blacktriangleleft_\Sigma^E)$$

**Exemple 4** *Considérons à nouveau $B_1 = \{(a, 6), (b, 3)\}$, $B_2 = \{(a \vee b, 3), (\neg b, 1)\}$ et $B_3 = \{(\neg a, 5\}$. La table 4 contient les profils associés à chaque interprétations de $\mathcal{L}$. Si l'on considère $\mu \equiv \top$, les éléments en gras représentent les modèles de $\triangle_\Sigma^\mu(E^{\mathcal{S}_i})$ pour une échelle donnée $\mathcal{S}_i$. Par exemple, $\omega_2$ et $\omega_3$ sont modèles de $\triangle_\Sigma^\mu(E^{\mathcal{S}_1})$ pour $\mathcal{S}_1$.*

| $\omega \in \Omega$ | a | b | $\nu_{E^{\mathcal{S}_1}}(\omega)$ | $\Sigma_{\mathcal{S}_1}$ | $\nu_{E^{\mathcal{S}_2}}(\omega)$ | $\Sigma_{\mathcal{S}_2}$ |
|---|---|---|---|---|---|---|
| $\omega_0$ | 0 | 0 | $\langle 2,2,0\rangle$ | 4 | $\langle 6,3,0\rangle$ | 9 |
| $\omega_1$ | 0 | 1 | $\langle 2,1,0\rangle$ | 3 | $\langle 6,2,0\rangle$ | 8 |
| $\omega_2$ | 1 | 0 | $\langle \mathbf{1},\mathbf{0},\mathbf{1}\rangle$ | **2** | $\langle 4,0,4\rangle$ | 8 |
| $\omega_3$ | 1 | 1 | $\langle \mathbf{0},\mathbf{1},\mathbf{1}\rangle$ | **2** | $\langle \mathbf{0},\mathbf{2},\mathbf{4}\rangle$ | **6** |

TABLE 4 – Profils associés

### 3.2 Caractérisation du résultat de la fusion incommensurable

En l'absence de borne supérieure, il est impossible de considérer l'ensemble des échelles compatibles. C'est pourquoi nous caractérisons dans cette section le résultat de la fusion basée sur les échelles compatibles en terme d'ordre sur les interprétations.

Tout d'abord, considérons la proposition suivante qui montre qu'une interprétation $\omega$ est modèle de $\blacktriangle^{\mu}_{\Sigma}(E)$ si et seulement si il existe une échelle compatible pour laquelle $\omega$ est modèle de la fusion commensurable $\triangle^{\mu}_{\Sigma}(E^{\mathcal{S}})$. Plus formellement :

**Proposition 1** *Soit $E$ un multi-ensemble de bases de croyance. Une interprétation $\omega \in Mod(\blacktriangle^{\mu}_{\Sigma}(E))$ si et seulement si il existe une échelle compatible $\mathcal{S} \in \mathbb{S}_E$ telle que $\omega \in Mod(\triangle^{\mu}_{\Sigma}(E^{\mathcal{S}}))$.*

Cette proposition peut être généralisée en traitant globalement l'ordre $\blacktriangleleft^{E}_{\Sigma}$ et non plus seulement ses éléments minimaux. Ce critère s'avère être un *critère de Pareto* [15].

**Proposition 2** *Soient $\omega$ et $\omega'$ deux interprétations de $\Omega$. Alors $\omega \blacktriangleleft^{E}_{\Sigma} \omega'$ si et seulement si :*

*(i)* $\forall j \in \{1,..,n\},\ \kappa_{B_j}(\omega) \leq \kappa_{B_j}(\omega')$ *et*

*(ii)* $\exists i \in \{1,...,n\},\ \kappa_{B_i}(\omega) < \kappa_{B_i}(\omega')$.

La première condition signifie que $\omega$ est au moins aussi préférée que $\omega'$ dans chacune des bases initiales et la la seconde condition stipule que $\omega$ est strictement préférée à $\omega'$ dans au moins une base.

## 4 Comportement logique et postulats de rationalité

Afin de caractériser les opérateurs de fusion, des postulats ont été proposé dans la littérature, les plus connus provenant de [10]. Cependant, ces postulats ont été proposés dans un cadre différent, où les croyances des agents sont représentées uniquement à l'aide de formules propositionnelles. Nous avons donc adapté ces postulats à notre propos [2], afin qu'ils puissent traiter des bases de croyance contenant des formules pondérées.

**(IC0)** $\triangle^{\mu}(E) \models \mu$ ;

**(IC1)** Si $\mu$ est cohérent, alors $\triangle^{\mu}(E)$ est cohérent ;

**(IC2*)** Si $\bigwedge_{B\in E} B^*$ est cohérent avec $\mu$, alors $\triangle^{\mu}(E) \equiv \bigwedge_{B\in E} B^* \wedge \mu$ ;

**(IC3*)** Si $E_1 \equiv_R E_2$ et $\mu_1 \equiv \mu_2$, alors $\triangle^{\mu_1}(E_1) \equiv \triangle^{\mu_2}(E_2)$ ;

**(IC4*)** Si $B_1^* \models \mu$ et $B_2^* \models \mu$, alors $\triangle^{\mu}(\{B_1, B_2\}) \wedge B_1^*$ est cohérent ssi $\triangle^{\mu}(\{B_1, B_2\}) \wedge B_2^*$ est cohérent ;

**(IC5)** $\triangle^{\mu}(E_1) \wedge \triangle^{\mu}(E_2) \models \triangle^{\mu}(E_1 \bigsqcup E_2)$ ;

**(IC6)** Si $\triangle^{\mu}(E_1) \wedge \triangle^{\mu}(E_2)$ est cohérent, alors $\triangle^{\mu}(E_1 \bigsqcup E_2) \models \triangle^{\mu}(E_1) \wedge \triangle^{\mu}(E_2)$ ;

**(IC7)** $\triangle^{\mu_1}(E) \wedge \mu_2 \models \triangle^{\mu_1 \wedge \mu_2}(E)$ ;

**(IC8)** Si $\triangle^{\mu_1}(E) \wedge \mu_2$ est cohérent, alors $\triangle^{\mu_1 \wedge \mu_2}(E) \models \triangle^{\mu_1}(E) \wedge \mu_2$.

D'autres postulats, capturant d'autres comportements, ont également été proposés dans [10] :

**(IC6')** Si $\triangle^{\mu}(E_1) \wedge \triangle^{\mu}(E_2)$ est cohérent, alors $\triangle^{\mu}(E_1 \bigsqcup E_2) \models \triangle^{\mu}(E_1) \vee \triangle^{\mu}(E_2)$ ;

**(MAJ)** $\exists m \in \mathbb{N} :\ \triangle^{\mu}(E \sqcup B_i^m) \models B_i^*$, (avec $B_i^m = \{B_i\} \sqcup ... \sqcup \{B_i\}$ $m$ fois et $\sqcup$ l'union multi-ensembliste) ;

**(MI)** $\forall m,\ \triangle^{\mu}(E_1 \sqcup E_2^m) \equiv \triangle^{\mu}(E_1 \sqcup E_2)$.

Le postulat **(MAJ)** caractérise les opérateurs majoritaire : si un nombre suffisamment grand d'agents ont les mêmes croyances, alors ces croyances doivent être acceptées collectivement. À l'opposé, le postulat d'indépendance à la majorité stipule que le résultat de la fusion est totalement indépendant de la répétition des croyances des agents. La fusion obtenue dans ce cas est alors plus robuste, mais également moins intuitive. Le postulat **(IC6')** est quant à lui un affaiblissement de **(IC6)**. Enfin nous introduisons un dernier postulat, original, dit de consensus, qui stipule que si toutes les bases sont cohérentes avec la contrainte $\mu$, alors le résultat de la fusion contient au moins un modèle préféré pour chacune des bases.

**(CSS)** $\forall B_i \in E$, si $B_i^* \models \mu$, alors $B_i^* \wedge \triangle^{\mu}(E)$ est cohérent.

Ce postulat généralise le postulat de tempérance proposé dans [7].

### 4.1 Le cas commensurable

Nous étudions dans cette section l'opérateur basé sur la somme dans le cas commensurable $\triangle^{\mu}_{\Sigma}$. Il s'avère que celui-ci vérifie la plupart des postulats :

**Proposition 3** $\triangle^{\mu}_{\Sigma}$ *satisfait* **(IC0)**, **(IC1)**, **(IC2$^*$)**, **(IC3$^*$)**, **(IC5)**, **(IC6)**, **(IC6')**, **(IC7)**, **(IC8)** *et* **(MAJ)**.

Cependant, $\triangle^{\mu}_{\Sigma}$ falsifie **(IC4$^*$)**, **(MI)** et **(CSS)**. Le fait que $\triangle^{\mu}_{\Sigma}$ falsifie **(MI)** et **(CSS)** est attendu, puisqu'il vérifie **(MAJ)**.

L'exemple suivant montre que $\triangle^{\mu}_{\Sigma}$ ne peut pas satisfaire **(IC4$^*$)** :

**Exemple 5** *Considérons deux base de croyance* $B_1 = \{(\neg a, 2), (b, 1)\}$ *et* $B_2 = \{(a, 3), (b, 2)\}$, *et un contrainte d'intégrité* $\mu = a \vee b$.

*En considérant* $E = \{B_1, B_2\}$, *nous avons* $B_1 \models \mu$ *et* $B_2 \models \mu$. *De la table 5, en considérant toujours* $\mu = a \vee b$, *nous obtenons que* $Mod(\triangle^{\mu}_{\Sigma}(E)) = \{\omega_3\}$, *et donc que* $\triangle^{\mu}_{\Sigma}(E) \equiv a \wedge b$. *Ainsi, sur cet exemple,* $\triangle^{\mu}(\{B_1, B_2\}) \wedge B_2^*$ *est cohérent, mais* $\triangle^{\mu}(\{B_1, B_2\}) \wedge B_1^*$ *ne l'est pas, ce qui contrevient bien au postulat* **(IC4$^*$)**.

| | a | b | $\kappa_{B_1}(\omega)$ | $\kappa_{B_2}(\omega)$ | $\nu_E(\omega)$ | $\Sigma(\nu_E(\omega))$ |
|---|---|---|---|---|---|---|
| $\omega_0$ | 0 | 0 | 1 | 3 | ⟨1,3⟩ | 4 |
| $\omega_1$ | 0 | 1 | 0 | 3 | ⟨0,3⟩ | 3 |
| $\omega_2$ | 1 | 0 | 2 | 2 | ⟨2,2⟩ | 4 |
| $\omega_3$ | 1 | 1 | 2 | 0 | ⟨2,0⟩ | 2 |

TABLE 5 – Profils associés pour l'exemple 5

### 4.2 Le cas incommensurable

Lorsqu'on considère l'opérateur basé sur la somme adapté au cadre incommensurable, le comportement diffère de celui défini pour le cas commensurable. En particulier, de dernier devient indépendant de la majorité et il devient même consensuel.

**Proposition 4** $\blacktriangle^{\mu_1}_{\Sigma}(E)$ *satisfait* **(IC0)**, **(IC1)**, **(IC2$^*$)**, **(IC3$^*$)**, **(IC4$^*$)**, **(IC5)**, *et* **(IC7)**.

En revanche, $\blacktriangle^{\mu}_{\Sigma}$ falsifie **(IC6)**, **(IC6')**, **(IC8)**, et **(MAJ)**. Nous fournissons un contre-exemple pour **(IC6)**.

**Exemple 6** *Considérons* $\mu = \top$, $E_1 = \{B_1 = \{(a, 1)\}, B_2 = \{(\neg a, 1)\}\}$ *et* $E_2 = \{B_3 = \{(a, 1)\}\}$. *La table 6 contient les profils associés à chaque interprétation pour* $\blacktriangle^{\mu}_{\Sigma}(E_1)$, $\blacktriangle^{\mu}_{\Sigma}(E_2)$ *et* $\blacktriangle^{\mu}_{\Sigma}(E_1 \bigsqcup E_2)$.

| | a | b | $\nu_{E_1}(\omega)$ | $\nu_{E_2}(\omega)$ | $\nu_{E_1 \bigsqcup E_2}(\omega)$ |
|---|---|---|---|---|---|
| $\omega_0$ | 0 | 0 | ⟨**1,0**⟩ | ⟨1⟩ | ⟨**1,0,1**⟩ |
| $\omega_1$ | 0 | 1 | ⟨**1,0**⟩ | ⟨1⟩ | ⟨**1,0,1**⟩ |
| $\omega_2$ | 1 | 0 | ⟨**0,1**⟩ | ⟨**0**⟩ | ⟨**0,1,0**⟩ |
| $\omega_3$ | 1 | 1 | ⟨**0,1**⟩ | ⟨**0**⟩ | ⟨**0,1,0**⟩ |

TABLE 6 – Profils associés à $\blacktriangle^{\mu}_{\Sigma}(E_1)$, $\blacktriangle^{\mu}_{\Sigma}(E_1)$ et $\blacktriangle^{\mu}_{Max}(E_1 \bigsqcup E_2)$

*Nous avons bien* $\blacktriangle^{\mu}_{\Sigma}(E_1) \equiv \top$ *et* $\blacktriangle^{\mu}_{\Sigma}(E_2) \equiv a$. *En outre, nous avons* $\blacktriangle^{\mu}_{\Sigma}(E_1 \bigsqcup E_2) \equiv \top$, *mais* $\blacktriangle^{\mu}_{\Sigma}(E_1) \wedge \blacktriangle^{\mu}_{\Sigma}(E_2) \equiv a$. *Ainsi,* **(IC6)** *n'est pas satisfait car* $\blacktriangle^{\mu}_{\Sigma}(E_1) \wedge \blacktriangle^{\mu}_{\Sigma}(E_2)$ *est cohérent, mais* $\blacktriangle^{\mu}_{\Sigma}(E_1 \bigsqcup E_2) \not\models \blacktriangle^{\mu}_{\Sigma}(E_1) \wedge \blacktriangle^{\mu}_{\Sigma}(E_2)$.

La non-satisfaction de **(IC8)** est due au fait que lorsqu'on considère des échelles incommensurables, $\blacktriangleleft^{E}_{\Sigma}$ n'est qu'un ordre partiel. À titre illustratif, l'opérateur basé sur l'inclusion ensembliste (qui est un ordre partiel très particulier) proposé dans [6], ne satisfait pas **(IC8)** non plus.

**Proposition 5** $\blacktriangle^{\mu}_{\Sigma}(E)$ *satisfait* **(MI)** *et* **(CSS)**.

Pour ce qui est de la majorité, l'opérateur incommensurable basé sur la somme devient consensuel et indépendant de la majorité : l'opérateur devient donc d'un certain point de vue égalitariste, alors qu'il était utilitariste dans sa version commensurable.

## 5 Restrictions des échelles compatibles

Considérer l'ensemble des échelles compatibles produit un résultat prudent. Mais il repose sur l'hypothèse forte d'incommensurabilité, justifié par l'absence d'informations permettant de sélectionner un sous-ensemble d'echelles compatibles. Le but de cette section est d'affaiblir l'hy-

pothèse d'incommensurabilité et donc de limiter cette notion dans le but d'obtenir des ensemble d'échelles compatibles plus restreints. Nous avons pour cela besoin de définir des classes d'incommensurabilité qui peuvent être découpées ainsi :

- *la classe des échelles bornées* $S^{(p)}$ : le plus grand poids associé à une échelle ne peut dépasser un seuil fixé, représenté par un entier positif $p$. Cette classe empêche de considérer des poids arbitrairement grands.
- *la classe des échelles linéaires* $S^l$ : cette classe maintient la proportion entre les poids à une constante translative près ;
- *la classe des échelles proportionnelles* $S^w$ : cette classe maintient la proportion entre les poids d'une source donnée ;
- *la classe des échelles translatives* $S^s$ : cette classe maintient la distance entre les différents poids pour une source donnée, seule la distance des poids à 0 peut être augmentée.

La table 7 donne une définition formelle de chacune de ces restrictions.

| Classe | Notation | $\{\mathcal{S}\ t.q.\ \forall B_i \in E,\ \forall \varphi \in B_i : \dots\}$ |
|---|---|---|
| Bornée | $S^{(p)}$ | $\mathcal{S}(\varphi) \leq p$ |
| Linéaire | $S^l$ | $\mathcal{S}(\varphi) = a_i.R_{B_i}(\varphi) + b_i$ $avec\ a_i > 0,\ b_i \geq 0$ |
| Proport. | $S^w$ | $\mathcal{S}(\varphi) = a_i.R_{B_i}(\varphi)$ $avec\ a_i > 0$ |
| Translative | $S^s$ | $\mathcal{S}(\varphi) = R_{B_i}(\varphi) + b_i$ $avec\ b_i \geq 0$ |

TABLE 7 – Différentes classes d'échelles compatibles

Les échelles bornées font réellement sens, puisque dans les cas pratiques la taille des poids est finie. Étant donnés un ensemble de base de croyance, la plus petite borne $p$ possible est $p_{min} = max\{|B_i| : B_i \in E\}$, avec $|B_i|$ représentant le nombre des différents rangs de $B_i$. Si $p < p_{min}$, alors l'ensemble des échelles compatible est vide.

La classe des échelles translatives quant à elle maintient les distances entre les différents poids. En effet, quelque soit l'échelle translative considérée, $\forall(\varphi_{ij}, R_{B_i}(\varphi_{ij})), (\varphi_{ij'}, R_{B_i}(\varphi_{ij'})) \in B_i$, nous avons :

$$\mathcal{S}(\varphi_{ij}) - \mathcal{S}(\varphi_{ij'}) = R_{B_i}(\varphi_{ij}) - R_{B_i}(\varphi_{ij'}).$$

Les échelles proportionnelles sont obtenues en multipliant chaque poids $R_{B_i}(\varphi_{ij})$ d'une source donnée par un entier strictement positif $a_i$. Dans ce cas, $a_i$ peut être vu comme un degré de fiabilité de la source $B_i$ et l'opérateur de fusion devient ainsi une somme pondérée. Enfin, la classe des échelles linéaires généralise les classes translatives et proportionnelles.

Un opérateur de fusion peut ainsi être obtenu à partir de chacune de ces classes. En effet, nous pouvons définir $\blacktriangleleft^{\mu}_{\Sigma,S^{(p)}}$ (resp $\blacktriangleleft^{\mu}_{\Sigma,S^l}$, $\blacktriangleleft^{\mu}_{\Sigma,S^w}$, et $\blacktriangleleft^{\mu}_{\Sigma,S^s}$ ) comme étant les ordres issus de la définition 5 en remplaçant $\mathbb{S}$ par $S^{(p)}$ (resp. $S^l$, $S^w$, and $S^s$).

Nous analysons dans les section suivantes l'impact de ces différentes restrictions sur la prudence et la productivité de ces opérateurs de fusion ainsi que leur comportement vis-à-vis des postulats de rationalité.

### 5.1 Impact sur les inférences

Le premier résultat surprenant est que la restriction des échelles compatibles aux classes affines et linéaires n'affecte pas le résultat de la fusion.

**Proposition 6** *Soient $\omega$, $\omega'$ deux interpretations de $\Omega$. Alors :*

$$\omega \blacktriangleleft^{E}_{\Sigma,S^l_E} \omega' \ \textit{ssi}\ \omega \blacktriangleleft^{E}_{\Sigma,S^w_E} \omega' \ \textit{ssi}\ \omega \blacktriangleleft^{E}_{\Sigma} \omega'.$$

En revanche, l'inférence basée sur les échelles bornées est en général plus productive que $\blacktriangle^{\mu}_{\Sigma}$. Mais cette inférence dépend fortement du choix de $p$. En particulier, si l'on choisit une valeur particulière de $p$, l'opérateur de fusion standard basé sur la somme peut être recouvré dans certains cas bien précis :

**Proposition 7** *Supposons que $E$ contiennent des bases de croyance qui renferment exactement le même nombre de rangs, $p_0$. Supposons également que le poids maximal des différents $B_i$ soit $p_0$. Dans ce cas $\forall\omega,\ \omega' \in \Omega$ :*

$$\omega \blacktriangleleft^{E}_{\Sigma,S^{(p_0)}_E} \omega' \ \textit{ssi}\ \omega \lhd^{E^{\mathcal{S}_{p_0}}}_{\Sigma} \omega'$$

Si l'on considère plus précisément l'inférence basée sur les translations, elle est également en général plus productive que $\blacktriangle^{\mu}_{\Sigma}$. Il est même possible d'exhiber un critère permettant de caractériser l'ordre induit par $\blacktriangleleft^{E}_{\Sigma,S^s_E}$, cette caractérisation étant décrite dans la proposition suivante.

**Proposition 8** *Soient $\omega$, $\omega'$ deux interprétations de $\Omega$. Alors $\omega \blacktriangleleft^{E}_{\Sigma,S^s_E} \omega'$ si et seulement si :*

*(i)* $\Sigma(\nu_E(\omega)) < \Sigma(\nu_E(\omega'))$ *et*
*(ii)* $\forall B_i : \omega' \models B_i, \omega \models B_i$.

Le premier critère stipule que la somme des éléments du profil associé à $\omega$ doit être inférieure à la somme des éléments du profil associé à $\omega'$, tandis que le second critère précise que $\omega$ doit également être modèle des bases dont $\omega'$ est modèle.

L'ensemble des relations liant les différents opérateurs est repris et complété par la figure 1.

### 5.2 Impact vis-à-vis des postulats de rationalité

La table 8 synthétise le comportement des opérateurs précédemment définis vis-à-vis des postulats de rationalité traitant de majorité ou de consensus. Nous avons également comme résultats complémentaires $\blacktriangle^{\mu}_{\Sigma,S^{(p)}}$ et $\blacktriangle^{\mu}_{\Sigma,S^s}$ qui satisfont tous deux **(IC0)**, **(IC1)**, **(IC2$^*$)**, **(IC3$^*$)**, **(IC5)**, et **(IC7)**.

| | (IC4$^*$) | (IC6) | (IC8) | (MAJ) | (MI) | (CSS) |
|---|---|---|---|---|---|---|
| $\triangle^{\mu}_{\Sigma}$ | - | √ | √ | √ | - | - |
| $\blacktriangle^{\mu}_{\Sigma}$ | √ | - | - | - | √ | √ |
| $\blacktriangle^{\mu}_{\Sigma,S^s}$ | √ | - | - | - | √ | √ |
| $\blacktriangle^{\mu}_{\Sigma,S^{(p)}}$ | √ | - | - | √ | - | - |

TABLE 8 – Synthèse des comportements logiques

La principale raison du comportement non-majoritaire de $\blacktriangle^{\mu}_{\Sigma}(E)$ tient au fait que les nouveaux poids associés par les échelles compatibles ne sont pas bornés. Par exemple, supposons qu'une base $B_1$ contienne $\varphi$ et qu'une autre base $B_2$ contienne $\neg\varphi$. Comme les échelles compatibles ne sont pas bornées, même si $B_1$ est répétée $m$ fois, il sera toujours possible de construire une échelle compatible avec des poids au moins $m$ fois plus élevés pour $B_2$ (et donc pour $\neg\varphi$), ce qui bloquera l'inférence de $\varphi$. C'est aussi pourquoi $\blacktriangle^{\mu}_{\Sigma,S^{(p)}}$ vérifie **(MAJ)**, au contraire des autres opérateurs.

Cependant, $\triangle^{\mu}_{\Sigma}$ satisfait la plupart des postulats de rationalité, à l'exception des postulats d'équité et de consensus, **(IC4$^*$)** et **(CSS)**. Une question peut néanmoins se poser : peut-on sélectionner une seule échelle compatible qui permette de satisfaire à la fois **(IC4$^*$)** et **(CSS)** ? La réponse est négative dans le cas général, sauf dans des cas très particuliers, comme celui décrit par la proposition suivante.

**Proposition 9** *Soit $E = \{B_1, B_2\}$ tel que $B_1 = (\varphi, R_{B_1}(\varphi))$ et $B_2 = (\varphi', R_{B_2}(\varphi'))$. Soit $\mathcal{S}$ une échelle compatible avec $E$, alors : $\triangle^{\mu}_{\Sigma,\mathcal{S}}$ vérifie $(\mathbf{IC4}^*)$ et $(\mathbf{CSS})$ ssi $\mathcal{S}(\varphi) = \mathcal{S}(\varphi')$.*

Mais comme nous l'avons précédemment indiqué, dans le cas général, il n'y a aucun espoir de satisfaire à la fois **(IC4$^*$)** et **(CSS)** en utilisant une seule échelle compatible.

**Proposition 10** *Il n'existe pas d'échelle compatible telle que $\triangle^{\mu}_{\Sigma,\mathcal{S}}$ satisfasse les postulats d'équité et de consensus pour un $E$ donné :*

$\nexists \mathcal{S}\, t.q. \forall E : \triangle^{\mu}_{\Sigma,\mathcal{S}}$ *satisfait* $(\mathbf{IC4}^*)$ *ou* $(\mathbf{CSS})$.

Comme contre-exemple, il est possible de considérer $E = \{B_1, B_2, B_3\}$ avec $B_1 = \{(a \wedge c, 1)\}$, $B_2 = \{(\neg a, 1)\}$, et $B_3 = \{(\neg c, 1), (a, 2)\}$.

## 6 Étude comparative d'un point de vue des inférences

Cette section étudie comparativement les opérateurs de fusion précédemment définis entre eux, mais aussi avec d'autres, du point de vue de la productivité et de la prudence des différentes relations d'inférence.

Par exemple, considérons les opérateurs basés sur le maximum $\blacktriangle^{\mu}_{Max}(E)$ (resp. $\triangle^{\mu}_{Max}(E)$ pour le cas commensurable) définis exactement comme $\blacktriangle^{\mu}_{\Sigma}(E)$ (resp. $\triangle^{\mu}_{\Sigma}(E)$) donné par la définition 5 (resp. la définition 3), excepté le fait que l'opérateur d'agrégation somme $\Sigma$ est remplacé par l'opérateur d'agrégation $Max$ (pour plus de détails, voir [2]). Il est bien connu, dans le cadre commensurable, que les opérateurs de fusion basés sur la somme et sur le maximum sont incomparables en terme de conclusions. En revanche, dans le cas incommensurable, ces deux opérateurs sont comparables, puisque nous avons une relation d'inclusion stricte entre ces deux relations d'inférence :

$$\blacktriangle^{\mu}_{\Sigma}(E) \models \blacktriangle^{\mu}_{Max}(E).$$

Une autre façon d'aborder la fusion de bases de croyance incommensurables est de voir l'ensemble $E$ comme une base unique de croyance partiellement ordonnée $(K^E, <_{K^E})$, avec $K^E$ un multi-ensemble contenant toutes les formules de $E$ et $<_{K^E}$ défini par : $\varphi_{ij} <_{K^E} \varphi_{ik}$ ssi $\exists B_i \in E$ tel que $\varphi_{ij} \in B_i$, $\varphi_{ik} \in B_i$ *et* $R_{B_i}(\varphi_{ij}) < R_{B_i}(\varphi_{ik})$. Dans ce cas, le résultat de la fusion devient une sélection de modèles préférés. Pour ce faire, la méthode basée sur l'inclusion définie dans [3] et rappelée ici peut être utilisée :

**Définition 6** *Une interprétation $\omega$ est dite $Incl$-préférée à une interpretation $\omega'$, noté $\omega \lhd_E^{Incl} \omega'$, si et seulement si :* $\forall \varphi \in K^E$ *t.q.* $\omega \not\models \varphi$ *et* $\omega' \models \varphi$, $\exists \psi \in K^E$ *t.q.* $\omega \models \psi$ *et* $\omega' \not\models \psi$ *et* $\psi <_{K^E} \varphi$.

Ceci nous permet maintenant de définir directement un opérateur de révision $\triangle_{Incl}^{\mu}$ basé sur $\lhd_E^{Incl}$. Les modèles de cette fusion seront les modèles minimaux pour $\lhd_E^{Incl}$ :

$$Mod(\triangle_{Incl}^{\mu}(E)) = Min(Mod(\mu), \lhd_E^{Incl}).$$

La proposition suivante stipule que si chaque base possède un seul élément, alors $\triangle_{Incl}^{\mu}$ et $\blacktriangle_{\Sigma}^{\mu}$ mènent au même résultat.

**Proposition 11** *Supposons que chaque $B_i$ contienne exactement une formule. Alors* $\forall \varphi \in \mathcal{L}$,

$$\triangle_{Incl}^{\mu}(E) \models \varphi \text{ ssi } \blacktriangle_{\Sigma}^{\mu}(E) \models \varphi$$

Cette proposition montre une fois de plus la différence de comportement de l'opérateur de fusion basé sur la somme dans le cas incommensurable. En effet, dans le cas commensurable, si chaque base ne possède qu'une seule formule pondérée à 1, alors l'opérateur basé sur la somme est plus productif que l'approche basée sur l'inclusion. En revanche, si les bases de croyance $B_i$ contiennent plus d'une formule, $\triangle_{Incl}^{\mu}(E)$ et $\triangle_{\Sigma}^{\mu}(E)$ redeviennent incomparables en terme de conclusions. Nous illustrons ceci à l'aide de l'exemple suivant :

**Exemple 7** *Considérons $E_1 = \{B_1, B_2, B_3\}$ tel que $B_1 = \{(a \wedge b, 1)\}$, $B_2 = \{(\neg a \wedge b, 1)\}$ et $B_3 = \{(a, 1)\}$. L'opérateur basé sur la somme va conclure $\{a\}$, contrairement à celui basé sur l'inclusion. Considérons maintenant $E_2 = \{B_1, B_2\}$ avec $B_1 = \{(b, 2), (a, 1)\}$ et $B_2 = \{(\neg a, 2)\}$. Cette fois-ci, l'opérateur basé sur l'inclusion déduit $\{b\}$, ce que ne déduira pas l'opérateur basé sur la somme.*

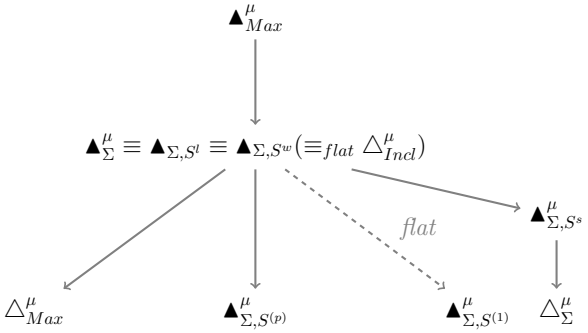



FIGURE 1 – Productivité et prudence des différents opérateurs proposés

La figure 1 présente une synthèse de l'ensemble des liens de tous les opérateurs présentés dans cet article en terme de productivité et de prudence d'inférence. La flèche $\triangle_1 \rightarrow \triangle_2$ signifie que le résultat de la fusion $\triangle_1$ peut être inféré par $\triangle_2$. Le cas de bases de croyance ne possédant qu'une seule formule est aussi représenté. Dans ce cas, $\triangle_{Incl}^{\mu}$ est plus prudent que la version bornée avec $p = 1$. En fait, dans ce cas, nous retrouvons l'inférence lexicographique [1, 11].

## 7 Conclusion

Cet article a analysé en détail différentes déclinaisons de l'opérateur de fusion basé sur la somme dans le cadre incommensurable. Nous avons proposé plusieurs caractérisations du résultat de ces opérateurs en terme d'ordre de Pareto et d'échelles compatibles. Nous avons également étudié le comportement de cet opérateur et nous avons montré que son comportement diverge dans le cas incommensurable : l'opérateur somme n'est plus dans ce cas un opérateur majoritaire. De surcroît, nous avons analysé différentes restrictions sur les échelles compatibles en définissant différentes classes. Par exemple les échelles bornées permettent de retrouver la propriété de majorité. En outre, nous avons considéré le postulat d'équité **(IC4$^*$)** et le nouveau postulat de consensus présenté dans cet article, **(CSS)**. En particulier, nous avons montré l'impossibilité de ne sélectionner qu'une

seule échelle compatible et de satisfaire ainsi ces deux postulats en même temps. Enfin, nous avons comparé les différentes relations d'inférence d'un point de vue de productivité et de prudence afin d'obtenir un panorama extrêmement complet des comportements d'opérateurs basés sur la somme lorsque l'hypothèse de commensurabilité ne tient plus. Cependant, diverses questions demeurent, en particulier l'existence d'une caractérisation complète en terme de postulats des différents opérateurs proposés ou encore la complexité théorique des méthodes de fusion définies.

## Références


[1] Salem Benferhat, Claudette Cayrol, Didier Dubois, Jérôme Lang, and Henri Prade. Inconsistency management and prioritized syntax-based entailment. In *Proceedings of the 13$^{th}$ International Joint Conference on Artificial Intelligence (IJCAI'93)*, pages 640–645, 1993.

[2] Salem Benferhat, Sylvain Lagrue, and Julien Rossit. Maximum-based merging of incommensurable ranked belief bases. In *Proceedings of the 22$^{th}$ National Conference on Artificial Intelligence (AAAI'07)*, pages 367–372, 2007.

[3] Gerhard Brewka. Preferred subtheories : an extended logical framework for default reasoning. In *Proceedings of the 11$^{th}$ International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1043–1048, 1989.

[4] Didier Dubois, Jérôme Lang, and Henri Prade. Possibilistic logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 439–513. Oxford University Press, 1994.

[5] Patricia Everaere, Sébastien Konieczny, and Pierre Marquis. Conflict-based merging operators. In *Proceedings of the 11$^{th}$ International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 348–357, 2008.

[6] Patricia Everaere, Sébastien Konieczny, and Pierre Marquis. A diff-based merging operator. In *Proceedings of the 12$^{th}$ International Workshop on Non-Monotonic Reasoning (NMR'08)*, pages 19–25, 2008.

[7] Patricia Evraere, Sébastien Konieczny, and Pierre Marquis. Propositional merging operators based on set-theoretic closeness. In *Proceedings of the18$^{th}$ European Conference on Artificial Intelligence (ECAI'08)*, pages 737–738. IOS Press, 2008.

[8] Qi Guilin, Weiru Liu, and David A. Bell. Merging stratified knowledge bases under constraints. In *Proceedings of the 21$^{th}$ National Conference on Artificial Intelligence (AAAI'06)*, pages 348–356, july 2006.

[9] Sébastien Konieczny, Jérôme Lang, and Pierre Marquis. DA$^2$ merging operators. *Journal of Logic and Computation*, 157(1-2) :49–79, 2004.

[10] Sébastien Konieczny and Ramón Pino Pérez. Merging information under constraints : a logical framework. *Journal of Logic and Computation*, 12(5) :773–808, 2002.

[11] Daniel J. Lehmann. Another perspective on default reasoning. *Annals of Mathematics and Artificial Intelligence*, 15(1) :61–82, 1995.

[12] Paolo Liberatore and Marco Schaerf. Arbitration : A commutative operator for belief revision. In *Proceedings of the 2$^{nd}$ World Conference on the Fundamentals of Artificial Intelligence*, pages 217–228, 1995.

[13] Jinxin Lin. Integration of weighted knowledge bases. *Artificial Intelligence*, 83(2) :363–378, 1996.

[14] Thomas Meyer. On the semantics of combination operations. *Journal of Applied Non-Classical Logics*, 11(1-2) :59–84, 2001.

[15] Hervé Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.

[16] Peter Revesz. On the semantics of arbitration. *International Journal of Algebra and Computation*, 7(2) :133–160, 1997.

[17] Wolfgang Spohn. Ordinal conditional functions : a dynamic theory of epistemic state. *Causation in Decision, Belief Change and Statistics*, pages 105–134, 1988.

[18] Mary-Anne Williams. Iterated theory base change : a computational model. In *Proceedings of 14$^{th}$ International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1541–1547, 1995.

# Argumentation et CP-jeux booléens

E. Bonzon*
elise.bonzon@mi.parisdescartes.fr

C. Devred†
caroline.devred@info.univ-angers.fr

M-C. Lagasquie-Schiex‡
lagasq@irit.fr

*LIPADE, Université Paris Descartes
45 rue des Saints Pères
75006 Paris, France

†LERIA, Université d'Angers
2 Boulevard Lavoisier
49045 Angers Cedex 01, France

‡IRIT, Université Paul Sabatier
118 route de Narbonne
F-31062 Toulouse Cedex 9, France

**Résumé :**
Des liens entre argumentation et théorie des jeux existent déjà. Par exemple, les jeux dynamiques permettent de simuler les interactions entre agents dans un processus d'argumentation. Dans ce papier, nous établissons un nouveau lien entre ces domaines dans un cadre statique : nous montrons comment un CP-jeu booléen peut être utilisé pour calculer des extensions en argumentation, et donnons des algorithmes formels pour le faire.

**Mots-clés :** Argumentation, théorie des jeux

**Abstract:**
There already exist some links between argumentation and game theory. For instance, dynamic games can be used for simulating interactions between agents in an argumentation process. In this paper, we establish a new link between these domains in a static framework : we show how CP-Boolean games can be used for computing extensions of argumentation semantics, and give formal algorithms to do so.

**Keywords:** Argumentation, game theory

## 1 Introduction

Depuis quelques années, l'argumentation est devenue une approche incontournable pour résoudre des problèmes en intelligence artificielle ; elle peut servir, entre autres, pour faire du raisonnement par défaut ou pour simuler certaines formes de dialogue entre agents (voir par exemple [21, 22, 4, 1, 10]).
Dans ce cadre, l'argumentation correspond à l'étude de l'échange d'arguments entre agents en interaction, cette interaction prenant habituellement la forme d'un conflit (appelé "attaque"). Si on se place par exemple dans un cadre logique, un argument peut alors être vu comme une paire ⟨ensemble d'hypothèses, conclusion⟩, où l'ensemble des hypothèses supporte la conclusion selon certains schémas d'inférence logique, et il y aura conflit si, par exemple, la conclusion d'un argument contredit l'hypothèse d'un autre argument.

La principale difficulté pour toute théorie argumentative est la sélection d'un ensemble acceptable d'arguments, basée sur les interactions entre arguments. Intuitivement, un ensemble acceptable d'arguments doit être dans un certain sens "cohérent" (par exemple ne contenant pas d'arguments en conflit) et "assez fort" (capable par exemple de se défendre contre tous les arguments qui l'attaquent). Il est plus facile d'étudier cette notion d'acceptabilité en utilisant des systèmes d'argumentation abstraits (SA), comme ceux proposés par Dung ([16]) ; dans ces systèmes, la notion d'abstraction repose sur le fait que les arguments et leurs interactions sont donnés respectivement sous la forme d'un ensemble d'arguments et d'une relation binaire sur cet ensemble représentant l'attaque (on ne se préoccupe donc ni de la nature des arguments, ni de la manière dont les interactions sont construites). Notons toutefois que, même dans le cadre abstrait de Dung, la complexité des problèmes associés reste prohibitif dans le cas général (par exemple, "vérifier si un ensemble d'arguments est une extension préférée" est un problème coNP-complet).

D'autre part, la théorie des jeux tente d'analyser formellement les interactions stratégiques entre agents. Intuitivement, un jeu non coopératif est composé d'un ensemble d'agents (ou joueurs), et, pour chaque agent, d'un ensemble de stratégies possibles et d'une fonction d'utilité qui associe toutes les combinaisons possibles de stratégies à une valeur numérique (dans cet article, nous ne considérons que des jeux *statiques*, dans lesquels les agents choisissent leurs stratégies en parallèle, sans observer les choix des autres joueurs).
Un des principaux problèmes de cette approche est la difficulté d'exprimer efficacement[1] la

[1] Car le nombre de combinaisons possibles de stratégies est expo-

fonction d'utilité. Une solution à ce problème consiste à représenter les préférences des agents en utilisant un langage de représentation *structuré* et *compact*. Les jeux booléens permettent de répondre à ce problème : chaque agent contrôle un ensemble de variables *booléennes* (binaires), tandis qu'une formule propositionnelle permet de représenter ses préférences (voir [20, 19]). Ce cadre reste un peu limité (à cause de la dichotomie de préférences, une formule ne pouvant être que vraie ou fausse), mais il peut facilement être étendu en remplaçant les formules propositionnelles par des *CP-nets* (voir [5]). Il existe dans ce contexte quelques résultats de complexité intéressants (par exemple, "vérifier si un profil de stratégies est un équilibre de Nash en stratégies pures" est un problème polynomial lorsque tous les CP-nets sont acycliques).

Des liens entre argumentation et théorie des jeux ont déjà été identifiés ; par exemple, dans [16, 9], des jeux sont utilisés pour définir la théorie de la preuve et les algorithmes associés à certains problèmes d'acceptabilité en argumentation. Toutefois, les jeux considérés sont toujours dynamiques, et il n'y a pas de travaux spécifiques concernant les jeux statiques et l'argumentation. L'objectif premier de cet article est donc de proposer une traduction d'un système d'argumentation vers un jeu statique (un CP-jeu booléen) ; cela nous permettra, dans un second temps, de calculer les extensions préférées en utilisant les concepts bien connus de la théorie des jeux (équilibres de Nash en stratégies pures : PNE). Ainsi nous établirons un nouveau lien entre argumentation et jeux, tout en fournissant une nouvelle méthode de calcul des extensions préférées (même si cette nouvelle méthode n'est pas plus efficace que les algorithmes existants – voir par exemple [14, 17, 18, 9]).

Cet article est organisé comme suit : le cadre d'argumentation de Dung et les CP-jeux booléens sont présentés brièvement en section 2 et section 3. La section 4 présente le noyau de ce travail : comment traduire un système d'argumentation en un CP-jeu booléen, et comment utiliser ce jeu pour calculer les extensions de la sémantique argumentative. Quelques travaux connexes sont ensuite présentés en section 5, puis la conclusion est donnée en section 6.

nentiel en nombre de joueurs et en nombre de variables contrôlées par chaque joueur.

## 2 Systèmes d'argumentation (SA)

Dung a proposé dans [16] un cadre abstrait pour l'argumentation, dans lequel il s'intéresse principalement aux définitions des statuts des arguments. Pour cela, il suppose qu'un ensemble d'arguments est donné, ainsi que différents conflits entre ces derniers. Nous rappelons brièvement ce cadre :

**Définition 1** *Un* système d'argumentation (SA) *est une paire* $\langle \mathcal{A}, \mathcal{R} \rangle$ *composée d'un ensemble d'arguments* $\mathcal{A}$ *et d'une relation binaire* $\mathcal{R}$ *sur* $\mathcal{A}$ *appelée* relation d'attaque. $\forall a_i, a_j \in \mathcal{A}$, $a_i \mathcal{R} a_j$ *signifie que* $a_i$ attaque $a_j$ *(ou que* $a_j$ est attaqué par $a_i$*). Un SA peut être représenté par un graphe orienté appelé le* graphe d'interaction*, dont les nœuds sont des arguments, et les arcs représentent les relations d'attaque.*

Dans le cadre de Dung, l'*acceptabilité d'un argument* dépend de son appartenance à des ensembles appelés extensions qui doivent vérifier certaines propriétés caractéristiques représentant l'acceptabilité collective. Soit $\mathsf{SA} = \langle \mathcal{A}, \mathcal{R} \rangle$, un système d'argumentation, et $S \subseteq \mathcal{A}$ un ensemble d'arguments. Les principales propriétés caractéristiques sont :

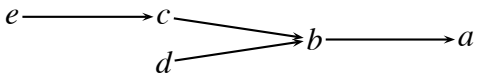
**Définition 2** *S est* sans conflit *pour* SA *si et seulement s'il n'existe pas de* $a_i$, $a_j$ *dans S tels que* $a_i \mathcal{R} a_j$.
*Un argument a est* acceptable pour *S dans* SA *ssi* $\forall b \in \mathcal{A}$ *tel que* $b \mathcal{R} a$, $\exists c \in S$ *tel que* $c \mathcal{R} b$. *S est* acceptable *pour* SA *ssi* $\forall a \in S$, *a est acceptable pour S dans* SA.

Plusieurs *sémantiques pour l'acceptabilité* ont alors été définies dans [16]. Par exemple :

**Définition 3** *S est un* ensemble admissible *de* SA *ssi S est sans conflit et acceptable pour* SA.
*S est une* extension préférée *de* SA *ssi S est* $\subseteq$*-maximal parmi les ensembles admissibles de* SA.
*S est une* extension stable *de* SA *ssi S est sans conflit et S attaque tout argument qui n'appartient pas à S.*

Ces notions sont illustrées sur les exemples suivants.

**Exemple 1** *Soit* $\mathsf{SA} = \langle \{a,b,c,d,e\}, \{(b,a), (c,b), (d,b), (e,c)\} \rangle$ *représenté par :*




*$\{e,c\}$ n'est pas sans conflit ; b n'est pas acceptable pour $\{e\}$ ; a est acceptable pour $\{e,d\}$ ; $\{d,a\}$ est un ensemble admissible et $\{e,d,a\}$ est l'extension stable et préférée de* SA.

Nous aurons besoin des propriétés suivantes ([16, 17, 18, 15]) :

**Proposition 1** *Soit* $\mathsf{SA} = \langle \mathcal{A}, \mathcal{R} \rangle$ *tel que* $\mathcal{A} \neq \varnothing$.

1. *Chaque argument non attaqué appartient à toute extension préférée de* SA *(voir [16]).*
2. *Un système d'argumentation* SA *acyclique contient seulement une extension préférée (voir [17, 18]).*
3. *Si $\varnothing$ est l'unique extension préférée, alors* SA *contient au moins un circuit de longueur impaire (voir [17, 18, 15]).*
4. *Si* SA *ne contient aucun circuit de longueur impaire, alors ses extensions préférées ne sont pas vides (voir [17, 18, 15]).*
5. *Si* SA *ne contient aucun circuit de longueur impaire, alors chaque extension préférée est également stable*[2] *(voir [17, 18, 15]).*

De nombreux travaux dans le domaine de l'argumentation utilisent l'hypothèse simplificatrice suivante : les systèmes d'argumentation ne contiennent pas de circuits de longueur impaire, de tels circuits étant considérés comme des paradoxes (car ils généralisent des situations dans lesquelles un argument s'attaque lui-même). Nous adopterons ici ce point de vue[3] en considérant que *les systèmes d'argumentation traduits ne doivent pas contenir de circuits de longueur impaire*[4] ; par contre, les systèmes d'argumentation contenant des circuits de longueur paire seront pris en compte.
Notons que, dans le cas où il n'existe pas de circuits de longueur impaire, le système d'argumentation présente des propriétés particulièrement intéressantes :

**Conséquence 1** *Soit* $\mathsf{SA} = \langle \mathcal{A}, \mathcal{R} \rangle$ *un SA sans circuit de longueur impaire. Soit E une extension préférée de* SA*, et soit a un argument de* SA*. Si E ne contient pas d'attaquant de a, alors* $a \in E$.

[2] Le système d'argumentation est alors dit "cohérent" (voir [16]).

[3] Même si certains circuits de longueur impaire peuvent avoir du sens.

[4] Et, si ce n'est pas le cas, ces circuits de longueur impaire seront supprimés du système d'argumentation avant traduction.

*Preuve :* Nous pouvons appliquer la proposition 1.5 issue de [17, 18, 15]. $E$ est donc une extension stable. Donc, comme $E$ est stable, si on suppose que $a \notin E$ alors $\exists b \in E$ tel que $b \mathcal{R} a$. Mais ceci est impossible car il n'existe pas d'attaquant de $a$ dans $E$. Donc $a \in E$. ■

# 3 CP-jeux booléens

Nous allons tout d'abord introduire quelques notations qui nous seront utiles. Soit $V$ un ensemble fini de variables propositionnelles et $L_V$ le langage propositionnel construit à partir de $V$, des connecteurs usuels et des constantes booléennes $\top$ (*vrai*) et $\bot$ (*faux*). Les formules de $L_V$ sont dénotées par $\varphi, \psi$, etc. $2^V$ est l'ensemble des interprétations pour $V$, avec la convention usuelle établissant que pour $M \in 2^V$ et $x \in V$, $M$ donne la valeur *vrai* à $x$ si $x \in M$ et *faux* sinon. Soit $X \subseteq V$. $2^X$ est l'ensemble des *$X$-interprétations*. Les $X$-interprétations sont dénotées par la liste de toutes les variables de $X$, associées au symbole $^-$ lorsque la variable est mise à faux. Par exemple, si $X = \{a,b,d\}$, la $X$-interprétation $M = \{a,d\}$ est notée $a\bar{b}d$. Une *relation de préférences* $\succeq$ est une relation binaire (non nécessairement complète) réflexive et transitive sur $2^V$. Soit $M, M' \in 2^V$. La relation de préférences stricte $\succ$ associée à $\succeq$ est définie par $M \succ M'$ ssi $M \succeq M'$ et non $M' \succeq M$.

## 3.1 CP-nets

Nous étudions dans cette section un langage compact de représentation de préférences sur des domaines combinatoires très populaire : les CP-nets. Ce modèle graphique exploite l'indépendance préférentielle conditionnelle afin de structurer les préférences d'un agent sous l'hypothèse *ceteris paribus* [6, 7]. Si les CP-nets sont généralement construits à partir de variables ayant un domaine arbitraire fini, nous allons ici utiliser des CP-nets "propositionnels", c'est-à-dire des CP-nets avec des variables binaires.

**Définition 4** *Soit $V$ un ensemble de variables propositionnelles et $\{X,Y,Z\}$ une partition de $V$. $X$ est* conditionnellement préférentiellement indépendant *de $Y$ étant donné $Z$ ssi $\forall z \in 2^Z$, $\forall x_1, x_2 \in 2^X$ et $\forall y_1, y_2 \in 2^Y$ nous avons : $x_1 y_1 z \succeq x_2 y_1 z$ ssi $x_1 y_2 z \succeq x_2 y_2 z$.*

Pour chaque variable $X$, l'agent spécifie un ensemble de *variables parents* $Pa(X)$ qui peuvent affecter ses préférences sur les valeurs de $X$. Formellement, $X$ et $V \setminus (\{X\} \cup Pa(X))$ sont conditionnellement préférentiellement indépendants étant donné $Pa(X)$. Cette information est utilisée pour créer le CP-net :

**Définition 5** *Soit $V$ un ensemble de variables propositionnelles. $\mathcal{N} = \langle \mathcal{G}, \mathcal{T} \rangle$ est un* CP-net sur $V$*, où $\mathcal{G}$ est un graphe orienté sur $V$, et $\mathcal{T}$ est un ensemble de tables de préférences conditionnelles $CPT(X_j)$ pour chaque $X_j \in V$. Chaque $CPT(X_j)$ associe un ordre linéaire $\succ_p^j$ à chaque instanciation $p \in 2^{Pa(X_j)}$.*

Les informations représentées par un CP-net $\mathcal{N}$ peuvent être vues comme un ensemble d'assertions logiques sur les préférences d'un agent concernant un ensemble complet d'assignations de variables dans un graphe. Ces préférences ne sont pas totales : elles ne permettent pas en général de déterminer une unique relation de préférences.

**Exemple 2** *Soit le CP-net donné en Figure 1 à propos de préférences sur un dîner. Les variables S et V correspondent respectivement à la soupe et le vin. Je préfère strictement manger de la soupe de poisson ($S_p$) plutôt que de la soupe de légumes ($S_l$), tandis que mes préférences sur le vin dépendent de la soupe que je mange : je préfère du vin rouge ($V_r$) avec la soupe de légumes ($S_l : V_r \succ V_b$), et du vin blanc ($V_b$) avec la soupe de poisson ($S_p : V_b \succ V_r$). Donc, $D(S) = \{S_p, S_l\}$ et $D(V) = \{V_r, V_b\}$.*

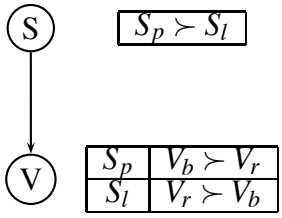



FIG. 1 – CP-net "Dîner"

*La figure 2 représente la relation de préférences induite par ce CP-net. L'élément du bas ($S_l \wedge V_b$) est le pire cas, tandis que l'élément du haut ($S_p \wedge V_b$) est le meilleur cas.*

*Il y a un arc entre les nœuds ($S_p \wedge V_b$) et ($S_l \wedge V_b$) car il est possible de comparer ces états toutes choses étant égales par ailleurs.*

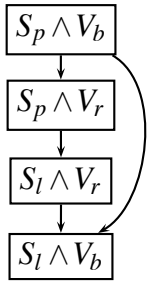



FIG. 2 – Graphe de préférences induit par le CP-net "Dîner"

*Nous pouvons donc complètement ordonner les états possibles (du préféré au moins bon) :*

$$(S_p \wedge V_b) \succ (S_p \wedge V_r) \succ (S_l \wedge V_r) \succ (S_l \wedge V_b)$$

*Cette relation $\succ$ est la seule qui satisfasse ce CP-net.*

### 3.2 CP-jeux booléens

Les jeux booléens [20, 19] permettent de représenter de façon compacte des jeux statiques à 2 joueurs ayant des préférences binaires. Dans [5], les jeux booléens sont généralisés avec des préférences non dichotomiques : ils sont couplés avec des CP-nets propositionnels.

**Définition 6** *Un* CP-jeu booléen *est un 4-tuple $G = (N, V, \pi, \Phi)$, où $N$ est un ensemble de joueurs, $V$ est un ensemble de variables propositionnelles, $\pi : N \mapsto V$ est une fonction d'affectation de contrôle qui définit une partition de $V$, et $\Phi = \langle \mathcal{N}_1, \ldots, \mathcal{N}_n \rangle$, chaque $\mathcal{N}_i$ étant un CP-net sur $V$ dont le graphe est dénoté par $\mathcal{G}_i$, et $\forall i \in N$, $\succeq_i = \succeq_{\mathcal{N}_i}$.*

La fonction d'affectation de contrôle $\pi$ associe à chaque joueur les variables qu'il contrôle, chaque variable étant contrôlée par un et un seul agent,[5] *i.e.*, $\{\pi_1, \ldots, \pi_n\}$ forme une partition de $V$.

**Définition 7** *Soit $G = (N, V\pi, \Phi)$ un CP-jeu booléen. Une* stratégie *$s_i$ pour le joueur $i$ est une $\pi_i$-interprétation. Un* profil de stratégies *$s$ est un $n$-tuple $s = (s_1, \ldots, s_n)$ où pour chaque $i$, $s_i \in 2^{\pi_i}$.*

En d'autres mots, une stratégie pour $i$ est une assignation de valeurs de vérité à chaque variable

[5] L'ensemble de toutes les variables contrôlées par $i$ sera noté $\pi_i$ plutôt que $\pi(i)$.

contrôlée par $i$. Comme $\{\pi_1,\ldots,\pi_n\}$ forme une partition de $V$, un profil de stratégies définit une interprétation (non ambiguë) pour $V$. Par abus de notation, nous noterons $s \in 2^V$ pour représenter la valeur assignée par $s$ à des variables.

Nous utiliserons dans la suite de ce papier les notations suivantes, qui sont classiques en théorie des jeux : soit $G=(N,V,\pi,\Phi)$ un CP-jeu booléen avec $N=\{1,\ldots,n\}$, et $s=(s_1,\ldots,s_n)$, $s'=(s'_1,\ldots,s'_n)$ deux profils de stratégies. $s_{-i}$ dénote la projection de $s$ sur $N\setminus\{i\}$ : $s_{-i}=(s_1,\ldots,s_{i-1},s_{i+1},\ldots,s_n)$ ; similairement, $\pi_{-i}$ dénote l'ensemble des variables contrôlées par tous les joueurs sauf $i$ : $\pi_{-i}=V\setminus\pi_i$ ; enfin, $(s'_i,s_{-i})$ représente le profil de stratégies obtenu à partir de $s$ en remplaçant $s_i$ par $s'_i$ sans modifier les autres stratégies : $(s'_i,s_{-i})=(s_1,\ldots,s_{i-1},s'_i,s_{i+1},\ldots,s_n)$.

Un équilibre de Nash en stratégies pures (PNE) est un profil de stratégies tel que la stratégie de chaque joueur est une réponse optimale aux stratégies des autres joueurs. Les PNEs sont classiquement définis pour des jeux dans lesquels les préférences sont complètes, ce qui n'est pas nécessairement le cas ici. Nous devons donc introduire la notion de PNE *fort*.

**Définition 8** *Soit $G=(N,V,\pi,\Phi)$ et $Pref_G=\langle\succeq_1,\ldots,\succeq_n\rangle$ la collection des relations de préférences sur $2^V$ induite par $\Phi$. Soit $s=(s_1,\ldots,s_n)\in 2^V$. $s$ est un* PNE fort *(SPNE) pour $G$ ssi $\forall i\in\{1,\ldots,n\}$, $\forall s'_i\in 2^{\pi_i}$, $(s'_i,s_{-i})\preceq_i(s_i,s_{-i})$.*

La proposition suivante a été prouvée dans [5] :

**Proposition 2** *Soit $G=(N,V,\pi,\Phi)$ un CP-jeu booléen tel que les graphes $\mathcal{G}_i$ sont tous identiques ($\forall i,j\in N$, $\mathcal{G}_i=\mathcal{G}_j$) et acycliques. $G$ a alors un et un seul PNE fort.*

La preuve de ce résultat utilise la *forward sweep* procédure [6] (cette procédure consiste à instancier les variables selon un ordre compatible avec le graphe, en choisissant pour chaque variable sa valeur préférée en fonction de l'instanciation de ses parents). De plus, il a été montré dans [5] que ce SPNE peut être construit en temps polynomial.

# 4 Argumentation et CP-jeux booléens

Notre objectif ici est de transformer un SA en un CP-jeu booléen $G$, puis d'utiliser les outils bien connus de la théorie des jeux, et plus particulièrement les propriétés spécifiques des CP-jeux booléens, afin de trouver les extensions préférées du SA. Ce travail nous permet de créer de nouveaux liens entre jeux et argumentation.

## 4.1 Traduction d'un système d'argumentation en un CP-jeu booléen

Cette transformation est faite par l'algorithme 1. Ce dernier suppose l'existence de deux autres algorithmes :

- EstCyclique qui retourne *vrai* s'il existe au moins un circuit dans le graphe d'argumentation[6],
- SupCircuitsImp pour effacer tous les circuits de longueur impaire s'il en existe dans le système d'argumentation[7].

L'exécution de ces deux algorithmes peut être vue comme une étape de pré-compilation de l'algorithme 1. Comme l'algorithme EstCyclique ne détecte pas directement les circuits de longueur impaire, il peut sembler inutile dans la pré-compilation de l'algorithme 1. Pourtant, comme EstCyclique est un algorithme linéaire tandis que SupCircuitsImp est polynomial, nous pensons qu'il est intéressant d'éviter une exécution inutile de SupCircuitsImp lorsque SA est acyclique.

[6] Cet algorithme est linéaire :

(Étape 1) effacer tous les nœuds qui n'ont pas de prédécesseurs ;

(Étape 2) recommencer l'étape 1 jusqu'à ce que, soit tous les nœuds restants aient au moins un prédécesseur (le graphe initial contient un circuit), soit le graphe est vide (donc le graphe initial ne contient pas de circuit).

[7] Cet algorithme est polynomial :

(Étape 1) calcul de la matrice d'adjacence booléenne correspondant à tous les chemins d'attaque de longueur impaire minimaux ; il est suffisant de prendre la matrice d'adjacence booléenne $\mathcal{M}$ ($\mathcal{M}(i,j)=1$ s'il existe un arc de $i$ à $j$ dans SA) et de calculer $\mathcal{M}^{\text{olc}}=\mathcal{M}^1+\mathcal{M}^3+\ldots+\mathcal{M}^{2n-1}$ avec $n=|\mathcal{A}|$ (la limite $2n-1$ est obtenue en utilisant un résultat général de la théorie des graphes : si un graphe orienté contient un chemin de $a$ vers $b$, il existe alors un chemin simple – un chemin dans lequel chaque nœud n'apparaît qu'une seule fois – de $a$ vers $b$) ;

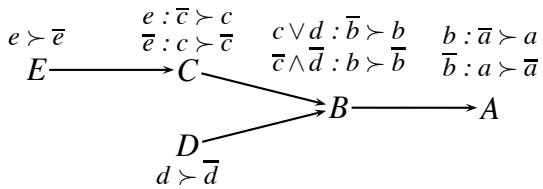
(Étape 2) effacer tous les arguments pour lesquels l'élément diagonal de $\mathcal{M}^{\text{olc}}$ vaut 1 ;

(Étape 3) effacer tous les arcs ayant un argument éliminé en point de départ ou d'arrivée.

Soit SA un système d'argumentation qui ne contient pas de circuit de longueur impaire. Les principes de l'algorithme 1 sont les suivants :
- chaque argument de SA est une variable de $G$ ;
- chaque variable est contrôlée par un joueur différent (nous avons donc autant de joueurs que de variables) ;
- les CP-nets de chaque joueur sont définis de façon identique :
  - le graphe du CP-net est exactement le graphe orienté du SA ;
  - les préférences sur chaque variable $v$ qui n'est pas attaquée sont $v \succ \overline{v}$ (si un argument n'est pas attaqué, nous voulons le protéger ; la valeur *vrai* de la variable $v$ est donc préférée à sa valeur *faux*),
  - les préférences sur chaque variable $v$ qui est attaquée par un ensemble de variables $\mathcal{R}^{-1}(v)$ dépend de ces variables : si au moins une variable $w \in \mathcal{R}^{-1}(v)$ est satisfaite, $v$ ne peut pas être satisfaite (nous avons donc $\bigvee_{w\in\mathcal{R}^{-1}(v)} w : \overline{v} \succ v$[8]) ; par contre, si aucune variable $w \in \mathcal{R}^{-1}(v)$ n'est satisfaite, $v$ peut être satisfaite (et donc $\bigwedge_{w\in\mathcal{R}^{-1}(v)} \overline{w} : v \succ \overline{v}$).

Le CP-jeu booléen $G$ est construit en temps polynomial à partir d'un système d'argumentation SA (même si SA est cyclique et que nous ayons d'abord à effacer les circuits de longueur impaire).

**Exemple 3** *Soit* $\mathsf{SA} = \langle\{a,b,c,d,e\},\ \{(b,a),\ (c,b),\ (d,b),(e,c)\}\rangle$ *(*SA *est acyclique). Sa traduction en un CP-jeu booléen* $G = (N,V,\pi,\Phi)$ *s'obtient en appliquant l'algorithme 1 :* $V = \{a,b,c,d,e\}$ *et* $N = \{1,2,3,4,5\}$, *avec* $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\pi_4 = \{d\}$ *et* $\pi_5 = \{e\}$. *Le CP-net suivant représente les préférences de tous les joueurs*[9] *:*




**Exemple 4** *Soit* $\mathsf{SA} = \langle\{a,b\},\{(a,b),\ (b,a)\}\rangle$. *On obtient en appliquant l'algorithme 1 :* $V = \{a,b\}$ *et* $N = \{1,2\}$, *avec* $\pi_1 = \{a\}$, $\pi_2 = \{b\}$

[8] La formule $w : \overline{v} \succ v$ (resp. $\overline{w} : \overline{v} \succ v$) signifie que pour la valeur *vrai* (resp. *faux*) de la variable $w$, la valeur *faux* de la variable $v$ est préférée à la valeur *vrai*.

[9] Afin de différencier le CP-net du SA, nous représentons les nœuds du CP-net en majuscule, et ceux du SA en minuscule.

**Algorithme 1** : Traduction d'un système d'argumentation en CP-jeu booléen

**début**

/* ENTREE : $\mathsf{SA} = \langle\mathcal{A},\mathcal{R}\rangle$ système d'argumentation */
/* SORTIES : $G = (N,V,\pi,\Phi)$ CP-jeu booléen, SA après suppression de tous les circuits impairs */
/* VARIABLES LOCALES : $i$ = agent courant, $a$ = argument courant */

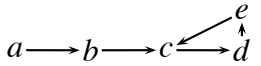
/* suppression de tous les circuits impairs */
**si** ESTCYCLIQUE(SA) **alors**
  SA = SUPCIRCUITSIMP(SA)
/* SA ne contient plus de circuits impairs*/
/* calcul des CPTs de chaque argument */
**pour** $a \in \mathcal{A}$ **faire**
  **si** $\mathcal{R}^{-1}(a) = \varnothing$ **alors** $CPT(a) = a \succ \overline{a}$ /* argument non attaqué */
  **sinon** /* cas de tous les autres arguments */
    $CPT(a) = \{\bigvee_{v\in\mathcal{R}^{-1}(a)} v : \overline{a} \succ a\}$
    $\cup\ \{\bigwedge_{v\in\mathcal{R}^{-1}(a)} \overline{v} : a \succ \overline{a}\}$
/* calcul du CP-net $\mathcal{N}$ */
$\mathcal{N} = \langle \mathsf{SA}, \cup_{a\in\mathcal{A}} CPT(a)\rangle$ /* graphe d'attaque */
/* après suppression de tous les circuits impairs, */
/* associé au CPT de chaque argument */
/* calcul de $N$, $V$, $\pi$ et $\Phi$ */
$i = 1$
$N = \varnothing$
$V = \mathcal{A}$ /* chaque argument est une variable */
**pour** $a \in \mathcal{A}$ **faire**
  $N = N \cup \{i\}$ /* un agent pour chaque argument */
  $\pi_i = \{a\}$ /* $i$ contrôle seulement cet argument */
  $\mathcal{N}_i = \mathcal{N}$ /* le même CP-net pour chaque agent */
  $i = i + 1$
**retourner** $(G = (N,V,\pi,\langle\mathcal{N}_1,\ldots,\mathcal{N}_{|V|}\rangle)$, SA)

**fin**

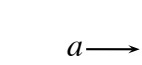
*(*SA *est cyclique mais ne contient que des circuits pairs). Le CP-net suivant représente les préférences de tous les joueurs.*

**Exemple 5** *Soit* $\mathsf{SA} = \langle\{a,b,c,d,e\},\ \{(a,b),\ (b,c),\ (c,d),\ (d,e),\ (e,c)\}\rangle$. *Le* SA *initial contient un circuit de longueur impaire, qui doit donc être supprimé. Le* SA *final ne contient que a et b.*

*Graphe initial pour le* SA




*Graphe final pour le* SA




*En appliquant l'algorithme 1, on obtient donc* $V = \{a,b\}$, $N = \{1,2\}$, *avec* $\pi_1 = \{a\}$, $\pi_2 = \{b\}$ *et le CP-net suivant qui représente les préférences de tous les joueurs :*

$$a \succ \overline{a}\ A \longrightarrow B \quad \begin{array}{l} a : \overline{b} \succ b \\ \overline{a} : b \succ \overline{b} \end{array}$$

**Proposition 3** *Soit* $\mathsf{SA} = \langle \mathcal{A}, \mathcal{R} \rangle$ *un système d'argumentation. Soit* $G = (N, V, \pi, \Phi)$ *le CP-jeu booléen et* $\mathsf{SA}'$ *le système d'argumentation obtenus en appliquant l'algorithme 1 sur* $\mathsf{SA}$.
*s est une extension préférée de* $\mathsf{SA}'$ *ssi s est un SPNE de*[10] *G.*

*Preuve :* Étudions tout d'abord la direction $\Rightarrow$.

Soit $s$ une extension préférée de $\mathsf{SA}'$. Supposons que $s$ n'est pas un SPNE du CP-jeu booléen associé. Donc, $\exists i \in N$, $\exists s'_i \in 2^{\pi_i}$, $\exists s_{-i} \in 2^{\pi_{-i}}$, tels que $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$. Soit $x_i$ la variable de $V$ telle que $\pi_i = \{x_i\}$ ($x_i$ est également un argument de $\mathsf{SA}'$). Plusieurs cas sont possibles :

- $\mathcal{R}^{-1}(x_i) = \varnothing$ ($x_i$ n'est pas attaquée). D'après l'algorithme 1 nous avons $CPT(x_i) = x_i \succ \overline{x}_i$. Comme $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$, nous savons que $s_i = \overline{x}_i$ ($x_i \notin s$). Mais d'après la proposition 1.1, si $\mathcal{R}^{-1}(x_i) = \varnothing$ alors $x_i \in s$. Nous avons une contradiction.
- $\mathcal{R}^{-1}(x_i) \neq \varnothing$ ($x_i$ a au moins un attaquant). D'après l'algorithme 1, nous avons $CPT(x_i) = \{\bigvee_{w \in \mathcal{R}^{-1}(x_i)} w : \overline{x}_i \succ x_i\} \cup \{\bigwedge_{w \in \mathcal{R}^{-1}(x_i)} \overline{w} : x_i \succ \overline{x}_i\}$. Deux cas sont possibles :
  - $\forall x_j$ tel que $x_j \mathcal{R} x_i$, $x_j \notin s$. Donc, $\bigwedge_{w \in \mathcal{R}^{-1}(x_i)} \overline{w}$ est vrai, et on peut déduire de $CPT(x_i)$ que $x_i \succ \overline{x}_i$. Comme $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$, on a $s_i = \overline{x}_i$ et $s'_i = x_i$. $x_i$ n'est donc pas dans $s$. Ceci est en contradiction avec la conclusion obtenue en appliquant la conséquence 1 qui dit que $x_i \in s$ ($\mathsf{SA}'$ est un système d'argumentation sans circuits impairs, et $x_i$ n'a pas d'attaquant dans l'extension préférée de $s$). Ce cas est donc impossible.
  - Au moins un argument $x_j$ de $\mathcal{R}^{-1}(x_i)$ appartient à $s$. $\bigvee_{w \in \mathcal{R}^{-1}(x_i)} w$ est donc vrai, et on peut déduire $\overline{x}_i \succ x_i$ en utilisant $CPT(x_i)$. Comme $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$, on a $s_i = x_i$. Mais ceci est en contradiction avec le fait que $s$ doit être sans conflit ($x_j \in s$ et $x_i \in s$). Ce cas est donc également impossible.

  Aucun cas n'est donc possible si nous supposons que $s$ n'est pas un SPNE. $s$ est donc bien un SPNE.

Étudions à présent la direction $\Leftarrow$.

Soit $s = (s_1, \ldots, s_n)$ un SPNE de $G$. Nous devons à présent montrer que $s$ est une extension préférée de $\mathsf{SA}'$. Supposons plutôt qu'elle n'en est pas une. Soit $x_i \in s$ une variable telle que (1) ou (2) soient vérifiées avec (1) : $\exists x_j \in s$ tel que $x_i \mathcal{R} x_j$ ou $x_j \mathcal{R} x_i$ ($s$ n'est pas sans conflit) et (2) : $\exists x_j \in \mathcal{A}$ tel que $x_j \mathcal{R} x_i$ et $\nexists x_k \in s$ telle que $x_k \mathcal{R} x_j$ ($s$ n'est pas acceptable).

- $s$ n'est pas sans conflit (1) :
  - $\exists x_j \in s$ tel que $x_i \mathcal{R} x_j$. Comme $x_i \in s$, nous savons que $\bigvee_{w \in \mathcal{R}^{-1}(x_j)} w$, et, en utilisant $CPT(x_j)$, nous pouvons déduire que $\overline{x}_j \succ x_j$. Comme nous savons que $s$ est un SPNE ; pour chaque joueur $j$ qui contrôle $x_j$, nous avons $\forall s'_j$, $\forall s_{-j}$, $(s_j, s_{-j}) \succ_j (s'_j, s_{-j})$. Donc, $s_j = \overline{x}_j$ et $x_j \notin s$, ce qui est une contradiction.
  - $\exists x_j \in s$ tel que $x_j \mathcal{R} x_i$. Comme $x_i \in s$ et que $s$ est un SPNE, nous savons que $x_i \succ \overline{x}_i$ ; donc, en utilisant $CPT(x_i)$, nous pouvons en déduire $\bigwedge_{w \in \mathcal{R}^{-1}(x_i)} \overline{w}$. Donc, $x_j \notin s$, ce qui est une contradiction.
- $s$ n'est pas acceptable (2) : $\exists x_j \in \mathcal{A}$ tel que $x_j \mathcal{R} x_i$ et $\nexists x_k \in s$ tel que $x_k \mathcal{R} x_j$. Comme $x_i \in s$ et $s$ est un SPNE, nous savons que $x_i \succ \overline{x}_i$. En utilisant $CPT(x_i)$, nous pouvons en déduire $\bigwedge_{w \in \mathcal{R}^{-1}(x_i)} \overline{w}$. Comme $x_j \mathcal{R} x_i$, nous savons que $x_j \notin s$. Donc, $\overline{x}_j \succ x_j$ et, en utilisant $CPT(x_j)$, nous pouvons en déduire $\bigvee_{w \in \mathcal{R}^{-1}(x_j)} w$. Cela signifie qu'il existe $x_k \in s$ tel que $x_k \in \mathcal{R}^{-1}(x_j)$, ce qui est une contradiction.

Aucun cas n'est donc possible si nous supposons que $s$ n'est pas une extension préférée. $s$ est donc une extension préférée. ∎

**Exemple 3 (suite):** $G$ a un SPNE $\{ed\overline{c}\overline{b}a\}$ et $\mathsf{SA}$ a seulement une extension préférée $\{e, d, a\}$.

**Exemple 4 (suite):** $G$ a deux SPNEs $\{a\overline{b}, \overline{a}b\}$ et $\mathsf{SA}$ a deux extensions préférées $\{a\}$, $\{b\}$.

**Exemple 5 (suite):** $G$ a un SPNE $\{a\overline{b}\}$ et le $\mathsf{SA}$ final (*après suppression des circuits impairs*) a une extension préférée $\{a\}$.

### 4.2 Calcul des extensions préférées

Comme les extensions préférées correspondent exactement aux SPNEs, les principales propriétés sur le calcul des SPNEs dans les CP-jeux booléens peuvent être appliquées. Le premier

[10] Rappelons que $s$ représente une $V$-interprétation, et donc si $s = a\overline{b}c$ par exemple, cela correspond à l'ensemble $\{a, c\}$.

cas intéressant concerne les systèmes d'argumentation acycliques :

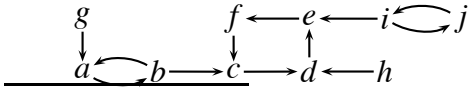
**Proposition 4** *Soit* SA *un système d'argumentation. Soit G le CP-jeu booléen et* SA′ *le système d'argumentation obtenus à partir de* SA *en appliquant l'algorithme 1.*
*Si* SA′ *est acyclique,* SA′ *a alors une et une seule extension préférée qui est calculable en temps polynomial à partir de G.*

*Preuve :* La transformation de SA en un CP-jeu booléen en appliquant l'algorithme 1 se fait en temps polynomial. Ensuite, le calcul du SPNE de ce jeu se fait en temps polynomial en utilisant la procédure forward sweep (proposition 2 donnée dans [5]), et enfin la proposition 3 montre que ce SPNE correspond exactement à l'extension préférée de SA′ (SA après suppression des circuits impairs). ■

Cette proposition est vraie pour le cas simple des systèmes d'argumentation acycliques. Le calcul de(s) SPNE(s) dans le cas des systèmes d'argumentation cycliques est beaucoup plus compliqué. Les algorithmes 2 et 3 permettent de calculer ces concepts de solution dans le cas où le système d'argumentation contient des circuits de longueur paire.

Ces algorithmes supposent l'existence de l'algorithme CALCCIRCUITINTPOURPROPAG qui renvoie le circuit (ou l'un des circuits s'il y en a plusieurs) qui permet d'atteindre le plus de variables possible dans un ensemble de variables donné[11].

Par exemple, sur le graphe suivant :



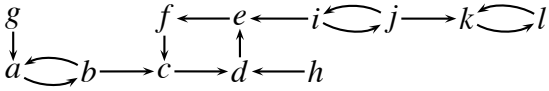

[11] Cet algorithme utilise la matrice booléenne d'adjacence comme l'algorithme SUPCIRCUITSIMP :
- calcul de la matrice booléenne d'adjacence $\mathcal{M}^{\text{ap}}$ qui correspond à tous les chemins minimaux dans le graphe réduit à un ensemble donné de variables $\mathcal{M}^{\text{ap}} = \mathcal{M} + \mathcal{M}^2 + \mathcal{M}^3 + \ldots + \mathcal{M}^{2n}$ avec $n = |V|$ ; $\mathcal{M}^{\text{ap}}(i)$ représente $(\mathcal{M}^{\text{ap}}(i,1), \ldots, \mathcal{M}^{\text{ap}}(i,n))$ ;
- `AVoir` $= V$ ; $C = \varnothing$ ; `fin`? $=$*faux* ;
- boucle : tant que NON(`fin`?) faire
  $v =$ `top(AVoir)` ; `AVoir` $=$ `AVoir` $\setminus \{v\}$ ;
  si ($\nexists w \in$ `AVoir` t.q.. $\mathcal{M}^{\text{ap}}(v) \subset \mathcal{M}^{\text{ap}}(w)$) alors
  /* pas de variable permettant d'atteindre plus de variables que $v$ */
  $C = C \cup \{v\}$ ;
  $\forall w \in$ `AVoir` faire si $\mathcal{M}^{\text{ap}}(v) = \mathcal{M}^{\text{ap}}(w)$ alors $C = C \cup \{w\}$ ;
  `fin`? $=$*vrai* ;
  sinon si `AVoir` est vide alors `fin`? $=$*vrai* ;
- Retourner $C$

$\{a,b\}$ permet d'atteindre les variables $a$, $b$, $c$, $d$, $e$, $f$, $\{i,j\}$ permet d'atteindre les variables $i$, $j$, $c$, $d$, $e$, $f$. Ces circuits sont plus intéressants que les autres pour propager les valeurs dans le graphe (si ce sont des points de départ pour un processus de propagation, cette propagation sera plus efficace).

Soit $\mathcal{N}$ le CP-net représentant les buts des joueurs d'un CP-jeu booléen. Les principes des algorithmes 2 et 3 sont :

1. instanciation de toutes les variables non attaquées (qui n'ont pas de parents dans $\mathcal{N}$ et sont satisfaites dans le SPNE) ;
2. propagation de ces instanciations autant que possible ;
3. une fois que toutes les instanciations possibles ont été effectuées, répéter les étapes suivantes :
   - si toutes les variables sont instanciées, le SPNE peut être retourné ;
   - sinon :
     - calculer le circuit le plus intéressant $C$ avec l'algorithme CALCCIRCUITINTPOURPROPAG (il y en a un sinon toutes les variables auraient été instanciées) ;
     - créer deux nouveaux SPNEs en utilisant l'état courant du SPNE auquel on ajoute pour le premier SPNE une variable de $C$ instanciée à *vrai*, pour l'autre SPNE cette même variable instanciée à $faux$ ;
     - propager ces instanciations pour chacun de ces SPNEs autant que possible ;
     - reboucler sur l'étape 3.

**Exemple 6** *Soit le graphe suivant :*




*Les étapes du processus de calcul sont :*
- *g et h sont instanciés à vrai (état courant du SPNE* $= gh$*) ;*
- *a et d sont instanciés à faux (état courant SPNE* $= gh\overline{a}\overline{d}$*) ;*
- *b est instancié à vrai (état courant du SPNE* $= gh\overline{a}\overline{d}b$*) ;*
- *c est instancié à faux (état courant du SPNE* $= gh\overline{a}\overline{d}b\overline{c}$*) ;*
- *la simple propagation s'arrête ici. Nous devons maintenant calculer les circuits intéressants dans l'ensemble de variables restant* $(e,f,i,j,k,l)$*, le résultat est* $(i,j)$ *;*

**Algorithme 2** : Calcul des SPNEs d'un CP-jeu booléen obtenu à partir d'un système d'argumentation

**début**
/* ENTREES : un CP-jeu booléen $G = (N, V, \pi, \Phi)$, où $\Phi = \langle \mathcal{N}_1, \ldots, \mathcal{N}_u \rangle$ */
/* SORTIES : un ensemble de SPNEs *SP* */
/* VARIABLES LOCALES : $v$ = variable courante, $In$ = (resp. $Out$ =) ensemble des variables instanciées à *vrai* (resp. *faux*), $R$ = ensemble des variables restant à instancier */

$In = \varnothing$, $Out = \varnothing$, $R = V$ /* Initialisation */
/* Instanciation de toutes les variables sans parents */
**pour** $v \in R$ **faire**
  **si** $Pa(v) = \varnothing$ **alors**
    $R = R \setminus \{v\}$
    $In = In \cup \{v\}$
/* propagation par récursion */
**retourner** CALCSPNEREC($G$, $R$, $In$, $Out$)
**fin**

**Algorithme 3** : CALCSPNEREC : Calcul récursif des SPNEs d'un CP-jeu booléen obtenu à partir d'un système d'argumentation

**début**
/* ENTREES : un CP-jeu booléen $G = (N, V, \pi, \Phi)$, $R$ = ensemble de variables restant à instancier, $In$ = ensemble de variables instanciées à *vrai*, $Out$ = ensemble de variables instanciées à *faux**/
/* SORTIES : ensemble de SPNEs *SP* */
/* VARIABLES LOCALES : $v$ = variable courante, $n$ = cardinal de $R$, $C$ = ensemble de variables formant un circuit*/

**si** $R = \varnothing$ **alors**
  /* variables toutes instanciées, SPNE trouvé */
  **retourner** $\{(In\overline{Out})\}$
**sinon**
  $n = |R|$ /* $n$ = nombre de variables restant à instancier */
  **pour** $v \in R$ **faire**
    /* simple processus de propagation */
    **si** $Pa(v) \subseteq Out$ **alors**
      /* parents tous instanciés à *faux* */
      $In = In \cup \{v\}$
      $R = R \setminus \{v\}$
    **sinon**
      **si** $(Pa(v) \cap In) \neq \varnothing$ **alors**
        /* existe parent(s) instancié à *vrai* */
        $Out = Out \cup \{v\}$
        $R = R \setminus \{v\}$
  **si** $n = |R|$ **alors**
    /* aucune variable instanciée dans l'instruction Pour */
    $C$ = CALCCIRCUITINTPOURPROPAG($G$, $R$)
    $v$ = TOP($C$)
    **return**(
      CALCSPNEREC($G$, $R \setminus \{v\}$, $In \cup \{v\}$, $Out$)
      $\cup$
      CALCSPNEREC($G$, $R \setminus \{v\}$, $In$, $Out \cup \{v\}$))
  **sinon**
    /* au moins une variable instanciée dans l'instruction Pour */
    **retourner** CALCSPNEREC($G$, $R$, $In$, $Out$)
**fin**

- *le processus de propagation repart avec deux SPNEs courants : $gh\overline{a}\overline{d}b\overline{c}i$ et $gh\overline{a}\overline{d}b\overline{c}\overline{i}$ ;*
- *pour le premier cas, la propagation simple donne $gh\overline{a}\overline{d}b\overline{c}i\overline{e}\overline{f}jk\overline{l}$, tandis que pour le second cas, nous obtenons $gh\overline{a}\overline{d}b\overline{c}\overline{i}e\overline{f}jk\overline{l}$. Il y a donc deux SPNEs qui correspondent aux deux extensions préférées $\{g,h,b,e,j,l\}$ et $\{g,h,b,f,i,k\}$.*

La proposition suivante montre que les algorithmes 2 et 3 permettent de calculer exactement l'ensemble des SPNEs du CP-jeu booléen.

**Proposition 5** *Soit G un CP-jeu booléen obtenu à partir de l'algorithme 1. Soit SP l'ensemble des profils de stratégies de G donné par les algorithmes 2 et 3. $s \in SP$ ssi $s$ est un SPNE de $G$.*

*Preuve :* Étudions tout d'abord la direction $\Rightarrow$.

Soit $s \in SP$. Supposons que $s$ n'est pas un SPNE du CP-jeu booléen associé. Donc, $\exists i \in N$, $\exists s'_i \in 2^{\pi_i}$, $\exists s_{-i} \in 2^{\pi_{-i}}$, tels que $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$.

Soit $x_i$ la variable de $V$ telle que $\pi_i = \{x_i\}$. Plusieurs cas sont possibles :
- $Pa(x_i) = \varnothing$. D'après l'algorithme 1, nous avons $CPT(x_i) = x_i \succ \overline{x}_i$. Comme $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$, nous savons que $s_i = \overline{x}_i$. Pourtant, nous savons d'après l'algorithme 2 que $x_i \in s$. Nous avons une contradiction.
- $Pa(x_i) \neq \varnothing$. Nous savons d'après l'algorithme 1 que $CPT(x_i) = \{\bigvee_{v \in Pa(x_i)} v : \overline{x}_i \succ x_i\} \cup \{\bigwedge_{v \in Pa(x_i)} \overline{v} : x_i \succ \overline{x}_i\}$. Deux cas sont possibles :
  - Aucune variable de $Pa(x_i)$ n'est satisfaite : $\bigwedge_{v \in Pa(x_i)} \overline{v}$. Dans ce cas, nous savons que $x_i \succ \overline{x}_i$. Donc, comme $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$, nous avons $s_i = \overline{x}_i$. Mais nous savons d'après l'algorithme 2 que si $Pa(x_i) \subseteq Out$, c'est-à-dire si aucune variable de $Pa(x_i)$ n'est satisfaite, alors nous avons $x_i \in In$, et donc $x_i \in s$, ce qui amène une contradiction.
  - Au moins une variable de $Pa(x_i)$ est sa-

tisfaite : $\bigvee_{v \in Pa(x_i)} v$. Dans ce cas, nous savons que $\overline{x}_i \succ x_i$. Donc, comme $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$, $s_i = x_i$. Mais nous savons d'après l'algorithme 2 que si $Pa(x_i) \cap In \neq \varnothing$, c'est-à-dire si au moins une variable de $Pa(x_i)$ est satisfaite, alors nous avons $x_i \in Out$, et donc $\overline{x}_i \in s$, ce qui amène une contradiction.

Étudions à présent à la direction $\Leftarrow$.

Soit $s = (s_1, \ldots, s_n)$ un SPNE de $G$. Nous devons montrer que $s \in SP$, c'est-à-dire que $\forall i$, si $s_i = x_i$, alors $x_i \in In$, sinon $(s_i = \overline{x}_i)$ $x_i \in Out$.

- $Pa(x_i) = \varnothing$. Nous savons d'après l'algorithme 1 que nous avons $CPT(x_i) = x_i \succ \overline{x}_i$. Comme $s$ est un SPNE, nous savons que $s_i = x_i$. De plus, nous savons d'après l'algorithme 2 que $x_i \in In$.
- $Pa(x_i) \neq \varnothing$. Nous savons d'après l'algorithme 1 que nous avons $CPT(x_i) = \{\bigvee_{v \in Pa(x_i)} v : \overline{x}_i \succ x_i\} \cup \{\bigwedge_{v \in Pa(x_i)} \overline{v} : x_i \succ \overline{x}_i\}$. Deux cas sont alors possibles :
  - Aucune variable de $Pa(x_i)$ n'est satisfaite. Nous avons alors $\bigwedge_{v \in Pa(x_i)} \overline{v}$ et nous savons dans ce cas que $x_i \succ \overline{x}_i$. Comme $s$ est un SPNE, nous savons que, dans ce cas, $s_i = x_i$. Nous savons aussi d'après l'algorithme 3 que si $Pa(x_i) \subseteq Out$, c'est-à-dire si aucune variable de $Pa(x_i)$ n'est satisfaite, alors nous avons $x_i \in In$.
  - Au moins une variable de $Pa(x_i)$ est satisfaite. Nous avons alors $\bigvee_{v \in Pa(x_i)} v$ et nous savons dans ce cas que $\overline{x}_i \succ x_i$. Comme $s$ est un SPNE, nous savons que $s_i = \overline{x}_i$. Nous savons également d'après l'algorithme 3 que si $Pa(x_i) \cap In \neq \varnothing$, c'est-à-dire si au moins une variable de $Pa(x_i)$ est satisfaite, alors nous avons $x_i \in Out$.

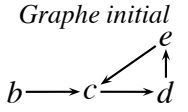
■

### 4.3 Gérer les circuits impairs

La suppression des circuits de longueur impaire a bien entendu une influence importance sur le calcul de(s) SPNE(s), et ce point peut être problématique si notre hypothèse initiale n'est pas acceptée : en général, les circuits de longueur impaire peuvent être considérés comme des paradoxes. Nous savons que certains circuits impairs ont un sens, en particulier lorsque ce ne sont pas des circuits impairs stricts[12]. Le travail présenté ici est préliminaire, et la suppression de ce genre de circuits permet de donner une traduction intéressante ayant des propriétés importantes. Nous sommes toutefois conscientes que le traitement des circuits impairs devra être fait (ce sera le sujet d'un prochain travail).

**Exemple 7** *Soit* SA $= \langle \{b,c,d,e\}, \{(b,c), (c,d), (d,e), (e,c)\} \rangle$. *Le* SA *initial est cyclique et contient un circuit de longueur impaire qui sera supprimé. Le* SA *final ne contient que b.*

*Graphe initial* *Graphe final*







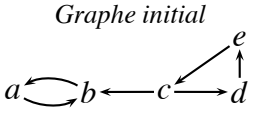
*Nous obtenons en appliquant l'algorithme 1* $V = \{b\}$, $N = \{1\}$, *avec* $\pi_1 = \{b\}$ *et le CP-net suivant qui représente les préférences du joueur :*

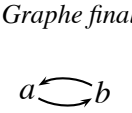
$$b \succ \overline{b} \quad B$$

*G a donc un SPNE* $\{b\}$ *qui correspond à l'extension préférée du* SA *final. Cependant, ce SPNE ne correspond pas à l'extension préférée du* SA *initial qui est l'ensemble* $\{b,d\}$. *Si nous supposons que les circuits de longueur impaire sont des paradoxes, et donc que leurs arguments ne sont pas significatifs, nous pouvons considérer que* $\{b\}$ *est une extension plus réaliste que* $\{b,d\}$[13].

**Exemple 8** *Soit* SA $= \langle \{a,b,c,d,e\}, \{(a,b), (b,a), (c,b), (c,d), (d,e), (e,c)\} \rangle$. *Le* SA *initial est cyclique, et il contient un circuit impair qui sera supprimé. Le* SA *final ne contiendra que les variables a et b.*

*Graphe initial* *Graphe final*




*Nous obtenons en appliquant l'algorithme 1* $V = \{a,b\}$, $N = \{1,2\}$, *avec* $\pi_1 = \{a\}$, $\pi_2 = \{b\}$ *et le CP-net suivant qui représente les préférences de tous les joueurs :*

$$\begin{array}{l} b : \overline{a} \succ a \\ \overline{b} : a \succ \overline{a} \end{array} \quad A \quad B \quad \begin{array}{l} a : \overline{b} \succ b \\ \overline{a} : b \succ \overline{b} \end{array}$$

[12] Par exemple quand on a en plus du circuit impair $a-b-c-a$, les attaques $(b,a)$, $(c,b)$ et $(a,c)$. Dans ce cas, il existe en "surimpression" du circuit impair, trois circuits pairs simples ($a-b-a$, $b-c-b$, $a-c-a$), un autre circuit impair simple ($a-c-b-a$) et plusieurs circuits pairs non simples ($a-c-a-b-a, \ldots$).

[13] Notons toutefois que cette manière de traiter cet exemple ne correspond pas aux principales sémantiques pour l'acceptabilité.

*G a donc deux SPNEs* $\{a\overline{b}, \overline{a}b\}$ *qui correspondent aux deux extensions préférées du* SA *final. Cependant, le* SA *initial a une seule extension préférée : l'ensemble* $\{a\}$*. Prendre en compte dans ce cas l'extension* $\{b\}$ *signifie que l'attaque* $c \rightarrow b$ *est considérée comme n'étant pas significative (car un circuit impair est un paradoxe, donc ses arguments ne sont pas significatifs, et donc ils ne peuvent pas fournir une attaque réelle contre un autre argument).*

## 5 Travaux connexes

Les principaux travaux connexes nous concernant traitent des liens entre argumentation et jeux identifiés par Dung. Ce lien apparaît déjà dans [16] ; un SA y est utilisé pour résoudre un jeu coopératif classique (le problème du mariage stable) ; Dung utilise les arguments du SA pour représenter les issues possible du jeu, et la relation d'attaque pour exprimer les conflits entres les différentes issues. Ce lien a également été utilisé dans [8] afin de démontrer l'acceptabilité d'un argument $a$, et cela à partir d'un jeu dynamique dans lequel un joueur est pour l'acceptabilité de $a$ et le second est contre. Les principales différences avec notre travail portent sur la nature du jeu (statique pour nous et dynamique ailleurs), puis sur le nombre et le rôle des joueurs.

Les seconds travaux connexes importants, qui ont déjà été mentionnés dans l'introduction, concernent le calcul des extensions préférées : des algorithmes ont déjà été proposés pour effectuer ce calcul (voir, par exemple [14, 17, 18, 9]). Et il est important de noter que nos algorithmes ne présentent pas de gain en efficacité par rapport aux algorithmes existants.

D'autres travaux connexes concernent l'utilisation d'un argument comme littéral dans une formule propositionnelle. Cette idée peut également être trouvée dans [13, 3, 12] (par exemple, dans [3], une caractérisation des extensions préférées est donnée sous la forme d'une formule propositionnelle).

Enfin, le dernier type de travaux connexes concerne le traitement des circuits de longueur impaire. Diverses approches existent dans la littérature : ces circuits peuvent apparaître dans un SA, mais sont interdits dans les extensions (voir par exemple [11]) ; ces circuits peuvent également être acceptés et traités comme des circuits de longueur paire pour le calcul des extensions (voir par exemple [2]). Notre approche correspond au premier cas : des circuits impairs peuvent apparaître dans un SA, mais ils seront supprimés pour le calcul des extensions (voir l'algorithme 1).

## 6 Conclusion

Nous avons montré comment traduire un système d'argumentation SA en un CP-jeu booléen, et comment ce jeu permet de calculer les extensions préférées du SA original en utilisant les équilibres de Nash en stratégies pures. Nous avons donné trois algorithmes formels permettant respectivement de transformer le SA en CP-jeu booléen, et de calculer les extensions préférées du SA. En outre, une fois les circuits impairs du SA éliminés, si le système d'argumentation SA′ est acyclique, alors les extensions préférées du SA′ sont calculables en temps polynomial.

Une limitation claire de nos résultats est que la traduction proposée élimine d'office les circuits de longueur impaire du système d'argumentation initial. Nous avons expliqué ce choix par le fait que notre travail est encore préliminaire, et que de tels systèmes d'argumentation ont d'importantes propriétés. Toutefois, il serait intéressant d'étudier des systèmes d'argumentation munis de circuits impairs, car certains de ces circuits peuvent avoir un sens. Une de nos pistes de travail est donc d'étudier si nos résultats s'appliquent toujours dans ce cas, et de modifier nos algorithmes en conséquence si ce n'est pas le cas.

## Références

[1] Leila Amgoud, Nicolas Maudet, and Simon Parsons. Modelling dialogues using argumentation. In *Proceedings of ICMAS'00*, pages 31–38, 2000.

[2] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-recursiveness : a general schema for argumentation semantics. *Artifical Intelligence*, 168 :162–210, 2005.

[3] Philippe Besnard and Sylvie Doutre. Characterization of semantics for argument systems. In *Proceedings of KR'04*, pages 183–193, 2004.

[4] Andrei Bondarenko, Phan Minh Dung, Robert A. Kowalski, and Francesca Toni. An abstract, argumentation-theoretic approach

to default reasoning. *Artificial Intelligence*, 93 :63–101, 1997.

[5] Elise Bonzon, Marie-Christine Lagasquie-Schiex, Jérôme Lang, and Bruno Zanuttini. Compact preference representation and Boolean games. *Journal of Autonomous Agents and Multi-Agent Systems*, 18(1) :1–35, 2009.

[6] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. CP-nets : A Tool for Representing and Reasoning with Conditional *Ceteris Paribus* Preference Statements. *Journal of Artificial Intelligence Research*, 21 :135–191, 2004.

[7] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. Preference-Based Constrained Optimization with CP-nets. *Computational Intelligence*, 20(2) :137–157, 2004.

[8] Claudette Cayrol, Sylvie Doutre, and Jérôme Mengin. Dialectical Proof Theories for the Credulous Preferred Semantics of Argumentation Frameworks. In *Proceedinfs of ECSQARU'01*, pages 668–679, 2001.

[9] Claudette Cayrol, Sylvie Doutre, and Jérôme Mengin. On Decision Problems Related to the Preferred Semantics for Argumentation Frameworks. *Journal of Logic and Computation*, 13(3) :377–403, 2003.

[10] Carlos I. Chesñevar, Ana G. Maguitman, and Ronald P. Loui. Logical models of argument. *ACM Computing surveys*, 32(4) :337–383, 2000.

[11] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Prudent semantics for argumentation frameworks. In *Proceedings of ICTAI'05*, pages 568–572, 2005.

[12] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Contrained argumentation frameworks. In *Proceedings of KR' 06*, pages 112–122, 2006.

[13] Nadia Creignou. The Class of Problems That are Linearly Equivalent to Satisfiability or a Uniform Method for Proving NP-Completeness. *Theorical Computer Sciences*, 145(1&2) :111–145, 1995.

[14] Sylvie Doutre, and Jérôme Mengin. Preferred Extensions of Argumentation Frameworks : Query Answering and Computation. In *Proceedings of IJCAR (LNAI 2083)*, pages 272-288, 2001.

[15] Sylvie Doutre. *Autour de la sémantique préférée des systèmes d'argumentation*. PhD thesis, Université Toulouse III, 2002.

[16] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-persons games. *Artificial Intelligence*, 77 :321–357, 1995.

[17] Paul E. Dunne and Trevor J.M. Bench-Capon. Complexity and combinatorial properties of argument systems. Technical report, University of Liverpool, Department of Computer Science (U.L.C.S.), 2001.

[18] Paul E. Dunne and Trevor J.M. Bench-Capon. Coherence in finite argument system. *Artificial Intelligence*, 141(1-2) :187–203, 2002.

[19] Paul Harrenstein. *Logic in Conflict*. PhD thesis, Utrecht University, 2004.

[20] Paul Harrenstein, Wiebe van der Hoek, John-Jules Meyer, and Cees Witteveen. Boolean Games. In *Proceedings of TARK'01*, pages 287–298, 2001.

[21] Paul Krause, Simon Ambler, Morten Elvang, and John Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, 11 (1) :113–131, 1995.

[22] Henry Prakken and Gerard Vreeswijk. Logics for defeasible argumentation. In *Handbook of Philosophical Logic*, volume 4, pages 218–319, 2002.

# Apports Complémentaires de la Subjectivité et des Biais Cognitifs à la Rationalité dans le Contexte de la Fonction d'Assistance

François Bouchet⋆ Jean-Paul Sansonnet⋆
bouchet@limsi.fr jps@limsi.fr

⋆ LIMSI-CNRS
Université Paris-Sud XI
BP 133, F-91403 Orsay Cedex

**Résumé :**
Les agents conversationnels sont un moyen prometteur d'assister des utilisateurs novices. Après une analyse sémantique, les requêtes en langue naturelle sont transformées en une représentation formelle utilisée en conjonction avec le modèle de l'application pour définir la réaction la plus appropriée. Cependant, les heuristiques associant des comportements à des schémas de requêtes sémantiquement similaires ne parviennent souvent pas à fournir une réaction efficace et réaliste quand elles se basent uniquement sur des décisions purement rationnelles. Nous proposons donc dans cet article une architecture d'agents conversationnels assistants fondée sur deux éléments : des heuristiques prenant en compte des paramètres à la fois rationnels et subjectifs (basés sur un modèle de personnalité de l'agent), et des biais utilisés pour modéliser des contraintes profondément liées à sa personnalité que l'agent ne peut modifier. Nous illustrons son fonctionnement sur des requêtes typiques issues d'un corpus de requêtes collectées avec un agent assistant.

**Mots-clés :** Agent conversationel, assistance, heuristiques, personnalité, biais cognitif

**Abstract:**
Conversational agents are a promising way to provide assistance to novice users. After a semantic analysis, natural language requests are transformed into a formal representation the agent is using in conjunction with a model of the application to define the most appropriated reaction. However, in many cases, heuristics associating behaviors to patterns of semantically similar requests fail to provide a reaction both efficient and realistic when they are only based on purely rational decisions. To face this issue, we propose in this article an architecture for assisting conversational agents based on two elements : heuristics taking into account both rational and subjective parameters (based on a psychological model of the agent), and biases used to model deep personality contraints that the agent is unable to modify. We illustrate its functioning over some typical requests extracted from a collected corpus of requests to an assisting agent.

**Keywords:** Conversational agent, assistance, heuristics, personality, cognitive bias

## 1 Introduction

### 1.1 Contexte

Les chercheurs ont montré que quand des utilisateurs novices placés face à un logiciel inconnu ont besoin d'assistance, ils ont tendance à demander de l'aide à "un ami derrière l'épaule" plutôt que d'avoir recours au système d'aide traditionnel disponible sur leur ordinateur [3]. Si ce phénomène s'explique par la saillance cognitive de la tâche en cours, appelé "paradoxe de la motivation" [4], il n'est pas interdit de penser qu'il est aussi lié au besoin d'une interaction plus intuitive et plus compréhensive. Cette dernière hypothèse est confirmée par l'observation du 'Persona Effect' par Lester [12], ainsi que de l'effet positif de l'interaction en Langue Naturelle dans le contexte de la Fonction d'Assistance. Tout ceci suggère que l'emploi d'Agents Conversationnels Animés (ECA – pour Embodied Conversational Agents [6]) pour aider les utilisateurs, en particulier les novices, pourrait se révéler efficace. Dans cet article, nous nous intéressons tout particulièrement à la sous-classe d'ECAs dédiés à la Fonction d'Assistance, que nous appellerons donc "Agents Conversationnels Assistants" (ACA).

Dans le cas où l'on considère les agents assistants munis d'une personnalisation (*i.e.* d'une apparence physique sous forme d'un personnage animé à l'écran), plusieurs études ont été menées afin d'améliorer la crédibilité physique des agents, en particulier via l'expression des émotions de base [1][13]. Dans cette veine, nous pensons que pour franchir la "vallée dérangeante" [14] qui fait obstacle à l'acceptabilité de ces outils par le grand public, nous aurons besoin dans le futur d'agents qui ne seront pas seulement *physiquement crédibles* mais aussi *cognitivement crédibles*, c'est-à-dire capables de produire des comportements complexes, proches de ceux des êtres humains. Une façon d'aller dans cette direction est de munir les agents : d'une part, de paramètres de personnalité semblables à ceux utilisés dans les études en psychologie humaine, et d'autre part, de contraintes cognitives intégrées au cœur même de l'architecture supportant les agents afin d'émuler les restrictions que des humains auraient en des circonstances similaires.

### 1.2 Travaux connexes

Dans le domaine de la simulation des communautés humaines à base de Systèmes Multi-Agents, la notion “d’agents cognitifs” (c’est-à-dire utilisant des théories cognitives afin de modéliser les capacités de raisonnement rationnel des agents) a déjà été explorée, par exemple via l’ajout d’une couche de traitement au-dessus des plateformes de déploiement d’agent traditionnelles comme, Co-JACK [15][8] pour JACK qui prend en compte les paramètres de simulation de certaines contraintes psychologiques humaines comme : les délais cognitifs, la limite de la mémoire de travail (par exemple, “oublier une croyance” si l’activation est faible ou encore “oublier quoi faire ensuite” dans une procédure donnée), l’utilisation du flou pour la remémoration des croyances, un domaine attentionnel limité, ou encore l’utilisation de ‘modérateurs’ altérant la cognition. Il y a eu aussi des tentatives d’ajout d’émotions aux architectures classiques de type BDI [18], par exemple pour prendre en compte la peur, l’anxiété ou la confiance en soi par l’ajout de paramètres comme les désirs fondamentaux, les capacités et les ressources [17]. L’idée d’ajouter des degrés en logique multivaluée pour les croyances, les désirs et les intentions a été étudiée dans [5] au moyen de la logique de Łukasiewicz. Ceci a conduit à montrer que l’ordre dans lequel les heuristiques sont appliquées a un impact essentiel sur la personnalité de l’agent, telle qu’elle est perçue par les humains : si nous considérons des classes de règles (comme les BDI ou encore les Obligations), ce peut aussi être une manière de caractériser la personnalité de l’agent avec des traits comme Stable, Égoïste ou Social [7].

### 1.3 Le besoin de personnalité pour un ACA

Dans cet article, nous mettons l’accent sur cette question essentielle dans le contexte des ACAs en considérant des situations où le raisonnement rationnel de l’agent seul ne suffit pas à produire une réaction pertinente en termes de :

– *Compétence* : la capacité de produire non seulement une réponse utile pour une requête utilisateur donnée mais aussi une réponse qui satisfasse les intentions de l’utilisateur ? celles-ci n’étant pas forcément exprimées de manière explicite dans sa requête mais pouvant être liées à la pragmatique linguistique (ex : “Dommage qu’on ne puisse revenir à la page d’accueil” → “Revenir à l’accueil”).
– *Réalisme* : la capacité à réagir de manière similaire à un assistant humain placé dans la même situation et qui plus est en maintenant un comportement consistant (par exemple ne pas basculer brutalement, sans raison, d’un comportement coopératif à un comportement antagoniste). Dans l’étude des émotions, la consistance a été identifiée comme un facteur crucial par Ortony [16].

Considérons par exemple le cas d’un agent recevant une requête simple, où l’utilisateur demande comment faire pour quitter l’application “Comment je fais pour quitter ?”. D’un point de vue purement rationnel, l’agent devrait répondre par exemple en désignant d’un geste déictique le bouton ‘QUIT’ ou encore en détaillant textuellement une procédure à suivre. En termes de compétence, ces réponses sont valables car elles satisfont parfaitement l’objectif explicitement exprimé par l’utilisateur. En termes de réalisme, cette interprétation n’est pas suffisante car elle ne prend pas en compte l’implication que porte cette simple demande qui n’est pas neutre psychologiquement. En fait, si on considère un assistant humain, selon le contexte d’interaction et selon sa personnalité, il pourrait avoir les réactions suivantes :

– *Surprise* : par exemple la tâche en cours n’est pas achevée (sensibilité au contexte) ;
– *Déception* : si par exemple il apprécie l’utilisateur (sensibilité à la subjectivité) ;
– *Satisfaction* : si au contraire l’utilisateur a été impoli (sensibilité à la subjectivité).

Ainsi, même si il y a des situations dans lesquelles un comportement purement rationnel est acceptable (dans une situation de contrôle/commande stricte ou encore dans le cas d’un agent ayant des traits de personnalité comme Servile ou Introverti), la décision doit tout de même être motivée sur la base des interactions précédentes et de la personnalité de l’agent. Si au contraire on se fonde sur des réactions purement rationnelles, le comportement de l’agent risque de n’être pas compris/admis par l’utilisateur, en raison de la tendance naturelle des humains à l’anthropomorphisme consistant à attribuer des traits psychologiques à des objets technologiques [19].

### 1.4 Le besoin de contraintes cognitives pour un ACA

Un autre aspect du réalisme concerne la façon dont l’agent prend les décisions mentionnées ci-dessus. En effet, dans un système à base d’heuristiques :

1) même les décisions conduisant à exprimer sa personnalité sont toujours calculées intentionnellement ; on entend par là que l'agent doit être capable de produire, si on le lui demande, les règles qui ont été appliquées et donc déterminer son comportement rationnel de base sans elles ;
2) si nous considérons un agent dans lequel les heuristiques sont construites par des concepteurs d'agents différents et indépendants, chaque règle peut être potentiellement cachée ou altérée par d'autres règles de priorité supérieure (*i.e.* appliquées après). Ceci peut poser problème dans le cas où des comportements fortement émotionnels sont produits. Par exemple, si nous considérons que l'agent est rendu 'en colère', celle-ci pourrait ne pas etre exprimée à cause d'une autre règle stipulant que l'agent ne devrait pas exprimer de comportements extrêmes (socialement non convenables) – un 'self-control' que n'a pas toujours un être humain soumis à de fortes émotions ;
3) si nous considérons le cas d'un agent qui apprend un comportement optimal en fonction de ses interactions avec des utilisateurs humains (ce que Sloman nomme 'self-monitoring' dans [21]), il pourrait finir par se débarrasser de certaines heuristiques qui augmentent son réalisme mais qui diminuent sa compétence. En effet, alors que l'efficacité peut se mesurer en comparant les résultats prédits par les heuristiques à l'état du monde réel obtenu en les appliquant ou d'après les retours langagiers des utilisateurs, il semble difficile d'imaginer une manière pour un agent d'évaluer son réalisme.

Pour toutes ces raisons, il faut implémenter les contraintes cognitives de manière suffisamment profonde pour rendre l'agent incapable a) d'expliquer son comportement et b) de les modifier. Certes cette approche est contre-intuitive vis-à-vis des systèmes à base de règles traditionnels, mais c'est le point de vue que nous allons adopter dans cet article, en introduisant la notion de "biais cognitifs" (dans la suite, simplement *biais*) qui sont des règles de transformation classiques mais avec deux différences essentielles les rendant assimilables à des filtres :
a) elles agissent comme des règles cachées : comme elles sont appliquées en-dehors du moteur de règles principal de l'agent, leur impact n'est pas analysable par l'agent ;
b) elles agissent sur les interactions même de l'agent d'une part avec le Monde (extérieur) et d'autre part avec sa propre Mémoire (intérieur). L'implémentation garantit que ces règles/filtres seront toujours les premiers/derniers à être appliqués ce qui les rend non-altérables par les heuristiques définissables par les concepteurs d'agents assistants. Enfin elles "fonctionnent à perte" ce qui revient à dire que le fait de les appliquer efface la requête utilisateur originelle qui n'est plus désormais accessible à l'agent. Par exemple dans l'agent mécontent mentionné plus haut, il ne pourrait plus contrôler l'expression de sa colère dans ses réponses ; en fait, il n'en serait même pas conscient (*i.e.* pas capable d'en posséder une représentation symbolique dans sa base de connaissances).

Afin d'exhiber ce que pourraient être les paramètres de personnalité pertinents pour un agent assistant, il est indispensable de procéder à des expérimentations avec diverses classes d'agents. Ceci débouche sur la nécessité de développer une architecture capable de supporter cette variabilité. Dans cet article, nous commençons par la proposition d'une architecture d'agent rationnel dans laquelle les heuristiques sont capables d'entremêler subjectivité et rationalité afin d'exprimer des comportements réalistes, en particulier par le recours à un modèle psychologique de la personnalité de l'agent. Dans une seconde étape, nous montrons comment il est possible de modifier cette architecture pour modéliser la notion de biais cognitif au travers d'exemples de contraintes cognitives que les humains peuvent faire apparaître dans leurs réactions. Enfin, nous concluons par une discussion au sujet de notre approche dans le contexte des agents conversationnels dédiés à la Fonction d'Assistance.

# 2 Architecture d'agent Rationnel et Subjectif

## 2.1 Notations du modèle

Nous définissons une *entité* comme étant toute notion que nous voulons réifier (*i.e.* représenter symboliquement). Formellement une entité est représentée par un ensemble de triplets (à la manière de RDF [10]) associée à un identifiant de la forme :

$$\#id = H\left[\bigcup_i a_i \rightarrow v_i\right]$$

Où $\#id$ dénote l'*identifiant* unique donné arbitrairement à la référence, $H \in \mathbb{H}$ est la *tête* de l'entité, $a_i$ un attribut pris dans la liste des attributs disponibles pour $H$ et $v_i$ une valeur prise dans le domaine des expressions définies par le type associé à $a_i$. En fonction du type, $v_i$ peut être :

– une valeur terminale (chaîne de caractères, nombre, valeur symbolique énumérée...),
– une nouvelle entité, selon ce même format d'entité ;
– un identifiant correspondant à une entité déjà définie.

Nous définissons aussi un *moteur* ou *engine* comme étant une façon de générer une nouvelle entité ou bien encore de transformer une entité en une autre, grâce à un ensemble de règles définissant le moteur.

Nous appelons *domaine* un ensemble d'éléments (entités ou moteurs) qui ont un accès direct les uns aux autres. Ainsi, nous distinguons quatre domaines différents :

1. *La Mémoire de l'agent* ($\mathcal{M}$), qui stocke toutes les connaissances que l'agent possède à l'origine ou acquises lors de ses interactions. Plus particulièrement, la mémoire se décompose en trois parties :
   – *La mémoire sémantique* ($\mathcal{M}_s$), qui contient le modèle partiel du Monde que l'agent a pu acquérir de manière directe par consultation ou indirecte par inférence.
   – *La mémoire épisodique* ($\mathcal{M}_e$), dont la différence avec $\mathcal{M}_s$ vient du fait qu'elle se focalise sur l'agent lui-même, représentant ainsi sa mémoire autobiographique (*i.e.* "ce qui lui est arrivé", *cf.* [22]). Dans le cas d'agents conversationnels, l'agent ne connait le Monde qu'au travers de ses interactions avec l'utilisateur d'une part et l'application assistée d'autre part : sa mémoire épisodique est alors réduite à l'enregistrement des entrées/sorties interactionnelles [1].
   – *La mémoire procédurale* ($\mathcal{M}_p$), qui est composée d'un ensemble d'heuristiques, c'est-à-dire de règles à appliquer selon certaines conditions. Pour cette étude, nous supposerons qu'elles sont déclenchées uniquement lors de la réception d'une requête de l'utilisateur.
2. *Les états mentaux de l'agent* ($\Psi$) contiennent l'information concernant la psychologie de l'agent, modélisée selon quatre types de notions : traits (traits), humeurs (moods), rôles (roles) et affects (relationships).
3. *Le moteur de l'agent*, ou encore *engine* ($\mathcal{E}$), chargé d'exécuter la partie active de l'agent. Il se décompose en :
   – Un moteur de Traitement Automatique de la Langue Naturelle ($\mathcal{E}_L$), qui transforme les questions en langue naturelle des utilisateurs en une représentation formelle de leur sémantique.
   – Un moteur comportemental ou encore *behavioral* ($\mathcal{E}_B$), qui détermine quelles heuristiques (dans $\mathcal{M}_p$) doivent être activées en fonction de la requête [2] et qui calcule les réactions en fonction de ces heuristiques en les paramétrant par les valeurs actuelles des variables y apparaissant.
4. Le Monde ($\mathcal{W}$) est défini, du point de vue de l'agent comme étant tout ce qui n'est pas interne (*i.e.* ni dans $\mathcal{E}$, $\Psi$ ou $\mathcal{M}$). En particulier, $\mathcal{W}$contient les informations sur les utilisateurs et l'application assistée. Nous supposerons que le Monde existe et évolue de manière indépendante de l'agent, même si l'agent peut influer sur le Monde.

### 2.2 Représentation du Monde ($\mathcal{W}$)

Le Monde est constitué d'entités représentées dans la syntaxe définie au § 2.1. Par exemple, l'information disponible au sujet d'un utilisateur donné peut être représentée comme :

```
#user7 = PERSON[
            name   -> "Smith",
            role   -> user,
            age    -> 20,
            gender -> male
]
```

### 2.3 États mentaux de l'agent ($\Psi$)

Nous distinguons quatre types d'états mentaux en fonction de leur dynamique et de leur arité fonctionnelle, tels que résumés dans la table 1. À chacun d'entre eux est associée une valeur numérique dans l'intervalle réel $[-1, 1]$, 0 définissant par défaut "l'état mental neutre". Les états mentaux sont représentés sous forme attribut-valeur de l'agent comme par exemple :

```
#ums = unary-mentalstate[
   mentalstate1 -> 0.8,
   mentalstate2 -> -0.3,
   ...
]
#bms = binary-mentalstate[
```

1. Bien que les interactions entre l'utilisateur et l'agent soient en théorie de nature multimodale, nous ne considérerons ici que la modalité linguistique comme support des interactions utilisateur/agent.

2. Nous considérerons dans cet article que l'agent est de type purement réactif (ou modal) : il ne prend pas l'initiative de démarrer une interaction avec l'utilisateur.

TABLE 1 – Les quatre types d'états mentaux d'un agent, selon leur dynamique et leur arité

| | **Unaire** | **Binaire** |
|---|---|---|
| **Statique** | Trait $\Psi_T$ | Role $\Psi_R$ |
| **Dynamique** | Humeur $\Psi_t$ | Affect $\Psi_r$ |

```
   towards -> #iduser,
   mentalstate1 -> 1,
   mentalstate2 -> 0,
   ...
]
```

**Traits $\Psi_T$.** Les traits correspondent aux attributs classiques de la personnalité qui peuvent être considérés comme stable au cours de la 'vie' d'un agent – ils correspondent aux "Big Five" ou "modèle OCEAN", couramment utilisé en psychologie [9] :
- *Ouverture à l'expérience* : représente l'imagination, la curiosité, le goût pour l'aventure ;
- *Caractère Consciencieux* : représente la tendance à être discipliné et à respecter ses obligations ;
- *Extraversion* : représente l'énergie, la force des émotions positives et le goût des autres ;
- *Agréabilité* : représente la tendance à être compatissant, affectueux et coopératif ;
- *Neuroticisme* : représente la tendance à ressentir facilement des émotions négatives (colère, anxiété, vulnérabilité *etc.*).

**Humeurs (Moods) $\Psi_t$.** Les humeurs (ou moods) représentent les facteurs de personnalité qui varient avec le temps, en fonction des heuristiques et des biais cognitifs. L'état humoral d'un agent est divisé en deux parties :

1. Propriétés physiques : elles modélisent l'agent en tant qu'entité physique du monde (à la manière de la métaphore des attributs physiques des jeux vidéo). Nous ne les traiterons pas dans cet article ;
2. Propriétés épistémiques : elles modélisent l'agent en tant qu'entité capable de raisonnement rationnel. C'est pourquoi elles seront prises en compte dans cette étude.

En résumé, nous aurons les humeurs suivantes :
- *Energy (énergie physique)* : représente la force physique de l'agent, au sens large ;
- *Happiness (bonheur physiologique)* : représente le bien-être physique de l'agent, selon sa situation physique ;
- *Confidence (confiance en soi)* : représente la force cognitive de l'agent ;
- *Satisfaction (satisfaction intellectuelle)* : représente le bien-être mental de l'agent, selon l'analyse qu'il fait de sa situation intentionnelle (*i.e.* vis-à-vis de ses B-D-I).

**Rôles $\Psi_R$.** Les rôles représentent des relations statiques à caractère institutionnel, entre l'agent et d'autres entités du monde (*e.g.* la relation utilisateur/assistant). On peut définir deux grandes catégories :
- *Authority (autorité)* : elle caractérise le droit que l'agent croit posséder d'être directif vis-à-vis de son interlocuteur ou réciproquement de ne pas accepter (facilement) des directives de celui-ci. Cette relation est souvent antisymétrique : `Authority(X,Y) = -Authority(Y,X)`
- *Familiarity (familiarité)* : elle caractérise le droit que l'agent croit posséder de se comporter de manière informelle vis-à-vis de son interlocuteur. Cette relation est souvent symétrique : `Familiarity(X,Y) = Familiarity(Y,X)`

Dans le cas de la Fonction d'Assistance, l'autorité sera a priori clairement en faveur de l'utilisateur et nous aurons par conséquent :

```
role[
   towards -> #iduser,
   authority -> val1,
   familiarity -> val2
]
```

dans lequel 'val1' doit avoir une valeur négative.

**Affects (relationships) $\Psi_r$.** Les affects modélisent les relations dynamiques entre l'agent et les autres entités (typiquement les utilisateurs). Nous en distinguons au moins trois sortes :
- *Dominance* : exprime que l'agent se sent puissant par rapport à l'interlocuteur. Cette relation est souvent antisymétrique, comme par exemple `Dominance(X,Y) = -Dominance(Y,X)` ;
- *Affection* : exprime une attirance et une tendance à agir amicalement de l'agent envers l'interlocuteur. Cette relation n'est pas nécessairement symétrique ;
- *Trust (confiance)* : exprime que l'agent fait confiance à l'interlocuteur. Cette relation n'est pas nécessairement symétrique.

## 2.4 La mémoire de l'agent ($\mathcal{M}$)

**Mémoire épisodique $\mathcal{M}_e$.** Elle contient les interactions précédentes de l'agent avec d'une part l'utilisateur et d'autre part l'application. Nous distinguons les messages entrants (`INBOX`) des messages sortants (`OUTBOX`). L'information est représentée par des triplets de la forme :

```
INBOX[
   from -> [sender],
   time -> [timestamp],
   message -> [message]
]
OUTBOX[
   to -> [recipient],
   time -> [timestamp],
   message -> [message]
]
```

**Mémoire sémantique $\mathcal{M}_s$.** Elle contient un sous-ensemble étendu du Monde et de ce fait utilise la même représentation. Il s'agit d'un sous-ensemble car la totalité de $\mathcal{W}$ est souvent inatteignable pour l'agent ; il s'agit d'une extension car l'agent crée à tout moment de nouveaux faits lors de l'exécution des heuristiques et des filtres de biais. La manière dont l'agent construit cette mémoire de manière automatique grâce à des heuristiques d'observation du Monde (ou *observateurs*) sort du cadre de cet article, mais des éléments sont donnés dans [11].

**Mémoire procédurale $\mathcal{M}_p$.** Elle contient un ensemble d'heuristiques définissant comment l'agent doit réagir à une requête entrante. Une heuristique contient deux parties :

- une *tête*, définissant les classes de requêtes avec lesquelles les heuristiques doivent s'apparier selon la syntaxe des expressions régulières. Par exemple, les requêtes concernant la possibilité d'exécuter une action.
- un *corps*, définissant un arbre de décision permettant de construire les réactions de l'agent à cette requête, chaque nœud de l'arbre étant basé sur une combinaison de valeurs retournées par les requêtes envoyées vers $\mathcal{M}$ et $\mathcal{W}$ (pour la partie de la réponse liée à la rationalité) ou bien vers $\mathcal{M}_s$ (pour la partie de la réponse liée à la subjectivité). Cela se termine toujours par un envoi de requête de performatif de type `INFORM` vers le Monde.

**Exemple :** Réaction d'un agent quand l'utilisateur lui interdit d'effectuer une action donnée, par exemple : "Je te défends d'ouvrir le fichier de configuration !". Il s'agit d'une réaction très subjective définie par le code suivant :

```
HEURISTIC[
   description -> "reaction to an interdiction",
   head -> NEG[AUTHORIZATION[
                granter -> person[id="user"],
                granted -> person[id="system"],
                todo -> A___]],
   body -> {
# If the agent is conscientious, it first checks
# if the action was doable in first place
if (conscientiousness > 0):
 # Choice of the repository to check: world or memory
 repository = (confidence > 0.5 ? W ; MS)
 # Check if the action is doable at first
 allowed = CHECK[repository, DOABLE[A], true]

# If the information is not found in the memory but
# was in the world, the agent loses some confidence
 if (allowed == unknown && repository = MS):
  allowed = CHECK[W, DOABLE[A], true]
  if (allowed != unknown):
   INFORM[memory, decrease(confidence)]

# If the action was not doable,
# the agent gets more confidence in itself
 if (allowed == false):
  INFORM[memory, increase(confidence)]
  # If it's cooperative, it says it, + or - nicely
  if (agreeableness > 0):
   if (affection(user)>=0 && familiarity(user)>=0):
    answer += POSITIVE[NOTPOSSIBLE[A___]];
   elif (affection(user) < -0.5):
    answer += NEGATIVE[NOTPOSSIBLE[A___]];
   else:
    answer += NOTPOSSIBLE[A___];

# If the user has authority the fact is stored
if (authority(user) > 0):
 INFORM[memory, forbidden(A___)]
 done = true
else:
 done = false

# If the agent is neurotic,
# being forbidden something makes it unsatisfied
if (neuroticism > 0):
 INFORM[memory, decrease(satisfaction)]
 # If it has a high dominance, it will mention it
  if (dominance(user) > 0):
   answer += UNHAPPY

# The action done is acknowledge
# possibily with a negative modelization
# which expression depends on the familiarity,
if (satisfaction < -0.3 && familiarity(user) > 0):
 answer += NEGATIVE[(done?ACK:NACK)]
elif (done && satisfaction < -0.8):
 answer += NEGATIVE[(done?ACK:NACK)]
else:
 answer += (done?ACK:NACK)

# it finally transmits the built answer to the user
INFORM[user, answer]
 }
]
```

**Remarques :**

- La réaction est construite progressivement dans la variable 'answer' ;
- La génération effective de la Langue Naturelle n'est pas traitée ici : on se contente d'employer des patrons comme NOTPOSSIBLE, NEGATIVE ou ACK qui devraient être post-traités. Ces symboles pourraient aussi être interprétés de manière multimodale, par exemple pour choisir les animations physiques de la personnification de l'agent.

Le contenu du corps de cette heuristique sera détaillé en 2.5.

## 2.5 Fonctionnement dynamique

**Le moteur de traitement de la Langue Naturelle $\mathcal{E}_L$.** Il a pour rôle de prétraiter les requêtes en Langue Naturelle pour les mettre sous forme de représentation formelle transmise à $\mathcal{E}_B$, selon le langage défini dans [2]. Typiquement, les requêtes en Langue Naturelle sont traitées en deux étapes :

- Une phase d'analyse grammaticale : les outils de TALN classiques sont appliqués (lemmatisation, étiquetage lexical, désambigüisation sémantique...) ;
- Une phase d'analyse sémantique : une requête formelle est construite selon le langage défini dans [2].

**Moteur comportemental $\mathcal{E}_B$.** Le moteur comportemental de l'agent accède régulièrement à l'information stockée dans d'autres domaines ($\mathcal{M}$, $\Psi$ ou $\mathcal{W}$). Par définition, $\mathcal{E}$ ne peut pas accéder directement aux autres domaines, il doit utiliser à cet effet des *requêtes*. Dans cet environnement simplifié, nous faisons une distinction entre trois sortes de requêtes :

- `INFORM[domain, request]` : transmet le contenu de la requête à un domaine. Cette requête n'attend pas de requête en retour.
- `GET[domain, value]` : requiert la valeur d'un élément auprès d'un domaine. Cette requête attend une requête `INFORM[X,Y]` en réponse de la part du domaine interrogé.
- `CHECK[domain, attribute, value]` : demande si la valeur d'un attribut du domaine interrogé est bien celle fournie comme troisième paramètre de la requête. Cette requête attend une requête `INFORM[X,Y]` en réponse de la part du domaine interrogé où la valeur peut être Faux, Vrai ou Inconnu.

Remarque : En raison du modèle de requêtes simplifié choisi, nous ne détaillons pas la sémantique du protocole à la manière de l'ACL-FIPA [20] qui utilise des pré-conditions et post-conditions en logique modale.

**Protocole d'Interaction général.** Ici, la seule façon dont l'utilisateur peut interagir avec l'agent est via l'interaction textuelle. En conséquence, une interaction commence toujours par une entrée en Langue Naturelle émanant de l'utilisateur qui déclenche la séquence d'événements suivante (*cf.* figure 1) :

1. La requête est traitée par $\mathcal{E}_L$ qui fournit une représentation formelle à $\mathcal{E}_B$ ;
2. La représentation formelle de la requête entrante est stockée dans $\mathcal{M}_e$ ;
3. En fonction de la représentation formelle de la requête et des patrons définis dans la tête de chaque heuristique stockée dans $\mathcal{M}_p$, $\mathcal{E}_B$ récupère le corps associé à l'heuristique déclenchée et exécute les règles qu'il contient. Ceci conduit à la génération d'une ou de plusieurs requêtes vers d'autres domaines :
   - si les règles requièrent l'utilisation d'éléments subjectifs, ils sont récupérés dynamiquement à la demande depuis $\Psi$ ;
   - si les règles requièrent l'utilisation d'informations au sujet de l'application, ils sont récupérés dynamiquement à la demande depuis $\mathcal{M}_s$ ou $\mathcal{W}$ ;
4. La réaction formelle venant d'être construite est envoyée vers $\mathcal{E}_L$ afin de générer une réponse en Langue Naturelle qui doit être transmise à l'utilisateur.

Durant le traitement des effets des règles, plusieurs comportements distincts sont envisageables qui dépendent essentiellement des paramètres de personnalité de l'agent. Tout d'abord pour les requêtes de type CHECK, il y a deux possibilités :

1. Priorité au test dans $\mathcal{M}_s$ : si l'agent a de la confiance en soi (confident) ou bien n'a plus accès à cette information à partir du Monde (par exemple, s'il s'agit d'une question relative à un état précédent de l'application) il essaye de retrouver directement l'information désirée depuis $\mathcal{M}_s$. Il ne teste $\mathcal{W}$ que s'il n'arrive pas à trouver une réponse dans sa propre mémoire.
2. Priorité au test dans $\mathcal{W}$ : si l'agent n'a pas confiance en lui (not confident), il ne se donnera pas la peine de vérifier dans sa propre mémoire et essayera de retrouver l'information depuis $\mathcal{W}$.

Alors se pose le problème de savoir ce qui doit être stocké ou non dans $\mathcal{M}_s$ (stockage non représenté dans la figure 1) quand une information est récupérée depuis le Monde. À nouveau, deux options sont possibles :

1. Copie stricte de $\mathcal{W}$ : pour un agent sérieux (Conscientious), la mémoire de l'agent se comporte exactement comme un cache et un historique du Monde ;
2. Copie surchargée de $\mathcal{W}$ : pour un agent prêt à prendre plus de risques, au plan épistémique, la mémoire de l'agent contient aussi le résultat des calculs qu'il est amené à effectuer en interne. Par exemple, si on lui demande combien l'application contient d'objets d'un certains type (*e.g.* des boutons), il

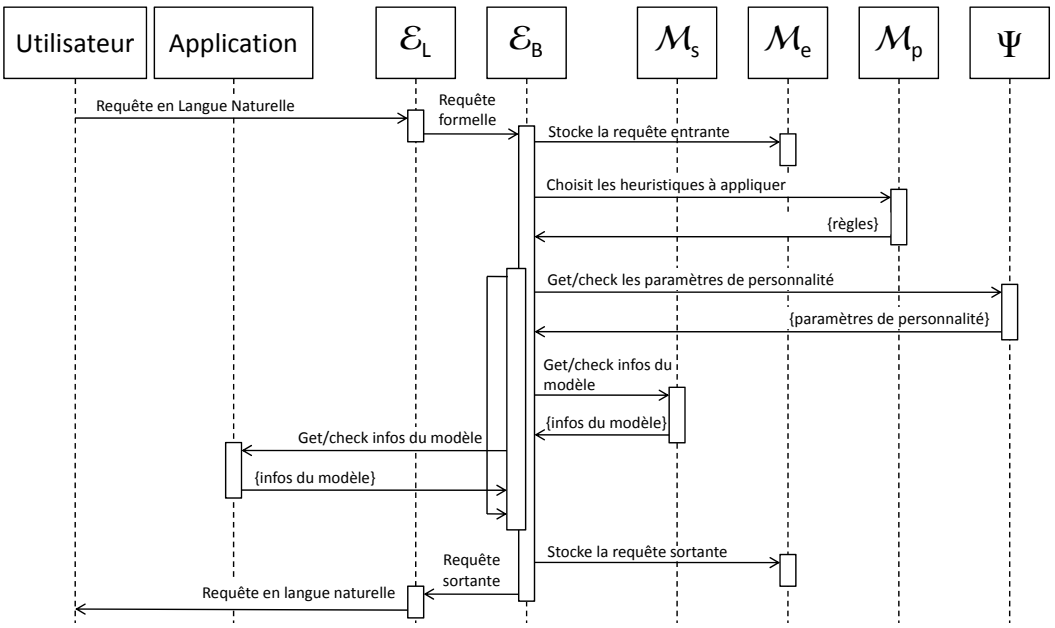



FIGURE 1 – Diagramme de séquence du traitement typique d'une requête usager par l'agent.

ne se contentera pas de stocker la liste des boutons mais il synthétisera aussi un fait décrivant explicitement cette information.

À partir d'ici, on constate qu'un agent confiant en lui mais pas très sérieux peut manquer d'efficacité car il est amené à répondre directement à l'utilisateur, sur la base d'informations qui sont peut-être obsolètes. Cependant, en termes de réalisme, ce comportement émule assez bien celui d'un être humain possédant ce trait de personnalité (Conscientious = -1) et cette humeur (Confident = 1) ; et s'il peut sembler inacceptable à un utilisateur lui-même sérieux, il n'est pas sûr qu'un utilisateur ayant le même profil blâmerait l'agent pour cela.

**Exemple d'interaction.** Considérons une requête de contrôle ou l'utilisateur interdit à l'agent d'ouvrir un fichier donné. Nous aurons :
*Phrase de l'utilisateur :* "N'ouvre pas le fichier !" ⇔ " Don't open the file !"

*Requête formelle produite par l'analyseur sémantique :*

```
NEG[AUTHORIZATION[
  granter -> person[id="user"],
  granted -> person[id="system"],
  todo -> Open[
    element -> object[
      properties -> {
        type -> type[val="file"]
        quantity -> quantity[val=1]
      }
]]]]
```

Si nous supposons pour simplifier qu'il n'y a qu'une seule heuristique dont la tête s'apparie avec cette requête formelle, à savoir celle décrite en 2.4, l'agent générera une réponse composée de trois sous-phrases : $[not_possible][unhappy][ack/nack]$

- $[not_possible]$ est généré seulement si l'agent est suffisamment sérieux (Conscientious> 0.6) pour avoir pris la peine de vérifier au préalable que le fichier est techniquement ouvrable et suffisamment coopératif (Agreeableness > 0.6) pour informer l'utilisateur à ce sujet. Cette phrase peut aussi être 'modalisée' positivement ou négativement selon l'affection et la familiarité envers l'utilisateur.
- $[unhappy]$ est généré seulement si l'agent est neurotique. On peut supposer que dans un tel cas il existe une règle stipulant qu'il n'aime pas les interdictions – et qu'il se sente suffisamment dominant envers l'utilisateur pour exprimer explicitement sa rancœur dans sa réponse.
- $[ack/nack]$ est toujours généré pour faire savoir à l'utilisateur si sa commande a été prise en compte ou pas par l'agent.

Cette heuristique illustre le fait qu'un agent peut faire plus de choses qu'il ne laisse transparaître dans sa réponse à l'utilisateur. Par exemple, s'il n'est pas coopératif (Agreeableness < 0) mais sérieux (Conscientious > 0.5) il vérifiera que l'opération demandée est techniquement effectuable, indépendamment de la décision d'en informer ou non l'utilisateur dans le cas d'une

TABLE 2 – Résumé des différences entres les règles appliquées par les biais et les heuristiques

| | **Heuristiques** | **Biais** |
|---|---|---|
| **Objectif** | Génération d'une réaction | Modification d'une requête |
| **Portée** | Une classe de requêtes d'utilisateur | Toute requête entre deux domaines |
| **Ressources** | Requête de l'utilisateur, $\mathcal{W}$ , $\mathcal{M}$ et $\Psi$ | $\Psi$ seulement |
| **Introspectables** | Oui | Non |

impossibilité technique. Le fait d'avoir effectué cette vérification, même si elle ne transparaît pas dans l'interaction, peut avoir un impact sur les états mentaux de l'agent.
Nous voyons aussi que certains paramètres de personnalité similaires (statiques ou dynamiques) comme l'autorité et la dominance peuvent être utilisés conjointement : si l'agent n'a pas d'autorité envers l'utilisateur mais se sent dominant envers lui, il peut fournir une réponse rebelle du type "Ca m'ennuie mais ok".

# 3 Introduction des biais cognitifs

## 3.1 Définition

Nous avons vu que $\mathcal{E}_B$ communique avec $\mathcal{M}$ et $\mathcal{W}$ au moyen de requêtes. Ces requêtes peuvent être modifiées lorsqu'elles transitent d'un domaine à l'autre par des filtres appelés *biais cognitifs* ou encore *biais*. Un biais agit donc comme une transformation sur les requêtes de l'agent sans que celui-ci ne puisse en avoir connaissance. Un biais $b$ sur une requête entre deux domaines X et Y sera représenté par : $\mathrm{X}\xrightarrow{b}\mathrm{Y}$.
La différence fondamentale avec les heuristiques stockées dans $\mathcal{M}_p$ est alors l'impossibilité pour l'agent d'expliquer les biais : dans la plupart des cas, il ne peut même pas se rendre compte (par exemple par inférence) qu'ils ont été appliqués. En outre, tandis que les heuristiques produisent des requêtes, c'est-à-dire qu'elles essayent de définir une réaction pour une situation spécifique donnée, au contraire les biais sont des contraintes s'appliquant à toute réaction. Néanmoins, les biais comme les requêtes sont affectés par les valeurs des paramètres de $\Psi$. La table 2 résume ces différences.

## 3.2 Catégories et exemples de biais

Les biais sont orientés, ce qui signifie pour une paire donnée de domaines X et Y il est possible de définir deux sortes de biais : $\mathrm{X}\xrightarrow{b}\mathrm{Y} \neq \mathrm{Y}\xrightarrow{b}\mathrm{X}$. De plus les biais sont dépendants du type de requête transmise entre deux domaines, ce qui fait que si chacun des quatre domaines définis plus haut pouvaient communiquer avec chacun des trois autres il y aurait six canaux bidirectionnels portant trois types de requêtes (INFORM, GET, CHECK) ce qui engendrerait une combinatoire de $6 \times 3 \times 2 = 36$ biais différents. Cependant, la plupart d'entre eux ne sont pas pertinents pour plusieurs raisons :

– Chaque fois que des processus sont actifs dans $\mathcal{M}$ ou $\Psi$, nous supposerons qu'ils n'ont pas la nécessité de communiquer avec un autre domaine, et que $\mathcal{E}$ est l'unique domaine de l'agent capable de communiquer avec le Monde : $\mathcal{E}_B$ constitue donc le centre de communication de l'agent.
– Il est difficile d'imaginer des situations où l'agent ne serait pas capable de connaître exactement son propre état mental, c'est pourquoi nous ne considérerons pas l'existence de biais entre $\mathcal{E}_B$ et $\Psi$. En fait, nous supposerons aussi que $\Psi$ est directement accessible à partir des heuristiques et des biais (*cf.* l'exemple d'heuristique donné plus haut où l'on accède à la valeur du paramètre Conscientious de manière directe plutôt que par l'utilisation d'une requête de la forme `GET[mentalstates, conscientiousness]`).

Une fois ces considérations posées, il ne reste plus que sept canaux unidirectionnels parmi lesquels cinq seulement peuvent être munis de biais, comme indiqué dans la figure 2. Les cinq catégories de biais restantes sont alors :

– Biais Perceptif ($\mathcal{W}\xrightarrow{B_p}\mathcal{E}_B$) : biais sur une requête `INFORM` émanant du Monde (en fait il s'agit de l'utilisateur dans le cas d'une requête en Langue Naturelle, ou encore le reste du Monde s'il s'agit de la conséquence d'une requête de type GET envoyée précédemment).
– Biais Expressif($\mathcal{E}_B\xrightarrow{B_e}\mathcal{W}$) : biais sur une requête `INFORM` envoyée vers le Monde.
– Biais de Recherche de faits en mémoire

($\mathcal{M} \xrightarrow{B_{mr}} \mathcal{E}_B$) : biais sur une requête`INFORM` émanant de la mémoire (en réponse à une requête`GET` ou `CHECK` envoyée précédemment).
– Biais de lecture mémoire ($\mathcal{E}_B \xrightarrow{B_{ma}} \mathcal{M}$) : biais sur une requête `GET` ou `CHECK` envoyée à la mémoire.
– Biais d'écriture mémoire ($\mathcal{E}_B \xrightarrow{B_{ms}} \mathcal{M}$) : biais sur une requête `INFORM` envoyée à la mémoire.

Des exemples pour ces cinq catégories de biais sont donnés dans la section 3.4.

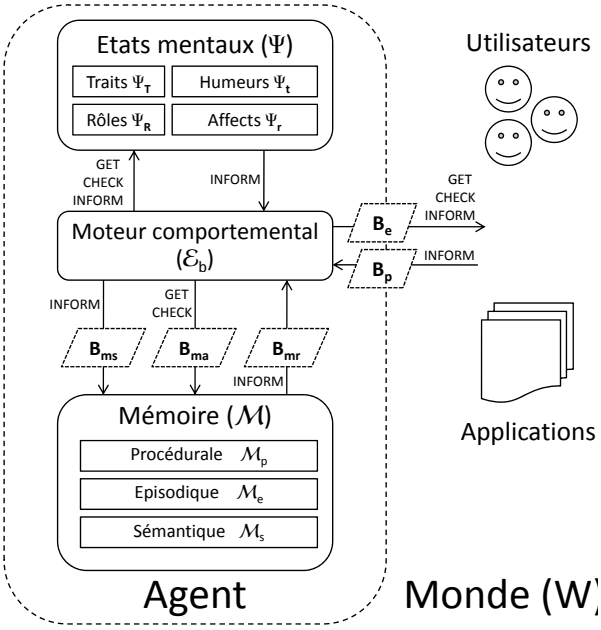



FIGURE 2 – Architecture générale de l'agent avec les canaux portant les cinq catégories de biais

### 3.3 Représentation des biais

Tout comme les heuristiques, les biais possèdent deux parties :
– La *catégorie* du biais : sélectionnée parmi une des cinq catégories définies ci-dessus ;
– Le *corps* : en termes formels, il n'y a pas de différence fondamentale entre la représentation du corps d'un biais et celle du corps d'une heuristique : les deux sont fondés sur un arbre de décision. Cependant, les nœuds d'un biais ne prennent en compte que les aspects de la subjectivité : ils n'ont pas accès en lecture ou en écriture aux éléments de $\mathcal{W}$ ou de $\mathcal{M}$. De plus, les actions possibles sont limitées aux modifications (directes) des paramètres dynamiques de la personnalité (humeurs et affects) ainsi que des requêtes (mais aucune requête supplémentaire ne peut être engendrée par un biais).

Par exemple, nous pouvons considérer un biais perceptif qui pourrait être appliqué par un agent nerveux (Neuroticism > 0) et malheureux (Happiness < 0) le poussant à percevoir négativement toute demande en entrée :

```
BIAS[
   description -> "victimization",
   category -> "perceptive"
   body -> {
   if (neuroticism < -0.5
       && satisfaction < -0.9):
      output = NEGATIVE[input]
   }
]
```

### 3.4 Exemples de biais

**Biais perceptifs.**
– Victimisation : comme détaillé ci-dessus, il s'agit de la tendance qu'a un agent nerveux et malheureux à percevoir plus de négativité qu'il n'y en a réellement dans une phrase de son interlocuteur ;
– Minimisation : inversement, si l'agent a un haut degré de satisfaction (confident > 0.5) et qu'il n'est pas nerveux (Neuroticism < 0), il aura tendance à sous-évaluer la charge négative contenue dans une phrase de l'interlocuteur.

**Biais expressifs.**
– *Stress* : si l'utilisateur est en position de forte autorité vis-à-vis de l'agent, celui-ci peut être amené à exprimer une dose de nervosité dans ses réactions. Ceci ne s'applique que pour modéliser le stress que l'agent ne peut contrôler et vient agir de manière complémentaire au "stress rationnel". En effet, il y a également du stress qui est lié au contenu propositionnel de la réponse : ce stress supplémentaire devrait être produit par une heuristique quand l'agent, par exemple, n'arrive pas à résoudre une requête posée par l'utilisateur.
– *Enjouement/tristesse* : si l'agent est expressif, il aura tendance à révéler son niveau de satisfaction épistémique par l'ajout de connotations positives (resp. négatives) à ses réponses.

**Biais de recherche de faits en mémoire.**
– *Doute* : si le niveau de confiance de l'agent envers ses propres connaissances est bas et que de plus il a un niveau de satisfaction bas, il peut être amené à rejeter ou tout du moins à

minimiser la valeur de certaines informations retrouvées dans sa mémoire.

**Biais de lecture mémoire.**
– *Mauvaise foi* : si l'agent est très insatisfait et que l'utilisateur a une forte autorité sur lui ou encore que l'agent est très mécontent, il peut être conduit à introduire de fausses informations dans les requêtes qu'il envoie vers $\mathcal{M}_s$, par exemple, par l'oubli d'un paramètre (recherche de tous les boutons alors que l'utilisateur a demandé combien il y avait de boutons rouges). L'information récupérée peut alors être partiellement fausse, mais l'agent sera "réellement convaincu d'avoir agir loyalement", et sa réponse pourrait donc presque être assimilée à un "acte manqué".

**Biais d'écriture mémoire.**
– *Oubli* : dans le cas ou l'agent n'est pas névrotique et qu'il est présentement satisfait, il peut choisir de ne pas enregistrer dans sa mémoire certaines informations négatives (comme par exemple une critique émise par l'interlocuteur), qu'il oublie tout simplement.
– *Désordonné* : dans le cas où l'agent n'est pas très sérieux, il peut perdre au hasard des morceaux d'information appartenant au contenu propositionnel de la requête vers $\mathcal{M}_e$sur laquelle agit le biais.

# 4 Conclusion

Nous avons vu que l'utilisation d'une architecture où les décisions dépendent à la fois de paramètres subjectifs et objectifs fait que l'efficacité de l'aide apportée pau un ACA devient alors fortement dépendante de ses traits de personnalité. Dans la mesure où les heuristiques prennent en compte l'état mental (dynamique) mais aussi la personnalité (statique) de l'agent, les ACAs conçus de cette manière offrent une certaine généricité et peuvent donc être adaptés :
– statiquement selon la personnalité de l'utilisateur, en choisissant des agents aux traits de personnalité similaires ($\Psi_T$) qui sont en général préférés par les interlocuteurs [19] ;
– dynamiquement selon les feedbacks précédents de l'utilisateur. Évidemment, comme les requêtes précédentes de l'utilisateur ont modifié les états mentaux ($\Psi_t$ et $\Psi_r$) de l'agent, cela aura un impact sur ses réactions futures.

L'implémentation de biais cognitifs, indépendamment des autres règles présentes dans le corps des heuristiques, permet d'émuler certaines contraintes des comportements humains et de donner par là même une certaine primauté aux états mentaux des agents par rapport aux processus de raisonnement strictement rationnel.
L'architecture actuelle présentée dans cet article a été implémentée et interfacée avec la chaîne de traitement de requêtes en langue naturelle (sous Mathematica). Toutefois, l'impact effectif de cette approche, en particulier dans le contexte des agents assistants, reste à évaluer dans des travaux futurs qui feront intervenir des usagers novices placés face à trois classes d'agents :
– S1. Un agent purement rationnel ;
– S2. Un agent rationnel et subjectif, implémenté au moyen de l'architecture définie dans la section 2 ;
– S3. Un agent rationnel et subjectif, incluant également les biais introduits dans la section 3.

Nous nous attendons à une amélioration, en termes de réalisme, en passant de S1 à S2 ainsi que de S2 à S3. Il pourrait aussi y avoir un léger accroissement de la compétence entre S1 et S2. Cependant il est probable que l'introduction des biais conduise à une décroissance du degré de compétence perçu, ce qui pose la question du choix difficile entre compétence et réalisme.

# Références

[1] Joseph Bates. The role of emotion in believable agents. *Commun. ACM*, 37(7) :122–125, July 1994.

[2] François Bouchet and Jean-Paul Sansonnet. Caractérisation de requêtes d'assistance à partir de corpus. In *Actes de MFI'07*, Paris, France, May 2007.

[3] Antonio Capobianco and Noëlle Carbonell. Contextual online help : elicitation of human experts' strategies. In *Proceedings of HCI'01*, pages 824–828, New Orleans, August 2001.

[4] John M. Carroll and Mary Beth Rosson. *Paradox of the active user*, pages 80–111. MIT Press, 1987.

[5] Ana Casali, Lluís Godo, and Carles Sierra. Graded BDI models for agent architectures. In *Proceedings of CLIMA-V*, volume 3487 of *Lecture Notes in Computer Science*, pages 126–143, Lisbon, Portugal, 2004.

[6] Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth Churchill, editors. *Em-*

*bodied Conversational Agents*. MIT Press, April 2000.

[7] Mehdi Dastani. A classification of cognitive agents. In *Proceedings of Cogsci02*, pages 256—261, 2002.

[8] Rick Evertsz, Franck E. Ritter, Paolo Busetta, and Matteo Pedrotti. Realistic behaviour variation in a BDI-based cognitive architecture. Melbourne, Australia, 2008.

[9] Lewis R. Goldberg. Language and individual differences : The search for universal in personality lexicons. *Review of personality and social psychology*, 2 :141–165, 1981.

[10] Ora Lassila and Ralph R. Swick. Resource description framework (RDF) model and syntax specification. W3C recommendation, 1999.

[11] David Leray and Jean-Paul Sansonnet. Ordinary user oriented model construction for assisting conversational agents. In *CHAA'06 at IEEE-WIC-ACM Conference on Intelligent Agent Technology*, 2006.

[12] James C. Lester, Sharolyn A. Converse, Susan E. Kahler, S. Todd Barlow, Brian A. Stone, and Ravinder S. Bhogal. The persona effect : affective impact of animated pedagogical agents. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 359–366, Atlanta, Georgia, United States, March 1997. ACM.

[13] Jean-Claude Martin, Christophe d'Alessandro, Christian Jacquemin, Brian Katz, Aurélien Max, Laurent Pointal, and Albert Rilliard. 3D audiovisual rendering and Real-Time interactive control of expressivity in a talking head. In *Proc. of IVA'2007*, pages 29–36, 2007.

[14] Masahiro Mori. Bukimi no tani [The uncanny valley]. *Energy*, 7(4) :33–35, 1970.

[15] Emma Norling and Franck E. Ritter. Towards supporting psychologically plausible variability in Agent-Based human modelling. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2004.

[16] Andrew Ortony. On making believable emotional agents believable. In R. Trappl and P. Petta, editors, *Emotions in humans and artifacts*. MIT Press, Cambridge, MA, 2003.

[17] David Pereira, Eugenio Oliveira, and Nelma Moreira. Formal modelling of emotions in BDI agents. In F. Sadri and K. Satoh, editors, *Proceedings of CLIMA-VIII*, volume 5056 of *LNAI*, pages 62–81, Porto, Portugal, 2008. Springer-Verlag.

[18] Anand S. Rao and Michael P. Georgeff. Modelling rational agents within a BDI architecture. In R. Fikes and E. Sandewall, editors, *Proceedings of Knowledge Representation and Reasoning*, pages 473–484, San Mateo, CA, USA, 1991. Morgan Kaufmann.

[19] Byron Reeves and Clifford Nass. *The media equation : how people treat computers, television, and new media like real people and places*. Cambridge university press edition, 1996.

[20] David Sadek. *Attitudes mentales et interaction rationnelle : vers une théorie formelle de la communication*. Ph.D. thesis, Université de Rennes I, 1991.

[21] Aaron Sloman. Architectural requirements for human-like agents both natural and artificial. In Kerstin Dautenhahn, editor, *Human Cognition and Social Agent Technology (Advances in Consciousness Research)*. John Benjamins Publishing Co, 2000.

[22] Endel Tulving. *Elements of episodic memory*. Clarendon Press, Oxford, England, 1983.

# Contribution aux comparaisons formelles des modèles de préférence en argumentation

Jean-Rémi Bourguet†‡ bourgujr@supagro.inra.fr
Leila Amgoud⋆ amgoud@irit.fr
Rallou Thomopoulos†‡ rallou@supagro.inra.fr

†INRA, Ingénierie des Agropolymères et des Technologies Emergentes (IATE), UMR1208, Montpellier, France

⋆CNRS, Institut de Recherche en Informatique (IRIT), UMR5505, Toulouse, France

‡CNRS et UMII, Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (LIRMM), UMR5506, Montpellier, France

**Résumé :**

L'argumentation est un modèle de raisonnement basé sur la construction et l'évaluation d'arguments. Dans son papier fondateur, Dung a proposé le plus abstrait des cadres de travail en argumentation. Les arguments y possèdent une même force, donnant à cette hypothèse une portée contraignante voire irréaliste dans de nombreux cas d'étude. En conséquence, plusieurs extensions ont été proposées, tour à tour, dans la littérature. Le premier modèle permet la représentation de la préférence entre arguments par la présence d'un préordre (partiel ou total), permettant de ce fait de capturer une différence de forces entre arguments. La source de cette relation de préférence n'est pas spécifiée, elle peut donc être instanciée de différentes manières. La seconde extension imagine que la force d'un argument dépend de la valeur promue par cet argument. Enfin la troisième extension fait état d'un ensemble d'arguments équipé de plusieurs préordres, chacun d'eux exprimant des préférences entre arguments dans un contexte donné. La contribution de ce papier a deux effets scientifiques sous-jacents, l'un méthodologique qui propose une étude comparative des trois extensions du cadre de travail en argumentation développé par Dung, montrant expressément sous quelles conditions deux propositions sont équivalentes ; l'autre intégrationel puisqu'il constitue une proposition de modèle fédérant les aspects expressif et générique présents dans ces extensions aux travaux de Dung.

**Mots-clés :** argumentation, cadre de travail, préférence

**Abstract:**

Argumentation is a reasoning model based on the construction and the evaluation of arguments. In his seminal paper, Dung has proposed the most abstract argumentation framework. In that framework, arguments are assumed to have the same strength. This assumption is unfortunately strong and not realistic. Consequently, three different extensions of the framework have been proposed in the literature.

The contribution of this paper is two-fold : First, it proposes a comparative study of the three extensions of Dung's framework. It clearly shows under which conditions two proposals are equivalent. The second contribution of the paper consists in integrating the three extensions into a common more expressive and generic framework.

**Keywords:** argumentation, framework, preference

## 1 Introduction

L'argumentation est un modèle de raisonnement basé sur la construction et l'évaluation des arguments en interaction. Ce modèle a notamment pu être étudié dans le cadre du raisonnement non monotone (e.g. [6]), en prise de décision (e.g. [3, 5, 7]) ainsi que pour modéliser différents types de dialogues incluant la négotiation notamment (e.g. [9, 10]). Une grande partie des modèles développés pour les applications se sont donc appuyés sur les travaux et le cadre de travail proposés par Dung dans [6]. Ce cadre de travail consiste en un ensemble d'arguments et une relation binaire sur cet ensemble, exprimant les conlits entre arguments. Un argument donne raison de croire en une affirmation, afin éventuellement de réaliser une action. L'une des caractéristiques de ce type d'approche est de considérer tous les arguments présents comme ayant la même force. Cette hypothèse paraît quasi fausse puisqu'il semblerait naturel de considérer un argument comme construit à partir de plusieurs informations, certaines pouvant apparaître avec plus ou moins de poids sur d'autres (ceci pouvant provenir par exemple de la présence d'informations défectibles). En conséquence, trois différentes extensions à ce cadre de travail on été proposées dans la littérature. Ainsi, dans [1], il est proposé d'ajouter à cette relation de conflit entre arguments, une autre relation binaire sur l'ensemble des arguments (appelée relation de préférence). Cette relation capture les différences de forces entre arguments, sa source n'étant pas spécifiée, lui permettant ainsi d'être instanciée de différentes manières. Dans [4], il est supposé que la force d'un argument dépend de la valeur qu'il promeut, chaque argument étant assuré de promouvoir au plus une valeur, n'ayant pas forcément la même importance que les autres promues par ailleurs. Ainsi, il va de soi qu'un argument pro-

mouvant la plus importante valeur est considéré comme au moins aussi fort que les autres. Enfin dans [2], il est acté que l'ensemble des arguments est équipé de plusieurs préordres, chacun d'eux exprimant des préférences entre arguments dans un contexte qu'il ne reste alors plus qu'à instancier. Par exemple, pour deux arguments $\alpha$ et $\beta$, $\alpha$ est préféré à $\beta$ dans un contexte donné, et $\beta$ est préféré à $\alpha$ dans un autre contexte. Cette extension permet donc de généraliser le modèle basé sur les préférences défini dans [1]. Il paraît donc important de comparer formellement (voir sémantiquement) ces trois extensions puis de paraphraser les similarités et différences entre ces approches.
La contribution scientifique de ce papier est donc bicéphale : en premier lieu il s'agit de proposer une étude comparative des trois extensions au cadre de travail défini par Dung, montrant sous quelles types de conditions deux propositions peuvent être envisagées comme équivalentes. La seconde contribution consiste en une intégration des trois extensions dans un cadre de travail commun et plus expressif.
Le reste de l'article est organisé comme suit : la partie 2 rappelle brièvement le cadre de travail de Dung aussi bien que ses trois extensions. La partie 3 présente une étude comparative des trois extensions. La partie 4 propose un cadre de travail unifié, lequel capture les caractéristiques de ces trois extensions et tente d'unifier les approches comparatives des travaux de [8]. Enfin, l'article se conclut en partie 5.

## 2 Rappel concernant quelques cadres de travail abstraits en argumentation

Cette partie rappelle brièvement le cadre abstrait de travail en argumentation suggéré par Dung, mais aussi les trois extensions intégrant des préférences dans ce cadre. Ceux-ci seront illustrés par un exemple, à travers lequel seront soulignées les forces et les faiblesses de chacun d'eux.

### 2.1 Cadre de travail abstrait de Dung

Un processus d'argumentation suit trois étapes principales : 1) la construction d'arguments et de contre arguments 2) l'évaluation de l'acceptabilité des différents arguments, et 3) la conclusion ou la définition des conclusions fondées. Dans [6], un cadre de travail en argumentation est défini comme suit :

**Définition 1 (Cadre de travail de Dung)** *Un* cadre de travail en argumentation *est une paire* $\mathtt{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$ *où* $\mathcal{A}$ *est un ensemble d'arguments et* $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ *est une relation d'*attaque. *Un argument* $\alpha$ *attaque un argument* $\beta$ *ssi* $(\alpha, \beta) \in \mathcal{R}$.

Dans la définition précédente, les arguments sont des entités abstraites, leurs origines et leurs structures demeurent inconnues. Il est à noter que peut être associé, à chaque système d'argumentation, un graphe orienté dans lequel les noeuds constituent les différents arguments, les arêtes représentent les relations d'attaque entre eux.
Parmi tous les arguments conflictueux, il faut définir quels arguments le système peut garder pour inférer des conclusions ou prendre des décisions. Dans [6], différentes sémantiques répondant à la notion d'acceptabilité ont été proposées. Pour la compréhension générale du papier, plusieurs notions concernant l'acceptabilité en argumentation sont rappelées ci-après.

**Définition 2 (Sans $\mathcal{R}$-conflit , Défense)** *Soit* $\mathcal{B} \subseteq \mathcal{A}$.
- $\mathcal{B}$ *est* sans $\mathcal{R}$-conflit *ssi* $\nexists\, \alpha_i, \alpha_j \in \mathcal{B}$ *tel que* $(\alpha_i, \alpha_j) \in \mathcal{R}$.
- $\mathcal{B}$ défend *un argument* $\alpha_i$ *ssi pour chaque argument* $\alpha_j \in \mathcal{A}$, *si* $(\alpha_j, \alpha_i) \in \mathcal{R}$, *alors* $\exists\, \alpha_k \in \mathcal{B}$ *tel que* $(\alpha_k, \alpha_j) \in \mathcal{R}$.
- *Un ensemble sans* $\mathcal{R}$*-conflit* $\mathcal{B}$ *d'arguments est une* extension admissible *ssi* $\mathcal{B}$ *défend tous ses éléments.*

Illustrons le cadre de travail abstrait à travers un exemple simple, qui décrit une situation de décision multi-critères. L'idée est de choisir entre deux étudiants, Rémi et Jean, sur la base de critère de pertinence, d'originalité et d'utilité.

**Exemple 1** *Le Tableau résume les notes obtenues par les deux étudiants dans les différents critères.*

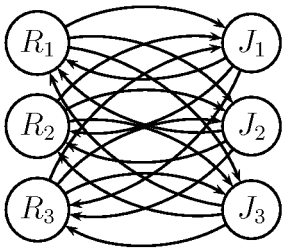
| | *Utilité* | *Pertinence* | *Originalité* |
|---|---|---|---|
| *Rémi* | *4* | *2* | *4* |
| *Jean* | *2* | *4* | *3* |

*Dans cette application, un argument donne une raison pour choisir tel ou tel étudiant. Par exemple, Rémi devrait être choisi puisqu'il obtient 4 en utilité. Il est a noté que toutes les notes*

*sont au moins égale ou plus haute que 2, permettant ainsi à tous les arguments d'être en faveur de l'un des deux candidats. Les six arguments suivants sont ainsi construits :*

*$R_1$ : Rémi devrait être choisi en obtenant 4 en Utilité.*
*$J_1$ : Jean devrait être choisi en obtenant 2 en Utilité.*
*$R_2$ : Rémi devrait être choisi en obtenant 2 en Pertinence.*
*$J_2$ : Jean devrait être choisi en obtenant 3 en Pertinence.*
*$R_3$ : Rémi devrait être choisi en obtenant 4 en Originalité.*
*$J_3$ : Jean devrait être choisi en obtenant 3 en Originalité.*

*Puisque seulement un candidat devrait être choisi, toute paire d'argument qui ne supporte pas le même candidat est considérée comme conflictueuse : les arguments présents dans ces paires sont donc engagés dans une relation d'attaque symétrique. La figure suivante résume les différents conflits entre arguments.*




*Le système possède deux extensions maximales admissibles (en terme d'inclusion ensembliste) : $\{R_1, R_2, R_3\}$ et $\{J_1, J_2, J_3\}$. Il est donc notifié par le système de Dung que les deux candidats Jean et Rémi sont similairement préférés.*

L'exemple montre que le cadre de travail utilisé pour capturer les argumentaires n'est pas très judicieux. En fait, dans l'exemple, il est clair qu'un candidat est meilleur qu'un autre puisqu'il possède plus de note plus élevées. Ainsi, puisque le cadre de travail ne permet pas de prendre en compte les forces des arguments, celui-ci a seulement permis de résoudre les conflits entre arguments et donc de conclure sur une acceptabilité similaire pour les deux candidats. Par la nécessité d'établir des comparaisons entre cadres de travail utilisés en argumentation, il va être défini la notion de cadres de travail équivalents. Deux cadres de travail en argumentation sont dits équivalents si ceux-ci retournent exactement les même extensions sous une sémantique donnée.

**Définition 3 (Cadres de travail équivalents)** *Soit $\mathtt{AF}_1$, $\mathtt{AF}_2$ deux cadres de travail en argumentation. $\mathtt{AF}_1$ et $\mathtt{AF}_2$ sont équivalents ssi Ext($\mathtt{AF}_1$) = Ext($\mathtt{AF}_2$), où Ext($\mathtt{AF}_i$) est l'ensemble de toutes les extensions de $\mathtt{AF}_i$ sous une sémantique donnée.*

## 2.2 Cadre de travail en argumentation basé sur la préférence

Dans [1], il est spécifié, au sein d'un cadre de travail en argumentation, une relation de préférence permettant la comparaison entre arguments et dans certains cas une sélection entre arguments conflictueux. Ainsi, dans l'exemple précédent, il est clair que l'argument $R_1$ est plus fort que $J_1$, l'argument $J_2$ est plus fort que l'argument $R_2$ et l'argument $R_3$ est plus fort que l'argument $J_3$. Ces informations doivent donc pouvoir être exploitées dans un système d'argumentation, permettant de ce fait de réduire le nombre d'attaques entre arguments. L'idée est donc simplement de voir une attaque échouer lorsque l'argument attaqué est plus fort que son attaquant.

Dans cette optique, un PAF est défini à partir d'un tuple $\langle \mathcal{A}, \mathcal{R}, \succeq \rangle$ où $\mathcal{A}$ est un ensemble d'arguments, $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ est une relation d'attaque et $\succeq \subseteq \mathcal{A} \times \mathcal{A}$ est un préordre (partiel ou total) qui représente une relation de préférence entre arguments. Cette relation est générale et peut être instanciée de différentes manières.

**Définition 4 (Preference-based argumentation)** *Un* PAF *est une paire* $\mathtt{PAF} = \langle \mathcal{A}, \mathtt{Def} \rangle$ *où* $\mathtt{Def} \subseteq \mathcal{A} \times \mathcal{A}$ *tel que* $(a, b) \in \mathtt{Def}$ *ssi* $(a, b) \in \mathcal{R}$ *et* $(b, a) \notin \succ$ [1].

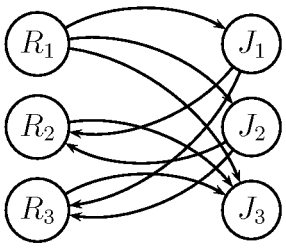
Par la suite, PAF pourra être appréhendé avec ses composants initiaux, autrement dit le triplet $\langle \mathcal{A}, \mathcal{R}, \succ \rangle$. Les sémantiques de Dung sont ainsi appliquées au nouveau cadre de travail $\langle \mathcal{A}, \mathtt{Def} \rangle$ pour évaluer les arguments de $\mathcal{A}$. Il est à noter que losque tous les arguments sont incomparables ou indifférents au regard de la relation $\succeq$, le cadre de travail de Dung est retrouvé.

**Propriété 1** *Les deux cadres de travail* $\mathtt{PAF}_1 = \langle \mathcal{A}, \mathcal{R}, \succ \rangle$ *et* $\mathtt{PAF}_2 = \langle \mathcal{A}, \mathcal{R}, \succeq \rangle$ *(avec $\succ$ [2] la relation stricte de $\succeq$) sont équivalents.*

1. $(a, b) \in \succ$ ssi $(a, b) \in \succeq$ et $(b, a) \notin \succeq$.
2. $\succ$ est irréflexive, antisymétrique et transitive.

Il paraît, à présent, intéressant de reconsidérer l'Exemple 1 et ainsi observer comment les préférences entre arguments aideront à réduire le nombre d'attaques et à retourner le résultat attendu.

**Exemple 2 (Exemple 1 cont.)** *Comme il a été suggéré précédemment, les préférences suivantes entre arguments peuvent être avérées :* $R_1 \succ J_1$, $J_2 \succ R_2$ *et* $R_3 \succ J_3$. *Une autre source de préférence entre arguments est l'importance laissée aux critères d'évaluation. Considérons, par exemple, le critère d'utilité comme plus important que le critère de pertinence et le critère de pertinence comme plus important que le critère d'originalité, en d'autres termes* $\{R_1, J_1\} \succ \{R_2, J_2\} \succ \{R_3, J_3\}$. *Le graphe de* $\langle \mathcal{A}, \texttt{Def} \rangle$ *est résumé ci-après.*




*Le cadre travail possède seulement une extension admissible maximale (en terme d'inclusion ensembliste), qui est* $\{R_1, R_2, R_3\}$. *Il est donc clair que Rémi est préférré à Jean puisque cette extension contient seulement les arguments supportant Rémi.*

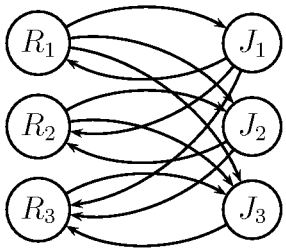
## 2.3 Cadre de travail en argumentation basé sur les valeurs

Dans [4], Bench Capon procède à une formalisation des idées du philosophe Perelman. Ce dernier souligne l'importance de promouvoir des valeurs au travers d'arguments. En d'autres termes, un argument peut promouvoir une valeur telle que, par exemple, la santé, l'économie, le partage... Ainsi, un $VAF$ peut être défini comme suit :

**Définition 5 (Value-based argumentation)** *Un VAF est une paire* $\texttt{VAF} = \langle \mathcal{A}, \texttt{defeats} \rangle$ *où* $\mathcal{A}$ *est un ensemble d'arguments,* $\texttt{defeats} \subseteq \mathcal{A} \times \mathcal{A}$ *tel que* $(a, b) \in \texttt{defeats}$ *ssi* $(a, b) \in \mathcal{R}$ *et* $(b, a) \notin \texttt{Pref}$*(*$\texttt{val}(b)$*,*$\texttt{val}(a)$*),*
*avec* $\mathcal{V}$ *vu comme un ensemble de valeurs,* $\texttt{val}$ *:* $\mathcal{A} \mapsto \mathcal{V}$ *et* $\texttt{Pref} \subseteq \mathcal{V} \times \mathcal{V}$ *une relation stricte de préférence étant irréflexive, antisymmétrique et transitive.*

Par la suite, VAF pourra être appréhendé par ses composants classiques, comme un tuple $\langle \mathcal{A}, \mathcal{R}, \mathcal{V}, \texttt{val}, \texttt{Pref} \rangle$. A l'image du PAF, les sémantiques d'acceptabilité sont appliquées pour évaluer les différents arguments. Illustrons à présent ce cadre de travail à travers l'exemple. Pour cela, il faut absolument définir ce que seront les valeurs aussi bien que les relation de préférences entre ces valeurs. Il y a deux possibilités : la première consiste à considérer les différents critères comme des valeurs. La seconde considère les notes obtenues (de 0 à 5) comme possibles valeurs. Il est à noter qu'il n'est pas viable de fusionner les deux ensembles puisque l'ensemble $\mathcal{V}$ n'aurait alors aucun sens (dans le sens aucune cohérence). Dans ce qui suit, les deux solutions présentées ci-dessus seront étudiées au travers de l'exemple et le choix d'attaque entre arguments sera basé uniquement sur l'option, montrant ainsi que ce cadre de travail est insuffisant pour obtenir le résultat attendu. Nous aborderons cependant une autre façon de bien traiter ce cas par ce cadre de travail, en suggérant une construction différente de relations d'attaque entre arguments.

**Exemple 3 (Exemple 1 cont.)** *Supposons que* $\mathcal{V}_1$ = *{Utilité, Pertinence, Originalité} tel que le critère d'utilité soit plus important que le critère de pertinence et que ce dernier soit plus important que le critère d'originalité. Ainsi,* $\texttt{Pref}_1$ = *{(Utilité,Pertinence), (Pertinence,Originalité), (Utilité,Originalité)}. La fonction* $\texttt{val}_1$ *se définit comme suit :* $\texttt{val}_1(R_1)$ = $\texttt{val}_1(J_1)$ = *Utilité,* $\texttt{val}_1(R_2)$ = $\texttt{val}_1(J_2)$ = *Pertinence, et* $\texttt{val}_1(R_3)$ = $\texttt{val}_1(J_3)$ = *Originalité. Le graphe associé au cadre de travail* $\texttt{VAF}_{ex1} = \langle \mathcal{A}, \texttt{defeats}_1 \rangle$ *est représenté ci-dessous :*



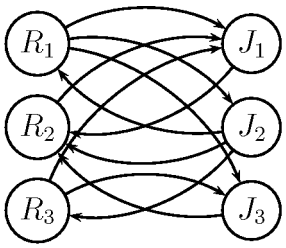

*Le cadre de travail possède sur cet exemple deux ensembles maximaux d'arguments :* $\{R_1, R_2, R_3\}$ *et* $\{J_1, J_2, J_3\}$. *Ainsi, à l'image*

*du résultat obtenu avec le cadre de travail proposé par Dung, ici les deux candidats Jean et Rémi sont préférés de façon similaire. Notons également, que les valeurs et les préférences sur ces valeurs n'ont aucun impact dans ce cas sur le résultat.*

Considérons à présent le cas où les valeurs capturent les différentes notes évaluatives.

**Exemple 4 (Exemple 1 cont.)** *Dans ce cas,* $\mathcal{V}_2 = \{1, \ldots, 5\}$ *(lequel n'exprime pas à proprement dit de valeurs* sociales*, comme imaginé dans les écrits de Perelman). Les préférences entre valeurs sont donc naturellement obtenus par un ordonnancement classique entre entiers. Par exemple, (4, 2)* $\in$ $\texttt{Pref}_2$*. Les valeurs des six arguments sont :* $\texttt{val}_2(R_1)$*= 4,* $\texttt{val}_2(J_1)$*= 2,* $\texttt{val}_2(R_2)$*= 2,* $\texttt{val}_2(J_2)$*= 4,* $\texttt{val}_2(R_3)$*= 4, et* $\texttt{val}_2(J_3)$*= 3.*

*Le graphe associé au cadre de travail* $\texttt{VAF}_{ex2} = \langle \mathcal{A}, \texttt{defeats}_2 \rangle$ *est représenté ci-dessous :*




*Le cadre de travail* $\langle \mathcal{A}, \texttt{defeats}_2 \rangle$ *possède également au travers de cet exemple deux ensembles maximaux admissible d'arguments :* $\{R_1, R_2, R_3\}$ *et* $\{J_1, J_2, J_3\}$*, ne permettant pas de conclure sur une quelconque préférence de candidats.*

Enfin, quand les valeurs sont associées aux critères d'évaluation d'utilité, de pertinence, et d'originalité, il peut être intuitif, dans ce cas d'application, de permettre à deux arguments supportant la même valeur d'être engagés dans une relation d'attaque non symétrique : ainsi l'argument $R_1$ attaque l'argument $J_1$ puisque $4 > 2$, alors que l'inverse n'est pas vrai. En d'autres termes, si deux arguments se réfèrent au même critère d'évaluation, ceux-ci peuvent être comparés en terme de notation, alors que s'ils se réfèrent à des critères d'évaluation différents, leurs préférences sont déterminées par les valeurs. Ainsi, le graphe de l'exemple 4 devient le même que celui de l'exemple 2, Rémi étant préféré à Jean.

Ce type de représentation du problème va ceci dit à l'encontre d'une certaine forme d'automaticité dans le raisonnnement argumentatif, en ces termes que deux arguments supportant deux options différentes ne sont donc pas nécessairement engagés dans une relation d'attaque symétrique, ce qui parait à contrario contre-intuitif.

## 2.4 Cadre de travail en argumentation basé sur les préférences contextuelles

Dans les travaux concernant le PAF et le VAF, les préférences entre arguments sont supposées comme étant non contradictoires ; cependant, dans les applications courantes, il peut arriver que cela ne soit pas toujours le cas. Considérons le cas de notre exemple courant. Reconsidérons la note de Rémi en utilité ($R'_1$), admettons qu'il obtienne 2. Ainsi, puisque l'utilité est plus importante que la pertinence, $R'_1$ peut être considéré comme étant plus fort que $J_2$.

Cependant, puisque la notation 4 est plus important que la notation 2, il parait également rationnel de considérer que $J_2$ est plus fort que $R'_1$. Dans [2], une extension de PAF a été proposée, l'idée étant d'y assumer que l'ensemble $\mathcal{A}$ des arguments peut être equipé de plusieurs relations de préférence $\succeq_1, \cdots \succeq_n$, chacun d'elle exprimant des préférences non conflictuelles entre arguments dans un *contexte* particulier, contextes qui peuvent être des valeurs, des critères, ou des agents, ..., sont supposés être ordonnés par une relation binaire complète et stricte dénoté par $\triangleright$.

Notons que pour deux arguments $a$ et $b$, il peut être possible que $(a, b) \in \succeq_i$ et $(b, a) \in \succeq_j$ avec $i \neq j$. Les différentes relations de préférence $\succeq_i$ peuvent être aggrégées au sein d'une unique relation dénotée par $\otimes(\succeq_1, \cdots \succeq_n)$. Dans [2], une proposition immédiate est de garder toutes les préférences du contexte le plus fort, et d'ajouter les préférences des contextes suivants qui ne sont pas en contradiction avec celles présentes dans le premier. Le même procédé est répété jusqu'à qu'il n'y ait plus de contextes restant.

**Définition 6 (Fonction d'aggrégation)** *Soit* $\mathcal{C} = \{c_1, \ldots, c_n\}$ *un ensemble de contextes.* $\otimes(\succeq_1, \cdots \succeq_n) = \texttt{Pref}^n$ *tel que*
– $\mathcal{C}^1 = \mathcal{C}$

- $\texttt{Pref}^1 = \{(a,b) \in \succeq_i$ *t.q.* $\forall c_j \in \mathcal{C}^1 \setminus \{c_i\}, c_i \rhd c_j\}$
- $\mathcal{C}^{k+1} = \mathcal{C}^k \setminus \{c_i\}$
- $\texttt{Pref}^{k+1} = \texttt{Pref}^k \cup \{(a,b) \in \succeq_i$, *t.q.* $c_i \in C^{k+1}$, *et* $(b,a) \notin C^k$, *et* $\forall c_j \in \mathcal{C}^{k+1} \setminus \{c_i\}, c_i \rhd c_j\}$.

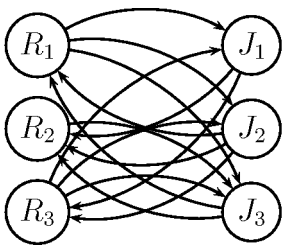
Il est utile de préciser que la relation $\otimes(\succeq_1, \cdots \succeq_n)$ n'est pas nécessairement transitive. Ainsi, un cadre de travail en argumentation basé sur des préférences contextuelles (CPAF) est défini comme suit :

**Définition 7 (CPAF)** *Soit $\mathcal{A}$ un ensemble d'arguments, $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ une relation d'attaque, $\mathcal{C} = \{c_1, \ldots, c_n\}$ un ensemble de contextes, $\rhd$ une relation stricte et complète sur $\mathcal{C}$, $\succeq_i$ un préordre (partiel ou complet) sur $\mathcal{A}$ issu du contexte $c_i$.*

*Un* CPAF *est une paire $\langle \mathcal{A}, \texttt{Def} \rangle$, où* Def *est défini comme suit : $\forall a, b \in \mathcal{A}$, $(a,b) \in$ Def ssi $(a,b) \in \mathcal{R}$ et $(b,a) \notin \otimes(\succeq_1, \cdots \succeq_n)$.*

Par la suite, un CPAF pourra être représenté par ses composants initiaux, autrement dit, tel un tuple $\langle \mathcal{A}, \mathcal{R}, \mathcal{C}, \rhd, \succeq_1, \cdots \succeq_n \rangle$. Les arguments de $\mathcal{A}$ sont évalués en utilisant les sémantiques d'acceptabilité de Dung appliquées au CPAF.

**Exemple 5 (Example 1. cont.)** *Soit $\mathcal{C}_1$ = {Utilité, Pertinence, Originalité} avec Utilité $\rhd$ Pertinence $\rhd$ Originalité. L'ensemble des arguments est équipé de trois relations de préférence, respectivement denotées $\succeq_U, \succeq_P, \succeq_O$. Ces relations sont définis comme suit : $\succeq_U = \{(R_1, J_1)\}$, $\succeq_P = \{(J_2, R_2)\}$, et $\succeq_O = \{(R_3, J_3)\}$. La relation de préférence aggrégée est dans ce cas l'union des trois relations. Le graphe associé à ce cadre de travail* CPAF *est représenté ci-dessous :*




*Il y a deux ensemble maximaux admissibles : $\{R_1, R_2, R_3\}$ et $\{J_1, J_2, J_3\}$. Les deux candidats Jean et Rémi sont donc similairement préférés dans ce cadre de travail.*

# 3 Comparaison des différents cadres de travail abstraits en argumentation

Dans cette partie, sont comparés en termes d'équivalence les différents cadres de travail en argumentation précédemment présentés, sur la base de la définition 3 proposée.

## 3.1 Comparaison du cadre de travail de Dung et du PAF

Le cadre de travail en argumentation de Dung peut être vu comme un cas particulier du cadre de travail basé sur les préférences (PAF). En dépit du fait que leurs contenus puissent paraître complètement évident pour tout néophyte de la priorité en argumentation, il a été spécifié quelques conditions sous lesquelles ces deux cadres de travail peuvent être entrevus comme équivalents (au sens de la définition 3). Ainsi, une première situation dans laquelle PAF et AF sont équivalents est précisément le cas où il n'existe aucune préférence stricte entre arguments.

**Proposition 1** *Soit* AF $= \langle \mathsf{Arg}, \mathcal{R} \rangle$ *et* PAF $= \langle \mathsf{Arg}, \mathcal{R}, \succeq \rangle$. *Si $\nexists\ \alpha, \beta$ tel que $(\alpha, \beta) \in \succ$ alors* AF *et* PAF *sont équivalents.*

Il existe également un second cas, où toutes les attaques entre arguments aboutissent dans le sens que si un argument $\alpha$ attaque un argument $\beta$ alors $\beta$ n'est pas préféré à $\alpha$.

**Proposition 2** *Soit* AF $= \langle \mathsf{Arg}, \mathcal{R} \rangle$ *et* PAF $= \langle \mathsf{Arg}, \mathcal{R}, \succeq \rangle$. *Si $\nexists\ \alpha, \beta \in \mathsf{Arg}$, tel que $(\alpha,\beta) \in \mathcal{R}$ et $(\beta, \alpha) \in \succ$ alors* AF *et* PAF *sont équivalents.*

## 3.2 Comparaison entre le VAF et le PAF

Dans cette partie, et dans le but de mener une étude comparative de VAF et de PAF, il sera successivement montré qu'une équivalence de PAF peut être construite à partir d'un VAF, puis que plusieurs familles de VAF existent et permettent de représenter un PAF.

**COnstruction d'un équivalent $\texttt{PAF}^V$ à partir d'un VAF :.**

**Définition 8** *Soit* VAF $= \langle \mathcal{A}, \mathcal{R}, \mathcal{V}, val, \texttt{Pref} \rangle$, *un cadre de travail en argumentation basé sur*

*les valeurs. A partir d'un* VAF*, un cadre de travail en argumentation basé sur les préférences peut être défini par* $\texttt{PAF}^V = \langle \mathcal{A}, \mathcal{R}, \succeq_V \rangle$*, avec la relation de préférence* $\succeq_V \subseteq \mathcal{A} \times \mathcal{A}$ *défini comme suit :* $\forall a, b \in \mathcal{A}$, $(a, b) \in \succ_V$ *ssi* Pref*(val(a), val(b)).*

Cette définition est illustrée à travers l'exemple suivant :

**Exemple 6 (Exemple 1. cont.)** *La relation de préférence extraite du cadre de travail* $\texttt{VAF}_{ex1}$ *est décrite comme suit :* $\succeq_{v1}$ = {($R_1$, $J_2$), ($R_1$, $J_3$), ($R_2$, $J_3$), ($J_1$, $R_2$), ($J_1$, $R_3$), ($J_2$,$R_3$)}. *Le système* $\texttt{PAF}^V_{ex1}$*, construit avec* $\succeq_{v1}$*, possède deux extensions préférées* $\{R_1, R_2, R_3\}$ *et* $\{J_1, J_2, J_3\}$*, similairement à* $\texttt{VAF}_{ex1}$*. De la même façon, la relation de préférence extraite du cadre de travail* $\texttt{VAF}_{ex2}$ *est décrite ainsi :* $\succeq_{v2}$ = {($R_1$, $J_1$), ($J_2$, $R_2$), ($R_3$, $J_3$), ($R_3$, $J_1$), ($R_3$, $J_2$), ($J_3$, $R_2$)}. *Le système* $\texttt{PAF}^V_{ex2}$*, construit avec* $\succeq_{v2}$*, possède seulement une extension préférée qui n'est autre que* $\{R_1, R_2, R_3\}$*, similairement à* $\texttt{VAF}_{ex2}$.

La relation $\succ_V$ possède les même propriétés que la relation Pref.

**Propriété 2** *La relation* $\succ_V$ *est irreflexive, antisymmétrique et transitive.*

Au sens mathématique du terme toute assertion démontrée peut prendre le nom de théorème. Cependant, dans les ouvrages concernant des domaines spécialisés, il est d'usage de réserver ce terme aux affirmations considérées comme particulièrement intéressantes et/ou importantes ; ainsi, dans la proposition suivante VAF et $\texttt{PAF}^v$ sont énoncés comme équivalents.

**Proposition 3** *Soit* $\langle \mathcal{A}, \mathcal{R}, \mathcal{V}, val, \texttt{Pref} \rangle$*, un* VAF*, et* $\succeq_V$ *une relation préférence extraite de* VAF *au sens de la définition 8.* $\texttt{PAF}^V = \langle \mathcal{A}, \texttt{Def}_V \rangle$ *est défini à partir de* $\succeq_V$ *et* $\mathcal{R}$ *selon la définition 4. Les deux systèmes* VAF *et* $\texttt{PAF}^V$ *sont donc équivalents.*

**Construction de plusieurs famille d'équivalence de $\texttt{VAF}^P$ à partir d'un PAF :.**

**Famille bijective** Un $\texttt{VAF}^P_b$ peut être intuitivement construit à partir d'un PAF, et ce en affectant à chaque argument une valeur différente, puis en respectant l'ordonnancement des valeurs, à l'image (bijective) de celui des arguments.

**Définition 9** *Soit* $\texttt{PAF} = \langle \mathcal{A}, \mathcal{R}, \succeq \rangle$ *un système d'argumentation à base de préférence. Un* $\texttt{VAF}^P_b$ *défini à partir d'un* PAF *est un tuple* $\langle \mathcal{A}^P, \mathcal{R}^P, \mathcal{V}_b, val_b, \texttt{Pref}^P_b \rangle$ *tel que :* $\mathcal{A}^P = \mathcal{A}$, $\mathcal{R}^P = \mathcal{R}$, $\mathcal{V}_b$ *est un ensemble de valeurs possédant la même cardinalité que* $\mathcal{A}^P$ ($|\mathcal{V}_b| = |\mathcal{A}^P|$), $val_b$ *est une bijection de* $\mathcal{A}^P$ *dans* $\mathcal{V}_b$*, et* $\texttt{Pref}^P_b \subseteq \mathcal{V}_b \times \mathcal{V}_b$ *est défini par :* $\forall\, v_1, v_2 \in \mathcal{V}_b$, ($v_1$, $v_2$) $\in \texttt{Pref}^P_b$ *ssi* ($val_b^{-1}(v_1)$, $val_b^{-1}(v_2)$) $\in \succ$ ($val_b^{-1}$ *dénote la fonction inverse de* $val_b$*).*

Ainsi, à partir du PAF présenté dans l'exemple 2, $\texttt{VAF}^P_b$ peut être construit en associant chaque argument $(R_1, J_1, \ldots)$ avec une valeur, par exemple son nom ("$R_1$", "$J_1$", ...), conservant la même préférence pour les noms comme pour les arguments sous-jacents.
La relation $\texttt{Pref}^P_b$ possède les même propriétés que la relation $\succ$.

**Propriété 3** *La relation* $\texttt{Pref}^P_b$ *est irréflexive, antisymmétrique et transitive.*

Ainsi, dans la proposition suivante, $\texttt{PAF}^v$ et $\texttt{VAF}^P_b$ seront énoncés comme équivalents.

**Proposition 4** *Soit* $\texttt{PAF} = \langle \mathcal{A}, \mathcal{R}, \succeq \rangle$ *et* $\texttt{VAF}^P_b = \langle \mathcal{A}^P, \mathcal{R}^P, \mathcal{V}_b, val_b, \texttt{Pref}^P_b \rangle$*.* PAF *et* $\texttt{VAF}^P_b$ *issus de la définition 9 sont équivalents.*

Il peut être intéressant à ce stade de définir une relation entre valeurs présentes dans un ensemble d'arrivée $\mathcal{V}_b$ d'un système d'argumentation à base de valeur $\texttt{VAF}^P_b$ issu d'un système d'argumentation à base de préférences PAF. Cette relation appelée équivalence typologique, dénotée Et est défini comme suit :

**Définition 10** *Soit deux valeurs* $v_i$, $v_j \in \mathcal{V}_b$ *et* Et *une relation d'équivalence typologique,* ($v_i$, $v_j$) $\in$ Et *ssi :*
- $\forall\, v_k \in \mathcal{V}_b$ *t.q.* ($val_b^{min-1}(v_k)$, $val_b^{min-1}(v_i)$) $\in \succ$ *alors* ($val_b^{min-1}(v_k)$, $val_b^{min-1}(v_j)$) $\in \succ$,

- *si* $\forall v_l \in \mathcal{V}_b$ *t.q.* $(val_b^{min-1}(v_i), val_b^{min-1}(v_l)) \in \succ$ *alors* $(val_b^{min-1}(v_j), val_b^{min-1}(v_l)) \in \succ$,

Les propriété suivante de cette relation peuvent être démontrées.

**Propriété 4** *La relation* Et *est réflexive, symétrique et transitive.*

La relation Et définit donc une relation d'équivalence dans l'ensemble $\mathcal{V}_b$. On peut alors définir des classes d'équivalence sur l'ensemble $\mathcal{V}_b$ muni de la relation d'équivalence Et.

**Définition 11** *La classe d'équivalence d'un élément v de* $V_b$ *muni de la relation d'équivalence* Et, *notée* Et*(v), est alors l'ensemble des images de v par* Et *:* $\texttt{Et}(v) = \{v' \in V_b \mid (v, v') \in \texttt{Et}\}$.

L'ensemble quotient permet de regrouper en un seul ensemble, les ensembles de valeurs situées dans la même classe d'équivalence.

**Définition 12** *L'ensemble quotient de* $V_b$ *par la relation d'équivalence* Et, *noté* $V_b / \texttt{Et}$, *est l'ensemble des classes d'équivalence de* $V_b$ *suivant* Et *:* $V_b / \texttt{Et} = \{\texttt{Et}(v) \mid v \in \mathcal{V}_b\}$.

L'ensemble quotient est donc un nouvel ensemble construit à partir de $V_b$ et de Et, et ce n'est autre qu'un sous-ensemble de l'ensemble des parties de $V_b$.

**Famille surjective** Une autre façon de représenter un PAF avec un $\texttt{VAF}_s^P$ peut être d'affecter une même valeur aux arguments qui sont indifférents entre eux visà vis de la relation de préférence $\succ$.

**Définition 13** *Soit* $\texttt{PAF} = \langle \mathcal{A}, \mathcal{R}, \succeq \rangle$ *un système d'argumentation à base de préférence. Un* $\texttt{VAF}_s^P$ *défini à partir d'un* PAF *est un tuple* $\langle \mathcal{A}^P, \mathcal{R}^P, \mathcal{V}_s, val_s, \texttt{Pref}_s^P \rangle$ *tel que :* $\mathcal{A}^P = \mathcal{A}$, $\mathcal{R}^P = \mathcal{R}$, $val_s \in Val_s$, $Val_s$ *étant une famille de surjections,* $V_s$ *est une famille d'ensemble d'arrivée (correspondantes aux surjections) possédant une cardinalité inférieure ou égale à celle de* $\mathcal{A}^P$ *:*
*Enfin,* $\texttt{Pref}_s^P \subseteq \mathcal{V}_s \times \mathcal{V}_s$ *est défini par :* $(v_1, v_2) \in \texttt{Pref}_s^P$ *ssi* $\exists\, a \in (val_s^{-1}(v_1)$ *et* $\exists\, b \in val_s^{-1}(v_2))$ *tel que* $(a, b) \in \succ$ $(val_s^{-1}$ *dénote la fonction inverse de* $val_s$, *elle renvoit des ensembles d'arguments).*

La relation $\texttt{Pref}_s^P$ possède les même propriétés que la relation $\succ$.

**Propriété 5** *La relation* $\texttt{Pref}_s^P$ *est irréflexive, antisymmétrique et transitive.*

Ainsi, dans la proposition suivante, $\texttt{PAF}^v$ et $\texttt{VAF}_s^P$ sont énoncés comme équivalents.

**Proposition 5** *Soit* $\texttt{PAF} = \langle \mathcal{A}, \mathcal{R}, \succeq \rangle$ *et* $\texttt{VAF}_s^P = \langle \mathcal{A}^P, \mathcal{R}^P, \mathcal{V}_s, val_s, \texttt{Pref}_s^P \rangle$ *issu de la définition 13.* PAF *et* $\texttt{VAF}_s^P$ *sont équivalents.*

On dénote $val_s^{min}$ la surjection donnant le sous-ensemble d'arrivée $\mathcal{V}_s$ minimal (en terme d'inclusion ensembliste) appelé $\mathcal{V}_s^{min}$ pour lequel le $\texttt{VAF}_s^P$ tient dans la proposition 5 et $\texttt{VAF}_b^P$ tient dans la proposition 4 pour un même PAF.

**Propriété 6** $|\mathcal{V}_s^{min}| = |V_b / \texttt{Et}|$

**Famille injective** Ce type de $\texttt{VAF}_i^P$ constitue une extension au VAF introduit par Bench-Capon, l'argument présent dans ce système pouvant promouvoir plusieurs valeurs. Ce nouveau cadre de travail en argumentation basé sur les valeurs est introduit dans [8], la fonction $val$ y est remplacé par une fonction $arg$ (de $\mathcal{V}$ dans $2^{|A|}$). Afin de garder une certaine cohérence dans la syntaxe de nos équivalences, la fonction injective $arg$ sera remplacée dans la définition par la fonction injective $val_i$.

**Définition 14** *Un cadre de travail en argumentation basé sur les valeurs dans lequel les arguments peuvent promouvoir plusieurs valeurs est une paire* $\texttt{MVAF} = \langle \mathcal{A}, \texttt{defeats} \rangle$ *où* $\mathcal{A}$ *est un ensemble d'arguments,* $\texttt{defeats} \subseteq \mathcal{A} \times \mathcal{A}$ *tel que* $(a, b) \in$ defeats *ssi* $(a, b) \in \mathcal{R}$ *et* $(b, a) \notin \texttt{Pref}(\texttt{val}_i(b), \texttt{val}_i(a))$, *avec* $\mathcal{V}_i$ *vu comme un ensemble de valeurs,* $\texttt{val}_i : \mathcal{A} \mapsto 2^{|\mathcal{V}_i|}$ *et* $\texttt{Pref} \subseteq 2^{|\mathcal{V}_i|} \times 2^{|\mathcal{V}_i|}$ *une relation stricte de préférence étant irréflexive, antisymmétrique et transitive.*

MVAF peut également être décrit par le tuple : $\langle \mathcal{A}, \mathcal{R}, \mathcal{V}, val_i, \texttt{Pref} \rangle$ et sera appelé par la suite $\texttt{VAF}_i$ de façon à garder une certaine cohérence dans la désignation contextuelle des modèles.

**Définition 15** *Soit* $\texttt{PAF} = \langle \mathcal{A}, \mathcal{R}, \succeq \rangle$ *un système d'argumentation à base de préférence. Un*

$\texttt{VAF}_i^P$ *défini à partir d'un* PAF *est un tuple* $\langle \mathcal{A}^P, \mathcal{R}^P, \mathcal{V}_i, val_i, \texttt{Pref}_i^P \rangle$ *tel que :* $\mathcal{A}^P = \mathcal{A}$, $\mathcal{R}^P = \mathcal{R}$, $\mathcal{V}_i$ *est un ensemble de valeurs possédant une cardinalité supérieure ou égale à celle de* $\mathcal{A}^P$, $val_i$ *est une injection de* $\mathcal{A}^P$ *dans* $2^{|\mathcal{V}_i|}$ *tel que :*
$\texttt{Pref}_i^P \subseteq 2^{|\mathcal{V}_i|} \times 2^{|\mathcal{V}_i|}$ *est défini par :* $(v_1, v_2) \in \texttt{Pref}_i^P$ *ssi* $(val_i^{-1}(v_1), val_i^{-1}(v_2)) \in \succ$ *(*$val_i^{-1}$ *dénote la fonction inverse de* $val_i$*).*

La relation $\texttt{Pref}_i^P$ possède les même propriétés que la relation $\succ$.

**Propriété 7** *La relation* $\texttt{Pref}_i^P$ *est irréflexive, antisymmétrique et transitive.*

Ainsi, dans la proposition suivante, $\texttt{PAF}^v$ et $\texttt{VAF}_i^P$ ont été énoncés comme équivalents.

**Proposition 6** *Soit* $\texttt{PAF} = \langle \mathcal{A}, \mathcal{R}, \succeq \rangle$ *et* $\texttt{VAF}_i^P = \langle \mathcal{A}^P, \mathcal{R}^P, \mathcal{V}_i, val_i, \texttt{Pref}_i^P \rangle$ *issu de la définition 15.* PAF *et* $\texttt{VAF}_i^P$ *sont équivalents.*

$val_i^{max}$ correspond à l'injection donnant le sous-ensemble d'arrivée $\mathcal{V}_i$ maximal (en terme d'inclusion ensembliste) appelé $\mathcal{V}_i^{max}$ pour lequel le $\texttt{VAF}_i^P$ tient dans la proposition 7 et $\texttt{VAF}_b^P$ tient dans la proposition 4 pour un même PAF.

**Propriété 8** $|\mathcal{V}_i^{max}| = 2 \times |Pref_b|$

### 3.3 Comparaison entre PAF et CPAF

Un CPAF peut être vu comme une aggrégation de plusieurs PAFs ordonnés, en utilisant un opérateur d'aggrégation dénoté $\oplus$ et défini comme suit :

**Définition 16** *Soit* $\{\texttt{PAF}_1 = \langle \mathsf{Arg}, \mathcal{R}, \succeq_1 \rangle$, ..., $\texttt{PAF}_n = \langle \mathsf{Arg}, \mathcal{R}, \succeq_n \rangle\}$ *un ensemble de cadres de travail en argumentation basés sur les préférences, totalement préordonné par la relation dénotée* $r_{PAF}$. $\oplus$ $(\texttt{PAF}_1, ..., \texttt{PAF}_n)$ *est un* $\texttt{CPAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{C}, \triangleright, \succeq_{c_1}, \cdots \succeq_{c_n} \rangle$ *tel que :*
- $\mathcal{C}$ *est un ensemble de* $n$ *contextes, chaque* $c_i$ *est associé avec* $\texttt{PAF}_i$ *;*
- $\succ$ *est un préordre total sur* $\mathcal{C} \times \mathcal{C}$*, tel que* $(c_i, c_j) \in \succ$ *ssi* $(\texttt{PAF}_i, \texttt{PAF}_j) \in r_{PAF}$ *;*
- $\succeq_{c_i} = \succeq_i$.

Il est clair qu'au regard de cette définition, PAF peut être vu comme un cas particulier de CPAF avec $n = 1$. D'autre part, l'évaluation du CPAF est liée à une fonction d'aggrégation (voir la définition 7), laquelle induit une unique relation de défaite, qui mène à la computation d'un seul PAF. Un PAF peut aussi être obtenu par une aggrégation des relations de préférences contextuelles à partir d'un CPAF.

**Définition 17** *Soit un* $\texttt{CPAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{C}, \triangleright, \succeq_1 , \cdots \succeq_n \rangle$*, un cadre de travail en argumentation basé sur des préférences aggrégées peut être défini comme suit :* $\texttt{PAF}_{ag} = \langle \mathsf{Arg}, \mathcal{R}, \otimes(\succeq_1 , \cdots \succeq_n) \rangle$.

A la suite de cette étude comparative, il ressort quelques aspects intéréssants, s'il paraît clair maintenant que plusieurs typologies de VAF peuvent représenter un seul PAF, un VAF lui ne peut être représenté qu'au travers d'un seul PAF [8]. Le PAF peut également être appréhendé comme un cas particulier du CPAF en terme d'équivalence, lequel ne permet cependant pas une expressivité aussi importante que certains modèles de valeurs en argumentation.

## 4 Vers un modèle unificateur des cadres de travail en argumentation intégrant des arguments et des préférences en contextes

La fin de ce papier est destinée à proposer un cadre permettant de conserver l'aspect générique des systèmes à base de préférences directes entre arguments tout en accentuant une certaine forme d'expressivité.

### 4.1 Un argument peut s'exprimer dans un ou plusieurs contextes

En permettant à un argument de pouvoir s'exprimer dans un ou plusieurs contextes, valeurs ou critères, les travaux précédemment énoncés pourront être intégrés dans un cadre de travail. Ce cadre commun et expressif pourra être utilisé notamment en situation de décision multicritère lors de futurs travaux.

**Définition 18** *Un* cadre de travail en argumentation multi-contextuels basé sur les préférences *(*MCPAF*) peut être représenté par un*

*tuple* $\langle \mathsf{Arg}_1, \ldots, \mathsf{Arg}_n, \mathcal{R}_1, \ldots, \mathcal{R}_n, c_1, \ldots, c_n, \succeq, \mathtt{Pref}_1, \ldots, \mathtt{Pref}_n \rangle$ *où :*

- $\mathcal{C} = c_1, \ldots, c_n$ *est un ensemble des contextes,*
- $\mathsf{Arg}_1, \ldots, \mathsf{Arg}_n$ *sont des ensembles d'arguments,* $\mathsf{Arg}_i \subseteq \mathsf{Arg}$ *(avec* $\mathsf{Arg} = \cup_{i \in [1,n]} \mathsf{Arg}_i$*) est l'ensemble des arguments s'exprimant dans un contexte* $c_i$*,*
- $\mathcal{R}_1, \ldots, \mathcal{R}_n$ *sont des relations binaires représentant une attaque,* $\mathcal{R}_i \subseteq \mathsf{Arg}_i \times \mathsf{Arg}$ *concerne l'attaque d'un argument s'exprimant dans un contexte* $c_i$ *sur un autre argument,*
- $\succeq$ *est un préordre complet sur* $\mathcal{C} \times \mathcal{C}$*,*
- $\mathtt{Pref}_1, \ldots, \mathtt{Pref}_n$ *est un ensemble de préférences,* $\mathtt{Pref}_i \subseteq \mathsf{Arg}_i \times \mathsf{Arg}$ *est un préordre et concerne les préférences d'un argument s'exprimant dans un contexte* $c_i$ *sur un autre argument.*

Un MCPAF est une paire $\langle \mathsf{Arg}, \mathtt{Def} \rangle$, où Def est définie comme suit : $\forall a, b \in \mathsf{Arg}$, $(a, b) \in \mathtt{Def}$ ssi $(a, b) \in \mathcal{R}$ (avec $\mathcal{R} = \cup_{i \in [1,n]} \mathcal{R}_i$ et $(b, a) \notin \otimes(\mathtt{Pref}_1, \ldots \mathtt{Pref}_n)$ où $\otimes$ est un aggrégateur de préférence.

Ce système permet de retranscrire qu'un argument peut s'exprimer dans un ou plusieurs contextes, de comparer deux arguments dans le même contexte ou un argument s'exprimant dans un contexte avec un argument s'exprimant dans un contexte différent (idem pour les attaques).

### 4.2 CPAF est un cas particulier de MCPAF

Le MCPAF est un système d'argumentation qui peut être entrevu comme un cas particulier du CPAF, dans la définition suivante un $\mathtt{MCPAF}^C$ est un MCPAF construit à partir d'un CPAF.

**Définition 19** *Soit un* CPAF *tel que* $\mathtt{CPAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{C}, \succ, \mathtt{Pref}_1, \ldots \mathtt{Pref}_n \rangle$*, un* $\mathtt{MCPAF}^C = \langle \mathcal{A}_1^C, \ldots, A_n^C, \mathcal{R}_1^C, \ldots, \mathcal{R}_n^C, \mathcal{C}^C, \succeq^C, \mathtt{Pref}_1^C, \ldots, \mathtt{Pref}_n^C \rangle$ *peut être défini tel que :*

$\mathcal{A}_1^C = \cdots = A_n^C = \mathcal{A}$
$\mathcal{R}_1^C = \cdots = R_n^C = \mathcal{R}$
$\mathcal{C}^C = \mathcal{C}$
$\succeq^C = \succ$
$\mathtt{Pref}_i^C = \mathtt{Pref}_i$

Ainsi, dans la proposition suivante, $\mathtt{MCPAF}^C$ et CPAF ont été énoncés comme équivalents.

**Proposition 7** *Soit un* CPAF *tel que* $\mathtt{CPAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{C}, \succ, \mathtt{Pref}_1, \ldots \mathtt{Pref}_n \rangle$ *et un* $\mathtt{MCPAF}^C = \langle \mathcal{A}_1^C, \ldots, A_n^C, \mathcal{R}_1^C, \ldots, \mathcal{R}_n^C, \mathcal{C}^C, \succeq^C, \mathtt{Pref}_1^C, \ldots, \mathtt{Pref}_n^C \rangle$ *issu de la définition 19,* $\mathtt{MCPAF}^C$ *et* CPAF *sont équivalents.*

### 4.3 Stratification contextuelle de l'ensemble des entités du MCPAF

L'ensemble des contextes du MCPAF étant muni d'un préordre complet, il existe des contextes indifférents. Il est rappelé ici brièvement et/ou de façon cartésienne, la relation d'indifférence dénotée Ind se définissant comme suit :

**Définition 20** *Dans un* MCPAF*, où* $\succeq$ *est un préordre complet sur* $\mathcal{C} \times \mathcal{C}$*,* $\forall\, c_i, c_j \in \mathcal{C}$*,* $(c_i, c_j) \in \mathtt{Ind}$ *ssi* $(c_i, c_j) \in \succeq$ *et* $(c_j, c_i) \in \succeq$*.*

Ainsi, il est possible de procéder à une stratification des contextes, faisant apparaître un nouvel ordonnancement strict et complet se basant sur les strates.

**Définition 21** $\mathcal{C}^i \subseteq \mathcal{C}$ *est un ensemble de contextes indifférents définis sur une strate i :*

$T^0 = \mathcal{C}$
$\mathcal{C}^1 = T^0 \setminus \{c_i$ *such that* $\exists\, c_j \in T^0 \setminus c_i$*,* $(c_i, c_j) \in \succeq$ *et* $(c_j, c_i) \notin \mathtt{Ind}\}$
$T^{i-1} = \mathcal{C} \setminus \{ \bigcup_{k=1}^{k=i-1} \mathcal{C}^k \}$
$\mathcal{C}^i = T^{i-1} \setminus \{ c_i$ *such that* $\exists\, c_j \in T^{i-1} \setminus c_i$*,* $(c_i, c_j) \in \succeq$ *et* $(c_j, c_i) \notin \mathtt{Ind}\}$

A partir de cette définition, les éléments suivants peuvent être définis, il s'agit de construire des ensembles d'arguments correspondant à ces strates ainsi que des relations de préférences sur ces arguments.

**Définition 22** $\mathsf{Arg}^i \subseteq \mathsf{Arg}$ *est l'ensemble des arguments s'exprimant dans un ensemble de contextes définis sur la strate i :*
$\mathsf{Arg}^i = \cup_{k \in [1,n]} \mathsf{Arg}_k$ *avec* $c_k \in \mathcal{C}^i$*.*

$\mathsf{Arg}^i$ est appelé ensemble des arguments qui s'expriment sur la strate i.

**Définition 23** $\mathtt{Pref}^i \subseteq \mathsf{Arg}^i \times \mathsf{Arg}$ *est l'ensemble des préférences entre arguments s'exprimant sur la strate i et d'autres arguments :*
$\mathtt{Pref}^i = \bigcup \mathtt{Pref}_k$ *tel que k ∈ [1,n] et* $c_k \in \mathcal{C}^i$*.*

$\texttt{Pref}^i$ est appelé strate i de préférences. Ci-dessous, il va être défini des ensembles de préférences en contradiction, et ce sur une même strate de préférence, un tel ensemble est dénoté $\texttt{PC}^k$ lorsqu'il est défini sur une strate k de préférence.

**Définition 24** $\texttt{PC}^k \subseteq \mathsf{Arg} \times \mathsf{Arg}$ *est défini comme suit :* $\texttt{PC}^k = \{$*(a, b) tel que (a, b)* $\in \texttt{Pref}_i$, *(b, a)* $\in \texttt{Pref}_j$ *et* $(c_i, c_j) \in \texttt{Ind}\}$

Il est donc possible de construire un aggrégateur de préférences pour le MCPAF lequel ommettra de rajouter les préférences en contradiction sur une même strate de préférences.

**Définition 25** *Un aggrégateur de préférence pour le* MCPAF *peut être défini comme suit* $\otimes(\texttt{Pref}^1, \ldots \texttt{Pref}^n) = \Pi^n$ *:*

$\Pi^1 = \{$*(a, b)* $\in \texttt{Pref}^1$ *et (a, b)* $\notin \texttt{PC}^1$ $\}$

$\Pi^{k+1} = \Pi^k \cup \{$*(a, b)* $\in \texttt{Pref}^{k+1}$ *et (b, a)* $\notin \texttt{PC}^{k+1} \cup \Pi^k\}$.

### 4.4 Exclusion des préférences contradictoires de même strate pour l'équivalence d'un CPAF à partir d'un MCPAF

Le définition suivante fait état du fait qu'un CPAF peut être extrait d'un cadre de travail en argumentation multi-contextuel basé sur les préférences et appelé MCPAF.

**Définition 26** *Soit un* MCPAF $= \langle \mathsf{Arg}_1, \ldots, \mathsf{Arg}_n, \mathcal{R}_1, \ldots, \mathcal{R}_n, \mathcal{C}, \succeq, \texttt{Pref}_1, \ldots, \texttt{Pref}_n \rangle$*, et un* $\texttt{CPAF}^M$ *défini à partir d'un* MCPAF *comme un tuple* $\langle \mathsf{Arg}^M, \mathcal{R}^M, \mathcal{C}^M, \succ^M, \texttt{Pref}_1^M, \ldots, \texttt{Pref}_m^M \rangle$ *décrit tel que :*

$\mathsf{Arg}^M = \bigcup \mathsf{Arg}_i$,

$\mathcal{R}^M = \bigcup \mathcal{R}_i$,

$\mathcal{C}^M$ *est un ensemble d'ensemble de contextes tel que* $\mathcal{C}^M = \{\mathcal{C}^1, \ldots, \mathcal{C}^m\}$*, avec* $\mathcal{C}^i$ *l'ensemble des contextes définis sur la strate i à partir de l'ensemble* $\mathcal{C}$,

$\succ^M$ *est défini tel que si* $i < j$ *alors* $(\mathcal{C}^i, \mathcal{C}^j) \in \succ^M$,

$\texttt{Pref}_i^M = \texttt{Pref}^i \setminus \texttt{PC}^i$*, avec* $\texttt{Pref}^i$ *la strate i de préférences issue de l'ensemble* $\{\texttt{Pref}_1, \ldots, \texttt{Pref}_n\}$ *et* $\texttt{PC}^i$ *son sous ensemble de préférences contradictoires.*

Enfin, dans la proposition suivante, MCPAF et $\texttt{CPAF}^M$ sont énoncés comme équivalents.

**Proposition 8** *Soit un* MCPAF $= \langle \mathsf{Arg}_1, \ldots, \mathsf{Arg}_n, \mathcal{R}_1, \ldots, \mathcal{R}_n, \mathcal{C}, \succeq, \texttt{Pref}_1, \ldots, \texttt{Pref}_n \rangle$ *et un* $\texttt{CPAF}^M = \langle \mathsf{Arg}^M, \mathcal{R}^M, \mathcal{C}^M, \succ^M, \texttt{Pref}_1^M, \ldots, \texttt{Pref}_m^M \rangle$ *issu de la définition 26.* $\texttt{CPAF}^M$ *et* MCPAF *sont équivalents.*

Il s'agit maintenant d'ajouter une indifférenciation entre deux contextes dans le contexte de l'exemple 1.

**Exemple 7 (Example 1. cont.)** *Dans le cadre d'un* MCPAF*, on propose les contextes suivants* $c_1$ *= Utilité,* $c_2$ *= Pertinence,* $c_3$ *= Originalité et* $c_4$ *= Inadéquation, le contexte* $c_4$ *privilégiant un inversement des préférences obtenues dans le critère* $c_2$ *(*$(R_2, J_2) \in \texttt{Pref}_4$*), avec un ordonnancement représenté comme ceci : Utilité* $\succeq$ *Pertinence* Ind *Inadéquation* $\succeq$ *Originalité. Les différents ensembles d'arguments sont équipés de quatre relations de préférence, respectivement denotées* $\succeq_U, \succeq_P, \succeq_O$ *et* $\succeq_I$*. Ces relations sont définis comme suit :* $\succeq_U = \{(R_1, J_1)\}$, $\succeq_P = \{(J_2, R_2)\}$, $\succeq_O = \{(R_3, J_3)\}$, $\succeq_I = \{(R_2, J_2)\}$*. La relation de préférence aggrégée est dans ce cas l'union des quatre relations avec une omission volontaire des préférences contradictoires* $\succeq_P$ *et* $\succeq_I$*. La représentation associé à ce cadre de travail équivalent au* CPAF *peut être obtenue ci-dessous par un tuple :* $\texttt{CPAF}^M = \langle \{ R_1, J_1, J_2, R_2, R_3, J_3 \}, \{ (R_1, J_1), (R_1, J_2), (R_1, J_3), (J_1, R_2), (J_1, R_3), (R_2, J_2), (R_2, J_3), (J_2, R_2), (J_2, R_3), (J_3, R_3) \}, \{$*Utilité, Présentation/Inadéquation, Orignalité* $\}$*, Utilité* $\succ$ *Pertinence/Inadéquation* $\succ$ *Orignalité,* $\texttt{Pref}_1 = (R_1, J_1)$, $\texttt{Pref}_2 = \emptyset$, $\texttt{Pref}_3 = (R_3, J_3)$ $\rangle$.

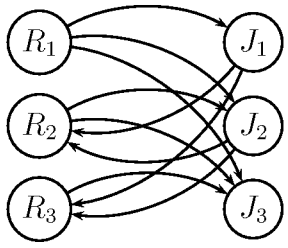
### 4.5 Expressivité du MCPAF utilisable dans les strates aggrégationelles

Il est possible de définir une strate de préférence i étendue dénotée $\texttt{Pref}_{ext}^i$ comme suit :

**Définition 27** $\texttt{Pref}_{ext}^i = \texttt{Pref}^i \cup \{$ *(a, b) tel que* $a \in \mathsf{Arg}^i$ *et* $b \notin \mathsf{Arg}^i\}$

Ainsi, cette nouvelle définition de la strate de préférence permet au MCPAF d'entrevoir de nouveaux résultats en sortie, de par cette nouvelle aggrégation et ses sémantiques d'extensions.

**Exemple 8 (Example 7. cont.)** *:*




*Ce cadre travail possède donc potentiellement une seule extension admissible maximale, qui est* $\{R_1, R_2, R_3\}$*. Il est donc clair que Rémi est préférré à Jean puisque cette extension contient seulement les arguments supportant Rémi.*

## 5 Conclusion

Ce papier soulève un problème et/ou une solution concernant la comparaison de trois extensions des cadres de travail en argumentation de Dung. De manière à mener une comparaison aussi bien syntaxique que sémantique de ces modèles, nous avons définis préalablement une notion de cadre de travail équivalent en argumentation. Ainsi, nous avons considéré deux cadres de travail comme équivalent si ceux-ci retournent exactement les même extensions sous une sémantique donnée (en l'occurence la sémantique d'admissibilité utilisée dans les preuves de cet article).
Il a donc été montré que ces cadres de travail peuvent être considérés comme équivalents et ce sous des conditions particulières. Un exemple trivial peut être le cas où tous les arguments d'un cadre de travail basé sur les préférences sont incomparables ou indifférents, le cadre de travail classique de Dung est ainsi retrouvé. L'étude a également montré qu'un cadre de travail en argumentation basé sur les préférences peut être construit à partir d'un cadre de travail en argumentation basé sur les valeurs et que ceux-ci seront équivalents, puis que plusieurs familles de cadres de travail en argumentation basés sur les valeurs peuvent également être considérées comme équivalentes au cadre de travail en argumentation basé sur les préférences.
Il a également été notifié qu'un cadre de travail en argumentation basé sur les préférences était un cas particulier du cadre de travail en argumentation basé sur les préférences contextuelles, mais qu'il restait tout de même moins expressif dans l'expression des contextes et/ou des valeurs liés à l'argument qu'un cadre de travail basé sur les valeurs par exemple.
Finalement, il a été proposé d'intégrer ces trois extensions de cadres de travail de type Dung dans un cadre commun au moins plus général et plus expressif dont les propriétés seront investiguées et son utilisation envisagée dans le cadre de la décision multi-critère cette fois-ci, lors de futurs travaux.

## Références

[1] L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34 :197–216, 2002.

[2] L. Amgoud, S. Parsons, and L. Perrussel. An argumentation framework based on contextual preferences. In *Proceedings of the International Conference on Formal and Applied and Practical Reasoning (FAPR'00)*, pages 59–67, 2000.

[3] L. Amgoud and H. Prade. Using arguments for making and explaining decisions. *Artif. Intell.*, 173(3-4) :413–436, 2009.

[4] T. J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3) :429–448, 2003.

[5] B. Bonet and H. Geffner. Arguing for decisions : A qualitative model of decision making. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI'96)*, pages 98–105, 1996.

[6] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and $n$-person games. *Artificial Intelligence Journal*, 77 :321–357, 1995.

[7] J. Fox and S. Das. *Safe and Sound. Artificial Intelligence in Hazardous Applications*. AAAI Press, The MIT Press, 2000.

[8] S. Kaci and L. van der Torre. Preference-based argumentation : Arguments supporting multiple values. *Int. J. Approx. Reasoning*, 48(3) :730–751, 2008.

[9] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation : a logical model and implementation. 104 :1–69, 1998.

[10] K. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28 :203–242, 1990.

# Multiagent Learning and Optimality Criteria in Repeated Game Self-play

Andriy Burkov
burkov@damas.ift.ulaval.ca

Brahim Chaib-draa
chaib@damas.ift.ulaval.ca

DAMAS Laboratory
Computer Science Department
Laval University, Canada

**Résumé :**
Nous présentons une approche d'apprentissage multiagent permettant de satisfaire un critère d'optimalité donné lorsque les jeux répétés se font en "self-play". Notre approche est confrontée aux approches classiques d'apprentissage pour les jeux répétés qui visent généralement à apprendre un équilibre (Nash, Pareto). Une comparaison est donnée d'un point de vue pratique (ou d'ingénieur), c'est-à-dire d'un point de vue d'un concepteur d'un système multiagent dont le but est de maximiser la performance totale du système selon un critère d'optimalité donné. De nombreuses experimentations dans une large variété des jeux répétés démontrent l'efficacité de notre approche.

**Mots-clés :** Apprentissage multiagent, jeux répétés, jeux en "self-play"

**Abstract:**
We present a multiagent learning approach to satisfy any given optimality criterion in repeated game self-play. Our approach is opposed to classical learning approaches for repeated games: namely, learning of equilibrium, Pareto-efficient learning, and their variants. The comparison is given from a practical (or engineering) standpoint, i.e., from a point of view of a multiagent system designer whose goal is to maximize the system's overall performance according to a given optimality criterion. Extensive experiments in a wide variety of repeated games demonstrate the efficiency of our approach.

**Keywords:** Multiagent learning, repeated games, self-play

## 1 Introduction

Until now, the main body of the state-of-the-art multiagent learning (MAL) research [2] has been focused on finding a learning rule possessing specific properties. For example, when adopted by all agents of a multiagent system (MAS), such a rule could bring to each agent an accumulated reward which is "optimal" in a certain sense. I.e., according to the classical approach, a learning rule is considered to be good if the rewards accumulated by the agents (also called "players") are close to some values satisfying a certain criterion of optimality[1]. Two most widely used optimality criteria in the context of learning in repeated games are: closeness of the value accumulated by each player to the value of a certain (a) Nash equilibrium or (b) Pareto-efficient joint strategy (which need not be an equilibrium).

The scenario where all agents use the same algorithm is called "self-play". Typically, the performance guarantees of a learning rule are given assuming self-play [4, 2, 7, 1]. First of all, this is because it is generally simpler to analyze the properties of a dynamical process induced by a number of identical learning rules. However, another important reason is that the "self-play" multiagent systems (SPMAS) are of a great practical interest.

Indeed, given an algorithm able to converge to a value (or values) close to a given optimality criterion in some SPMAS, an engineer can create a number of identical agents (in the case of software agents, one can just make as many copies of one agent as required) put these agents into a given (usually unknown) environment and let them converge.

However, in an arbitrary repeated game, the values corresponding to different optimality criteria can vary substantially from one criterion to another. So, when the game being played is unknown, it is usually hard to chose the best learning rule. Another problem, when using algorithms satisfying such optimality criteria as (a) or (b) listed above, is that, from a practical point of view, neither of those criteria can be satisfactory for all SPMAS. Let us clarify this claim.

Let suppose that we are an engineer that receives from a client a problem that needs to be solved by a number of identical agents. (We will call this problem "the environment" and this environment is supposed to be unknown in terms of players' rewards for different actions.)

[1] As a matter of fact, the terms "optimal" and "optimality" are not always appropriate in MAS. Indeed, there are often multiple entities (agents) having different interests in a MAS. In the classical game theoretical literature such terms as "Pareto efficiency" or "equilibrium" are used in place of "optimality". Nevertheless, we will use these terms to unify and simplify the presentation.

The agents (if embodied) are provided by the client, but we are free to decide about the algorithms used by the agents to solve the problem. The client expects the good solution to satisfy a certain quantitative criterion based on the values accumulated by the agents. For example, this criterion can require that the solution maximize a given (i.e., provided by the client) algebraic function of players' accumulated rewards. In this case, which of the existing MAL algorithms satisfying their respective optimality criteria will we choose?

One solution would be to run each MAL algorithm on the given problem, observe the results, and pick the best. However, such an approach can be time and ressource expensive, and does not guarantee optimality. A more consistent approach is to construct a new learning algorithm able to solve problems in SPMAS in a way to satisfy functional criteria.

These functional criteria are opposed to such criteria as (a) and (b) above, which we call "relational", meaning that they are defined by taking into account relations between the values accumulated by each individual agent. In this case, the absolute values themselves *are secondary*. For example, a joint-strategy of multiple players is said to be a Nash equilibrium (criterion a) if the expected reward of *each player* is maximized given that *the other players* have their strategies fixed. In a similar manner, a joint-strategy is said to be Pareto-efficient (criterion b) if by changing this strategy so as to increase the expected value of *any subset of players*, there will necessarily be *a player out of this subset* whose value decreases. The same reasoning is applicable to a number of other relational optimality criteria (such as, for example, correlated equilibrium [6]).

In this paper, we propose an approach to multiagent learning in repeated game self-play scenario when the goal is to satisfy a given functional optimality criterion. We show that in such a setting our new learning algorithm, called *Self-play Learner*, is a better choice than a whole family of equilibrium and Pareto-efficient strategy learning algorithms.

# 2 Formal notions

We focus our attention on repeated games as a model to represent a MAS. For simplicity of exposition, the most of our presentation will be given for two-player case. Extensions to an $n$-player setting (for an arbitrary $n > 2$) as well as to the multistate problems are discussed in Section 6.

## 2.1 Matrix games and their solutions

A finite repeated two-player matrix game $\Gamma$ (henceforth, a repeated game) consists of a set $P$ of two players, $p$ and $q$, with $(p, q) \in \{(1, 2), (2, 1)\}$, and a set $R$ of two two-dimensional matrices, $R = \{R^p, R^q\}$. Player $p$ has a finite number $M^p \in \mathbb{N}^+$ of actions it can choose from. The game is played iteratively. At iteration $i = 1, 2, \ldots$, each player $p$ chooses an action $a_i^p \leq M^p$ and the vector $\mathbf{a}_i = (a_i^p, a_i^q) \in A$ gives a *joint action*. $A$ is called the joint action space of players. For each player $p$ and to each joint action $\mathbf{a}_i \in A$ there corresponds a real valued number in matrix $R^p$ defining the reward of that player after playing joint action $\mathbf{a}_i$.

To choose an action from $M^p$ at any iteration, each player uses a certain rule. This rule is called player's *strategy*. A player's strategy can be *stationary* or *non-stationary*. Let $\pi_i^p$ denote the rule by which player $p$ chooses its action at iteration $i$. Then, $p$'s strategy $\pi^p$ is called stationary if $\pi_i^p = \pi_0^p$, $\forall i$. This means that $p$'s strategy does not depend on current iteration, or, in other words, that it cannot change with time. Otherwise the strategy is called non-stationary.

A strategy profile $\boldsymbol{\pi}$ is a joint strategy of players, $\boldsymbol{\pi} = (\pi^p, \pi^q)$. To compare strategies and strategy profiles between them (i.e., to say whether one is better than another) it is required to assign a metric to a strategy. We are using the expected limit of the means (ELM) metric. ELM assigns a unique value to an expected sequence of rewards that are obtained by a player when both players follow a given strategy profile $\boldsymbol{\pi}$ during an infinite number of iterations:

$$u^p(\boldsymbol{\pi}) = E_{\boldsymbol{\pi}}\left[\lim_{T\to\infty}\frac{1}{T}\sum_{i=1}^{T} R^p(\boldsymbol{\pi}_i)\right] \quad (1)$$

In the above equation, $u^p(\boldsymbol{\pi})$ is the ELM value of strategy $\boldsymbol{\pi}$. $R^p(\boldsymbol{\pi}_i)$ denotes the expected immediate reward obtained by player $p$ at iteration $i$ if both players follow the strategy $\boldsymbol{\pi}$ at that iteration.

Nash equilibrium is a strategy profile $\hat{\boldsymbol{\pi}} =$

$(\hat{\pi}^p, \hat{\pi}^q)$ such that the following condition holds:

$$u^p(\pi^p, \hat{\pi}^q) \leq u^p(\boldsymbol{\pi}) \text{ and } u^q(\hat{\pi}^p, \pi^q) \leq u^q(\boldsymbol{\pi}), \quad \forall \pi^q \neq \hat{\pi}^q, \pi^p \neq \hat{\pi}^p \tag{2}$$

Let $M(\Gamma)$ denote the set of all strategy profiles of the game $\Gamma$. A Pareto-efficient solution of $\Gamma$ is a strategy profile $\bar{\boldsymbol{\pi}} \in M(\Gamma)$ such that $\forall \boldsymbol{\pi}' \neq \bar{\boldsymbol{\pi}}$, the following condition holds:

$$u^p(\boldsymbol{\pi}') < u^p(\bar{\boldsymbol{\pi}}) \text{ or } u^q(\boldsymbol{\pi}') < u^q(\bar{\boldsymbol{\pi}}) \tag{3}$$

### 2.2 Optimality criteria

The equations (2–3) define two relational optimality criteria discussed in the previous section. And as we claimed above, there are tasks where a use of relational criteria is not justified from a practical standpoint. In such environments, we would prefer agents to learn (and to use thereafter) the strategies maximizing a certain mathematical function of their utility. This, functional, optimality criterion, depending on the task, can be based on such functions as *max*, *sum*, *product* or any other desirable function of players' individual utilities. If the utility is defined using the ELM metric then the functional optimality criterion $u(\boldsymbol{\pi})$ for a strategy profile $\boldsymbol{\pi}$ can be defined as $u(\boldsymbol{\pi}) = \text{Op}_p(u^p(\boldsymbol{\pi}))$, where $u^p(\boldsymbol{\pi})$ is the utility of player $p$ defined using equation (1). In the latter equation, Op denotes a certain mathematical operator. For a given problem, it needs to be replaced by $\max$, $\sum$, $\times$ or any required function.

We should mention here that the we are not the first to focus on functional optimality criteria. This principle was used by Nash [9] in his axiomatic analysis of bargaining, where he proposed to choose the solution point maximizing the product of individual values of players. From the computer science perspective, Littman and Stone [8] adapted Nash's idea to choose the best solution point in their algorithm computing Nash equilibrium in repeated games. For our part, we extend these ideas to the *learning* in self-play. Another related work is discussed in Section 7.

### 2.3 Self-play

We *explicitly* focus on the self-play setting; and this is a *controlled* self-play, not an accidental coincidence of learning algorithms of agents. This is what differs our approach from the main body of modern multiagent learning research proposing algorithms whose *behavior is justified for* (or *examined in*) the self-play scenario. Recall that by definition, self-play is a MAS setting in which *all* agents are *identical*. Until now, it has been typically assumed that agents' *algorithms*, or, in other words, rules of strategy update when learning, are identical. Other properties that can also be identical, such as (i) initial knowledge of agents, (ii) their utility metrics and (iii) optimality criteria, have escaped the attention of researchers. In this paper, we aim to fill this gap.

More precisely, in our controlled self-play scenario, which we call SPMAS, we assume that both players, (1) *use the same learning algorithm*, (2) *have internal variables initialized with the values known to both players*, (3) *use the same utility metric* and (4) *optimize the same functional criterion*. We claim that in any *controlled* self-play scenario, Assumptions 2–4 are as well natural as Assumption 1, which is made in many previous multiagent learning papers [4, 2, 7, 6, 1]. In particular, this means that in any SPMAS,

Asmpt. (1) satisfied $\Longleftrightarrow$ Asmpts. (2–4) can be satisfied.

Also, we assume that players can observe each other's actions and their own rewards after a joint action is executed. This is as well a common assumption for many of MAL algorithms [4, 7, 1, 5].

### 2.4 Information and communication

Two important questions characterizing any MAS are (i) whether the agents know their own reward function and the reward function of the other agent, and (ii) whether communication between agents is available during learning. In this paper, we assume that the answer to both questions is *No*. Indeed, an affirmative answer to the first question makes the learning unnecessary, since the agents can compute an optimal joint strategy using the reward matrices. On the other hand, if there is a communication between agents, the simplest scenario is to explore the reward structure of the game by executing joint actions one by one. Then, using communication, agents are able to share the acquired data. This, again, will make a further learning unnecessary.

## 3 Extended strategies

To present our new algorithm, we first need to discuss one important implication of using the product criterion: emerging of strategies extended in time, or simply "extended strategies". These non-stationary strategies are known to be able to maximize the product of players' individual utilities to a greater extent than any stationary strategy [8]. As we will demonstrate later, our Self-play Learner algorithm is able to learn extended strategies.

When two players $p$ and $q$ play a joint action $\mathbf{a} = (a^p, a^q)$ their rewards can be visualized as a point $\mathbf{x} = (x^p, x^q) = (R^p(a^p, a^q), R^q(a^p, a^q))$ in a two-dimensional space.[2] Let the set $X$ contain all such points: $X = \{(R^p(a^p, a^q), R^q(a^p, a^q)) : a^p \leq M^p, a^q \leq M^q)\}$. Players can achieve any point in $X$ as their ELM values by playing the corresponding joint action at every iteration. The convex hull of the set $X$ contains all points that can be obtained as a linear combination of a subset of points of $X$. It is easily observable that the points laying on the boundary of the convex hull are always constructed as a linear combination of only two points of $X$. In terms of players' strategies, a point $\mathbf{z}$ on the boundary (recall that a point in $X$ is a vector of players' ELM values) can be achieved by the players by playing a joint action, corresponding to a certain point $\mathbf{x}$, a $w$-fraction of all iterations, and by playing another joint action, corresponding to a point $\mathbf{y}$, the $(1-w)$-fraction of all iterations (where $w$, $0 \leq w \leq 1$, defines the coefficient of linear combination).

**Definition 1.** *Given $l \in \mathbb{N}^+$, $0 \leq w \leq 1$, $\mathbf{a} \in A$ and $\mathbf{b} \in A$, an Extended Joint Action (EJA) is a joint strategy in which players play $\mathbf{a}$ during the first $k = \lceil l \cdot w \rceil$ iterations and $\mathbf{b}$ during the following $l - k$ iterations.*

**Definition 2.** *An Extended Joint Strategy (EJS) is an EJA repeated infinitely often.*

We call $l$ the *length* of an EJA and $k$ its *switch point*. Notice that for each point $\mathbf{z}$ constructed as a combination of two points $\mathbf{x}$ and $\mathbf{y}$ from $X$, $\exists$ an EJS with certain $\mathbf{a}$, $\mathbf{b}$, $l$ and $w$.

When the product criterion is used, the boundary of the convex hull is of a particular interest because the point $\mathbf{z}$ maximizing the product criterion is always found on the boundary [9]. For two given points of $X$, $\mathbf{x} = (x^p, x^q)$ and $\mathbf{y} = (y^p, y^q)$, forming an edge of the boundary, the value of $w$ maximizing the product criterion on this edge can be computed as follows [8],

$$w = \frac{-y^q(x^p - y^p) - y^p(x^q - y^q)}{2(x^q - y^q)(x^p - y^p)} \qquad (4)$$

If $w < 0$ or $w > 1$, the maximum is achieved at respectively $\mathbf{x}$ or $\mathbf{y}$. To find $w^*$ maximizing the product criterion over all points of the convex hull, it is only required go over all pairs of points of $X$, compute $w$ using Equation (4) and then pick a pair $\mathbf{x}^*$ and $\mathbf{y}^*$ of points (and the corresponding $w^*$) for which the criterion is maximized.

As one can note, an EJS will achieve the optimal ELM value defined by $w^*$, $\mathbf{x}^*$ and $\mathbf{y}^*$ only when $l \to \infty$. Let us show that as $l \to \infty$ the error induced by using a finite value of $l$ rapidly decreases.

**Proposition 1.** *Let $R$ denote the ELM value of an optimal point $\mathbf{z}$ on the boundary of $X$ defined by the values $w^*$, $\mathbf{x}^*$ and $\mathbf{y}^*$ found as described above. Let $l$ be the length of an EJA defined for two joint actions $\mathbf{a}$ and $\mathbf{b}$ from $A$ corresponding to the points $\mathbf{x}^*$ and $\mathbf{y}^*$ from $X$. Let $\tilde{R}$ denote the ELM of this EJA. Let $\epsilon = \tilde{R} - R$ define the error of using $l < \infty$ in this EJA. Then $\epsilon \to 0$ as $l \to \infty$.*

*Proof.* We have $l^2 R = (lwx^p + (l - lw)y^p)(lwx^q + (l - lw)y^q)$ and $l^2\tilde{R} = (\lceil lw \rceil x^p + (l - \lceil lw \rceil)y^p)(\lceil lw \rceil x^q + (l - \lceil lw \rceil)y^q)$. We know that for any natural $x$, $\lceil x \rceil < x + 1$. Thus, we can write that $l^2\tilde{R} < ((lw + 1)x^p + (l - (lw + 1))y^p)((lw + 1)x^q + (l - (lw + 1))y^q)$. The difference between $l^2\tilde{R}$ and $l^2 R$ is then bounded as follows: $l^2\tilde{R} - l^2 R < l(x^p - y^p)(x^q - y^q)(2w - 1) + 2x^p x^q - x^p y^q - y^p x^q$. Since $l > 1$, the error $\epsilon = \tilde{R} - R$ is bounded as follows: $\epsilon < \frac{(x^p - y^p)(x^q - y^q)(2w-1)}{l} + \frac{2x^p x^q - x^p y^q - y^p x^q}{l^2}$. Therefore, as $l$ tends to $\infty$, $\epsilon$ tends to 0 with a rate inversely proportional to $l$. □

When $l$ and $w^*$ are known to the players (i.e., defined by the designer of the MAS before to start learning) they are able to construct the EJS maximizing, to the extent of the error induced by using a finite value of $l$, the product criterion.

[2] In this subsection, we use a simplified notation introduced in [8]. According to it, $\mathbf{x} = (x^p, x^q)$ and $\mathbf{y} = (y^p, y^q)$ denote the vectors of players' rewards for two different joint actions viewed as two points in a two-dimensional space.

# 4 Self-play Learner

In this section, we present our new algorithm called Self-play Learner (SPL).

## 4.1 Internal variables

Our algorithm has one internal variable that needs to be initialized with the same value for all agents before the learning is started. This variable, called $R_{max}$, reflects the maximum utility that an agent can obtain in the game. In many practical tasks, this value can be set by the MAS designer depending on how the utility is defined. For example, for robots cleaning the floor this value can be set based on the maximum possible surface one robot can clean given the initial volume of detergent in its tank. It is assumed that $R_{max} \geq R^p(a^p, a^q)$ for any player $p$ and for all $a^p \leq M^p$ and $a^q \leq M^q$. I.e., $R_{max}$ does not underestimate any of the rewards of players.

During learning, an SPL agent $p$ maintains several other internal variables. The variables $K^p(a^p, a^q)$, $\forall a^p \leq M^p, a^q \leq M^q$, reflect the number of times that a particular joint action $(a^p, a^q)$ has been played. The variables $L^p(a^p, a^q)$, $\forall a^p \leq M^p, a^q \leq M^q$, reflect the number of times that the action $a^q$ has been played by $q$ at the iteration $i+1$ following an iteration $i$ at which players were playing $(a^p, a^q)$.

## 4.2 Our algorithm

The main steps of our SPL algorithm are:

1. While learning (exploration phase)
   (a) Play a random action,
   (b) Observe the reward $R$,
   (c) Replay the same action proportionally to $R$,
   (d) Update counters of the other player's play.
2. While playing (exploitation phase)
   (a) Optimize according to the criterion and the counters,
   (b) Play optimally.

**Algorithm 1:** Main steps of Self-play Learner.

**Exploration phase.** During the finite exploration phase (whose length is known by both agents) players explore the reward structure of the game. SPL players are explicitly synchronized (this is an advantage of self-play). During the exploration phase, at each odd iteration $i$, an SPL player $p$ plays a random action $a_i^p$ and observes its reward $R_i^p$ and the action $a_i^q$ played by the other player. On the next iteration, $i+1$, player $p$ replays the action $a_i^p$ played at the previous iteration with probability $\delta = R_i^p / R_{max}$. Otherwise, with probability $(1-\delta)$ player $p$ plays a random action (different from $a_i^p$) from $M^p$.

**Exploitation phase.** As soon as, during the exploration phase, both players were using the same algorithm, during the exploitation phase player $p$ can make the following assumption about the unknown reward function of player $q$:

$$\frac{R^q(a^p, a^q)}{R_{max}} \approx \frac{L^p(a^p, a^q)}{K^p(a^p, a^q)}$$

Indeed, since the value $R_{max}$ is the same for (and is known by) both players; and as the players replayed at every iteration $i+1$ their action $a_i^p$, played at the previous iteration $i$, according to the proportion $R^p(a_i^p, a_i^q)/R_{max}$, the values of counters $L^p(a^p, a^q)$ and $K^p(a^p, a^q)$ can give to player $p$ a good estimate of the real value of $R^q(a^p, a^q)$. More precisely, player $p$ can compute an estimate of the other player's rewards as follows,

$$\tilde{R}^q(a^p, a^q) = \frac{L^p(a^p, a^q) R_{max}}{K^p(a^p, a^q)} \qquad (5)$$

By so doing, it becomes possible to compute the strategy maximizing any given functional criterion and to execute this strategy thereafter. For example, if the functional criterion is $sum$, the optimal strategy can be computed by the players as,

$$\pi_i^p = \underset{a^p:(a^p,a^q)\in A \wedge a^q \leq M^q}{\operatorname{argmax}} \left( \tilde{R}^q(a^p, a^q) + R^p(a^p, a^q) \right) \forall i \qquad (6)$$

If the functional criterion is $max$, the optimal strategy can be computed similarly. Finally, if the functional criterion is $product$, the optimal strategy can be computed as shown in Algorithm 2.

In many games, to construct the optimal strategy to execute during the exploitation phase, both players can use Equation (6) or the procedure of Algorithm 2. The only problem arises when there are several points in $X$ whose ELM values are close to each other according to the functional criterion in question. Let suppose we have

1. For all pairs of points $\mathbf{x}$, $\mathbf{y}$ from the set $X$, such that, $\mathbf{x} = (R^p(a^p, a^q), \tilde{R}^q(a^p, a^q))$ and $\mathbf{y} = (R^p(b^p, b^q), \tilde{R}^q(b^p, b^q))$ (where $a^p, b^p \leq M^p$ and $a^q, b^q \leq M^q$) compute $w$ using Equation (4) and construct the corresponding EJS using Definitions 1–2.
2. Pick the EJS having the highest ELM value (according to the $product$ functional optimality criterion).

**Algorithm 2:** Procedure to find the optimal strategy for the $product$ criterion.

a game as follows,

$$R^{1,2} = \begin{pmatrix} 1,2 & 0,0 \\ 0,0 & 2,1 \end{pmatrix}$$

At the end of the exploration phase, players have certain estimates of the other player's reward function computed using Equation (5). Two points, $(1,2)$ and $(2,1)$, in $X$ have the same ELM value. However, as the length of the exploration phase is finite, the players have an error in estimates of each other's rewards. Therefore, the strategies computed by the players independently can belong to different joint strategies. For example, player 1 can decide that the optimal strategy is to play the row 1, expecting to see the outcome $(1,2)$, but player 2 will play the column 2 foreseeing the outcome $(2,1)$. As the result, they will collect the suboptimal outcome $(0,0)$.

To settle this coordination problem, SPL players have different roles determined by their player numbers. Roles, as well as player numbers, is a shared information which is given to both players before the learning is started. Each player knows its role and behaves accordingly. Let us call the roles *leader* and *follower*[3][4]. The leader computes its strategy using Equation (6) or the procedure of Algorithm 2. The follower, in order to be synchronized with the leader, computes its strategy as a function of the leader's strategy. In particular, if the functional criterion is $sum$, the follower computes its strategy as,

$$\pi^p_{i+1} = \underset{a^p:(a^p,a^q_i)\in A}{\operatorname{argmax}} \left(\tilde{R}^q(a^p, a^q_i) + R^p(a^p, a^q_i)\right) \ \forall i \tag{7}$$

If the functional criterion is $max$, the follower computes its strategy in a similar way. Finally, if the functional criterion is $product$, the follower keeps in memory the most recent EJA played so far. Let $i+1$ denote the iteration corresponding to the beginning of a new period of length $l$. At the beginning of iteration $i+1$, the follower computes the new EJA containing its own sequence of $l$ optimal actions assuming that the leader will follow the strategy played in the previous period. Then, at iterations $i+1, i+2, \ldots, i+l$, the follower executes this sequence. More formally, let $H$ denote the leader's part of the most recent EJA played so far and let $H_k$, $1 \leq k \leq l$, be the leader's actions in this EJA. To find its optimal sequence of actions, the follower $p$ first finds the switch point $k^*$ as the smallest $k$ such that $H_{k-1} \neq H_k$. Then it sets $a^{q*} = H_{k-1}$ and $b^{q*} = H_k$ and finds the pair $(a^{p*}, b^{p*})$ that satisfies,

$$(a^{p*}, b^{p*}) = \underset{(a^p,b^p):a^p,b^p\leq M^p}{\operatorname{argmax}} \left(\tilde{R}^q(a^p, a^{q*}) \cdot R^p(b^p, b^{q*})\right) \tag{8}$$

Then the new EJA is defined by $k^*$ and the joint actions $\mathbf{a}^* = (a^{p*}, a^{q*})$ and $\mathbf{b}^* = (b^{p*}, b^{q*})$.

## 5 Experimental results

It is only fair to compare a new algorithm with the existing algorithms if it uses the same or relaxed assumptions and is searching for the same kind of solution. In our case, there is no other algorithm capable of learning strategies optimizing functional criteria in MAS (two exceptions and their limitations are discussed in Section 7). On the other hand, there exist a number of MAL algorithms, as those cited above, which, while using different assumptions, converge to the same kinds of relational solutions like Pareto-efficient or Nash equilibrium. So, in our case we will indirectly compare our algorithm with all these algorithms by comparing the ELM value of the solution found by SPL with the corresponding values (according to the same criterion) of different relational solutions. The goal of this comparison is to demonstrate that when the goal of the designer is to satisfy a given functional optimality criterion, SPL is the best choice.

We empirically tested SPL on two different testbeds. The first series of testbeds, called "Random Games M" (or, RGs M, for short), contains randomly generated two-player repeated games with the number of player actions, $M = M^p = M^q$, equal respectively to 2, 3, 5 and 10. In each game from Random Games M, the rewards of players are integer values uni-

[3]Despite such names, the agents are still making their actions simultaneously. Thus we are remaining in a repeated game formalism.

[4]Roles can be assigned at random: the ELM value of a joint strategy does not depend on a particular choice of leader and follower.

formly distributed between $0$ and $100$, and new values are generated each time a game is started.

The second testbed, called "Conflict Games" (CGs), contains $57$ games listed in [3]. These are two-player two-action repeated games whose rewards are integer values between $1$ and $4$. These games were called "conflict" because there exists no outcome that simultaneously maximizes the ELM value of both players in these games. Conflict Games are especially suitable to make a comparison of solutions computed by SPL for different functional criteria with other possible solutions usually found by other MAL algorithms in self-play (e.g., Nash equilibrium and Pareto-efficient solution).

We did our experiments in the following way. From each testbed, a game was randomly picked and played during $100,000$ iterations. This process (called an *experiment*) was repeated $100$ times and then the obtained data were averaged. Table 1 presents the ELM values of the strategies to which SPL players converge in different games. The alternative (relational) solutions are respectively the best (in terms of the corresponding ELM value) pure stationary Pareto-efficient solution (BPE column) and the best and the worst stationary Nash equilibria (BNE and WNE columns respectively). We did not compare the SPL's solution with non-stationary Pareto-efficient solutions because there are no algorithms whose convergence to such kind of solution was proved in a non-special case (one exception is discussed in Section 7). As one can see, in both testbeds the solution found by SPL outperforms all other possible solutions of those games. The advantage of SPL is especially pronounced if the functional optimality criterion is $product$. In this case, SPL often converges to an extended strategy, which is typically more efficient in optimizing this criterion.

The curves of Figures 1 (a–c) reflect the evolution of the ELM value during learning in games from Random Games M. For each learning iteration, the curves present the current ELM value according to one of three functional criteria. (These values were averaged over $100$ experiments.) We can observe that for all three functional optimality criteria, the ELM value of SPL becomes close to the optimal one after a reasonably small number of learning iterations.

# 6 Discussions

So far, we have seen that SPL is efficient in repeated games. Now, let us talk about extensions of SPL to more complex settings, such as $n$-player, with $n > 2$, repeated games and multistate environments (like those usually modeled as stochastic games [7]). One of possible extensions of SPL to $n$-player repeated games is quite straightforward. Instead of two roles – leader and follower – there will be $n$ roles, one per player. For example, player $1$ assigned with the role $1$ will behave as leader. Any other player $p$ assigned with a role $1 < p \leq n$ will behave as follower whose leader is player $p-1$. Obviously, in this case each player $1 \leq p \leq n$ will maintain the counters $L^q$ and $K^q$ for all other players.

To make an extension of SPL to multistate environments, one could once again take advantage of the self-play setting. Depending on the nature of the environment, agents can use a certain (known to all agents) algorithm to compute a set of strategies for a given environment. For example, for a multi-robot motion coordination problem [7], these strategies can be possible trajectories of a robot. To each of these strategies robots can associate an action of a top level repeated game $\Gamma$. Then, a joint-action in the game $\Gamma$ is a pair of trajectories in the original multistate environment, and the corresponding reward is the cumulative reward of players after simultaneously executing this pair of trajectories.

# 7 Related Work

We would emphasize two most pertinent works related to our research. In the first one [6] the desired solution of the learning problem is correlated equilibrium. When several equilibria are possible, the author proposes to choose a unique one by using an "objective function", an analog of our functional criterion. However, this implies that agents have two opposite goals: (1) to be selfish (inclination to equilibrium solution implies the agents to be selfish) and (2) to want to sacrifice, by selecting, using the objective function, an equilibrium, which is probably sub-optimal to itself. Generally, if the agents are supposed to want to sacrifice, there is no need in seeking after an equilibrium solution.

In the second work [5], the authors propose an approach for learning of multi-step strategies, analogous to our extended strategies. When the

Table 1: Utility of SPL for different function optimality criteria compared to the utilities of other solutions.

*max*

| | SPL | BPE | BNE | WNE |
|---|---|---|---|---|
| CGs | **4.00** | 3.62 | 3.44 | 3.44 |
| RGs 2 | **88.71** | 82.11 | 80.02 | 78.00 |
| RGs 3 | **94.68** | 88.85 | 84.19 | 80.37 |
| RGs 5 | **98.18** | 94.29 | 87.83 | 77.73 |
| RGs 10 | **99.46** | 96.60 | 92.68 | 71.25 |

*sum*

| | SPL | BPE | BNE | WNE |
|---|---|---|---|---|
| CGs | **6.41** | 6.29 | 6.02 | 5.97 |
| RGs 2 | **140.73** | 126.88 | 128.10 | 124.84 |
| RGs 3 | **161.14** | 151.02 | 150.34 | 138.83 |
| RGs 5 | **174.05** | 162.14 | 157.98 | 139.73 |
| RGs 10 | **187.15** | 182.09 | 175.93 | 135.01 |

*product*

| | SPL | BPE | BNE | WNE |
|---|---|---|---|---|
| CGs | **10.43** | 9.96 | 9.06 | 8.94 |
| CGs 2 | **5179.85** | 4549.06 | 4432.09 | 3970.02 |
| CGs 3 | **6555.84** | 5987.72 | 5805.01 | 4720.69 |
| CGs 5 | **7715.53** | 7050.49 | 6725.64 | 5320.81 |
| CGs 10 | **8745.05** | 8097.79 | 7806.81 | 4414.34 |

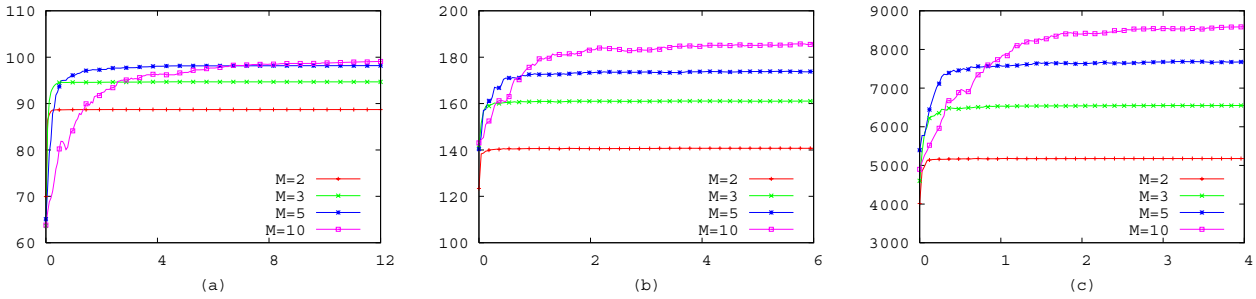



Figure 1: The evolution of ELM value of the learned policy in Random Games M according to different functional criteria: (a) $max$, (b) $sum$ and (c) $product$. The X axis represents learning iterations ($\times 10^3$); the Y axis represents ELM value.

length $l$ of a multi-step strategy is fixed to 1 (the only value used in their experiments), this yields in a relatively small number of learning iterations. However, by increasing $l$ (to allow more complex joint strategies) the number of joint strategies to explore becomes exponentially large, and only a small number of them is really interesting. In our approach, we find the best extended joint strategies directly, i.e., without enumeration of all pairs of action sequences of length $l$. Besides, this approach does not permit satisfying a given functional criterion.

## 8 Conclusions

In this paper, we presented a novel approach to multiagent learning in self-play. We argued that when the learning problem is a known (or controlled) self-play, a good learning algorithm should get additional benefit from this. Then, we presented the notion of functional optimality criterion, as opposed to relational optimality criteria such as Nash equilibrium. We demonstrated that the solution of a problem found by an algorithm seeking to satisfy a relational optimality criterion can be suboptimal if a functional optimality criterion needs to be satisfied. We then showed that in such problems, our algorithm is a better choice than a big class of classical multiagent algorithms for self-play. We also presented the notion of extended strategy and showed how it can be learned. These non-stationary strategies are especially efficient in satisfying the product optimality criterion.

## References

[1] B. Banerjee and J. Peng. Performance bounded reinforcement learning in strategic interactions. *Proceedings of AAAI-04*, 2004.

[2] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250, 2002.

[3] S.J. Brams. Theory of Moves. *American Scientist*, 81(6):562–570, 1993.

[4] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of AAAI'98*, 1998.

[5] J.W. Crandall and M.A. Goodrich. Learning to compete, compromise, and cooperate in repeated general-sum games. In *Proceedings ICML'05*, 2005.

[6] Amy Greenwald. Correlated-Q learning. In *In AAAI Spring Symposium*, 2003.

[7] J. Hu and M. Wellman. Nash Q-learning for general-sum stochastic games. *Journal of ML Research*, 4:1039–1069, 2003.

[8] M.L. Littman and P. Stone. A polynomial-time Nash equilibrium algorithm for repeated games. *Decision Support Systems*, 39(1):55–66, 2005.

[9] J. Nash. The Bargaining Problem. *Econometrica*, 18(2):155–162, 1950.

# Extraction de motifs dans des dialogues annotés par programmation dynamique

Émilie Chanoni* Thierry Lecroq† Alexandre Pauchet‡
Emilie.Chanoni@univ-rouen.fr Thierry.Lecroq@univ-rouen.fr Alexandre.Pauchet@insa-rouen.fr

* Université de Rouen, Psy.NCA EA 4306,
76821 Mont-Saint-Aignan Cedex, France

† Université de Rouen, LITIS EA 4108,
76821 Mont-Saint-Aignan Cedex, France

‡ INSA-Rouen, LITIS EA 4108,
BP 08 - 76801 Saint-Étienne-du-Rouvray, France

**Résumé :**
Dans le cadre de la modélisation du dialogue, nous nous intéressons à la recherche de régularités dans les échanges entre interlocuteurs, et plus particulièrement dans l'expression d'un processus mentaliste. Nous utilisons pour cela une grille mentaliste afin de coder l'expression des états mentaux et des émotions dans les énoncés.
Nous présentons une nouvelle heuristique pour calculer des alignements globaux et locaux de motifs en deux dimensions par programmation dynamique. Cette méthode consiste en une généralisation des formules de récurrence utilisées pour les alignements de séquences. A notre connaissance, cette étude est la première qui considère le calcul d'alignements locaux de motifs 2D, qui plus est avec des scores quelconques.
Nous avons évalué cette méthode sur la recherche de motifs répétés sur des annotations de dialogues entre parents et enfants de quatre ans lors de la narration de deux histoires enfantines.

**Mots-clés :** Dialogue annoté, extraction de motifs, identification de régularités, programmation dynamique

## 1 Introduction

La conception de systèmes informatiques interagissant efficacement avec l'utilisateur nécessite l'utilisation de modèles robustes et efficaces. Ces modèles doivent être définis non pas arbitrairement par le concepteur du sytème mais en s'appuyant sur une étude des mécanismes humains d'interaction et de communication. Par exemple, pour qu'une interface dialogique en langue naturelle soit agréable, elle ne doit pas s'appuyer uniquement sur un modèle de la tâche mais intégrer en sus un comportement langagier le plus naturel possible.

Il existe de nombreux modèles de dialogue, parmi lesquels on peut distinguer, comme le fait Lehuen [20], plusieurs approches distinctes :

– L'approche par la planification est fondée sur la reconnaissance de buts et la construction de plans dans les conversations. On citera comme exemples les modèles d'Allen et Perrault [1], de Balkanski et Hurault-Plantet [6] ou le système Artimis de Sadek [27].
– L'approche par la structuration cherche à dégager et à expliquer la structure des dialogues en décomposant les interactions verbales en unités de différents niveaux. Le modèle genevois de Roulet et Moeschler [26, 23] ou la théorie discursive de Vanderveken [33], par exemple, définissent les règles conversationnelles régissant ces unités.
– L'approche par l'interaction s'appuie sur des règles locales pour l'interprétation et la production d'actions communicatives durant les dialogues. Les exemples les plus connus sont les modèles de Trognon et Brassac [32], de Dessalles [14] et de Brassac et Pesty [8].

Il existe bien évidemment d'autres approches, ainsi que des approches mixtes, comme le modèle `Coala` [20], fondé sur la structuration et sur l'interaction.

La modélisation du dialogue par les approches citées précédemment s'appuie la plupart du temps sur un corpus de dialogues annotés, dont l'étude vise à repérer un ensemble de comportements récurrents significatifs, afin d'être en mesure de les expliquer. Lorsque les annotations sont judicieuses, ces comportements apparaissent alors sous la forme de motifs répétés dans les annotations. La représentation de ces répétitions peut se faire sous la forme d'automates [35, 7], d'automates temporisés [25], de réseaux de Petri [22], de diagrammes de séquences [31], *etc.* Le point commun de toutes ces représentations étant qu'elles s'appuient sur des motifs linéaires.

Dans le cas particulier de corpus de dialogues dont les annotations visent à caractériser un processus mentaliste, chaque énoncé doit être annoté par une série de codes. Il en résulte une recherche de répétitions non plus linéaires, mais

en deux dimensions, rendant impossible les approches classiques des études de corpus.

L'approche présentée ici répond tout à fait à cette problématique :

1. retranscription et annotation des dialogues ;
2. identification d'alignements de motifs en 2D par programmation dynamique ;
3. évaluation de la pertinence sémantique des motifs repérés (expertise) ;

La programmation dynamique [11] est une technique utilisée dans de multiples applications, voir par exemple [21, 16, 30, 29]. Les mots (séquences de symboles appartenant à un alphabet $\Sigma$) peuvent être facilement comparés à l'aide de cette technique. Des applications de ce problème incluent la bioinformatique, le traitement automatique de la langue naturelle ou bien le traitement de la parole. Les arbres peuvent aussi être comparés en utilisant des techniques similaires. Le lecteur peut se référer à [28] pour de plus amples détails. Il y a un grand nombre de travaux considérant la recherche exacte ou approchée en deux dimensions : étant donnés un motif $X$ et un texte $Y$, tous les deux en deux dimensions, trouver les occurrences de $X$ dans $Y$ (voir [3, 17]). Il y a eu des efforts pour indexer des matrices en utilisant des arbres de suffixes ou des tables de suffixes [15, 18, 24] mais ces index permettent de trouver des répétitions exactes et la généralisation pour trouver des motifs approchés est loin d'être immédiate. Très peu d'attention a été portée sur l'alignement de matrices à deux dimensions bien que les applications de ce problème incluent des domaines fondamentaux comme la fouille de donnés ou le traitement des images. [19] considère des motifs en deux dimensions de même taille alors que [5] considère seulement l'alignement global de deux motifs en deux dimensions. A notre connaissance, le présent article constitue la première tentative pour formaliser le calcul d'alignements locaux de deux motifs en deux dimensions par programmation dynamique. Récemment [4] a considéré le problème de la recherche de plus long sous motifs communs en dimensions $d > 2$ mais en utilisant des techniques différentes.

Nous donnons des formules de récurrence pour calculer des alignements globaux et locaux de motifs en deux dimensions de taille respective $M$ et $N$ en temps et espace $O(M \times N)$. Pour cela nous devons précalculer les similarités entre tous les préfixes de toutes les lignes et tous les préfixes de toutes les colonnes des deux motifs. Cela peut également être effectué en temps et espace $O(M \times N)$.

Une première évaluation de cette méthode sur la recherche de motifs répétés sur des annotations de dialogues entre parents et enfants de quatre ans lors de la narration de deux histoires enfantines a été réalisée.

Le reste de l'article est organisé comme suit : la section 2 rappelle les techniques de programmation dynamique pour l'alignement de séquences ; la section 3 présente la nouvelle méthode de programmation dynamique pour l'alignement de motifs en deux dimensions ; la section 4 montre l'évaluation réalisée, enfin la section 5 donne nos conclusions et des perspectives.

## 2 L'alignement de séquences

Les alignements sont généralement utilisés pour comparer les séquences. Ils sont largement répandus en bioinformatique. Ils constituent un moyen pour visualiser la ressemblance entre les séquences. Ils sont basés sur des notions de distance ou de similarité. Ils sont usuellement calculés par programmation dynamique. On considère deux types d'alignements de deux séquences $x$ et $y$ : les alignements globaux (qui considèrent la totalité des deux séquences) et les alignements locaux (qui permettent de détecter le segment de $x$ le plus similaire à un segment de $y$).

**Exemple**

**A C G – – A**
**A T G C T A**

est un alignement global de

**ACGA** et **ATGCTA**.

Une solution peut aussi être donnée sous forme de script d'édition comme suit :

| Opération | séquence résultante |
|---|---|
| substitution de **A** par **A** | **A** |
| substitution de **C** par **T** | **AT** |
| substitution de **G** par **G** | **ATG** |
| insertion de **C** | **ATGC** |
| insertion de **T** | **ATGCT** |
| substitution de **A** par **A** | **ATGCTA** |

### 2.1 Alignement global

Un alignement global de deux séquences $x$ et $y$ peut être obtenu en calculant la distance entre $x$ et $y$. La notion de distance entre deux séquences est largement utilisée pour comparer des fichiers. La commande **diff** d'UNIX implante un algorithme basé sur cette notion où chaque ligne des fichiers est considérée comme un symbole. La sortie d'une comparaison des deux fichiers effectuée par **diff** est le nombre minimum d'opérations d'édition (substitution, insertion ou suppression) pour transformer le premier fichier en le deuxième. Cette commande réalise un alignement global entre les deux fichiers.

On définit la distance d'édition entre deux séquences $x$ et $y$ comme étant le nombre minimum d'opérations d'édition élémentaires pour transformer $x$ en $y$. Les opérations d'édition élémentaires sont : substitution d'un symbole de $x$ par un symbole de $y$, suppression d'un symbole de $x$ et insertion d'un symbole de $y$. Il est également possible de calculer des alignements globaux en utilisant des scores de similarités plutôt qu'une distance. Un score est associé à chaque opération d'édition élémentaire. Pour $a, b \in \Sigma$ : $\mathit{Sub}(a, b)$ est le score de la substitution du symbole $a$ par le symbole $b$, $\mathit{Del}(a)$ est le score de la suppression du symbole $a$, $\mathit{Ins}(a)$ est le score de l'insertion du symbole $a$. Pour deux symboles $a$ et $b$, une valeur positive de $\mathit{Sub}(a, b)$ signifie que les deux symboles sont proches (ou similaires), et une valeur négative de $\mathit{Sub}(a, b)$ signifie que les deux symboles sont éloignés (ou dissimilaires). Nous pouvons maintenant définir le score d'édition généralisé entre deux séquences $x$ et $y$ par $d(x, y) = \max\{\text{score de } \gamma \mid \gamma \in \Gamma_{x,y}\}$ où $\Gamma_{x,y}$ est l'ensemble de toutes les séquences d'opérations d'édition qui transforment $x$ en $y$, et le score d'un élément $\gamma \in \Gamma_{x,y}$ est la somme des scores de ses opérations d'édition élémentaires.

Pour calculer $d(x, y)$ pour deux séquences $x$ et $y$ de longueur respective $m$ et $n$, on utilise une table à deux dimensions $t$ à $m+1$ lignes et $n+1$ colonnes telle que $t[i, j] = d(x[0\,..\,i], y[0\,..\,j])$ pour $i = 0, \ldots, m-1$ and $j = 0, \ldots, n-1$. Il s'ensuit que $d(x, y) = t[m-1, n-1]$.

Les valeurs de la table $t$ peuvent être calculées en suivant les formules de récurrence pour $i = 0, 1, \ldots, m-1$ et $j = 0, 1, \ldots, n-1$ :

$$t[-1, -1] = 0,$$

$$t[i, -1] = t[i-1, -1] + \mathit{Del}(x[i]),$$

$$t[-1, j] = t[-1, j-1] + \mathit{Ins}(y[j]),$$

$$t[i, j] = \max \begin{cases} t[i-1, j-1] + \mathit{Sub}(x[i], y[j]), \\ t[i-1, j] + \mathit{Del}(x[i]), \\ t[i, j-1] + \mathit{Ins}(y[j]). \end{cases}$$

La valeur à la position $(i, j)$ de la table $t$ ne dépend que des valeurs aux trois positions voisines $(i-1, j-1)$, $(i-1, j)$ et $(i, j-1)$ (voir [12]).

Une application directe de la formule de récurrence ci-dessus donne un algorithme en temps exponentiel pour calculer $t[m-1, n-1]$. Cependant la totalité de la table $t$ peut être calculée en temps quadratique, par une technique connue sous le nom de « programmation dynamique ». C'est une technique générale qui peut être utilisée pour résoudre différents types d'alignements.

Un alignement optimal (de score maximal) peut alors être produit. Cela consiste à effectuer un tracé arrière du calcul des valeurs de la table $t$ à partir de la position $(m-1, n-1)$ jusqu'à la position $(-1, -1)$.

### 2.2 Alignement local

Calculer un alignement local de deux séquences $x$ et $y$ consiste à trouver le segment de $x$ qui est le plus similaire à un segment de $y$.

Le score local d'édition de deux séquences $x$ et $y$ est défini par $s(x, y) =$ similarité maximale entre un segment de $x$ et un segment de $y$.

Pour calculer $s(x, y)$ pour deux séquences $x$ et $y$ de longueur respective $m$ et $n$, nous utilisons une table à deux dimensions $t_S$ à $m+1$ lignes et $n+1$ colonnes telle que $t_S[i, j] = \max\{s(x[\ell\,..\,i], y[k\,..\,j]) \mid 0 \leq \ell \leq i \text{ et } 0 \leq k \leq j\} \cup \{0\}$, pour $i = 0, \ldots, m-1$ et $j = 0, \ldots, n-1$. Donc $s(x, y) =$ valeur maximale dans $t_S$.

Les valeurs de la table $t_S$ peuvent être calculées avec les formules de récurrence suivantes : pour $i = 0, 1, \ldots, m-1$ et $j = 0, 1, \ldots, n-1$ :

$$t_S[-1,-1] = t_S[i,-1] = t_S[-1,j] = 0$$

$$t_S[i,j] = \max \begin{cases} t_S[i-1,j-1] \\ \qquad +\textit{Sub}(x[i], y[j]) \,, \\ t_S[i-1,j] + \textit{Del}(x[i]) \,, \\ t_S[i,j-1] + \textit{Ins}(y[j]) \,, \\ 0 \,. \end{cases}$$

Calculer les valeurs de $t_S$ pour un alignement local de $x$ et $y$ peut être fait en temps et espace $O(mn)$. Retrouver un alignement local peut être fait d'une manière similaire au cas des alignements globaux à ceci près que le tracé arrière doit commencer à une position d'une valeur maximale dans $t_S$ plutôt qu'à la position $(m-1, n-1)$.

# 3 Alignement de motifs 2D

Considérons maintenant deux motifs rectangulaires $X = X[0\,..\,m_1-1, 0\,..\,n_1-1]$ et $Y = Y[0\,..\,m_2-1, 0\,..\,n_2-1]$, de taille respective $M = m_1 \times n_1$ et $N = m_2 \times n_2$. Chaque élément $X[i,j]$ avec $0 \le i \le m_1-1$ et $0 \le j \le n_1-1$ et $Y[k,\ell]$ avec $0 \le k \le m_2-1$ et $0 \le \ell \le n_2-1$ appartient à l'alphabet $\Sigma$.

Nous voulons maintenant calculer le score maximum des opérations d'insertion, de suppression ou de substitution de symboles individuels pour transformer $X$ en $Y$. Les symboles peuvent être insérés, supprimés ou substitués séparément ou par portion de lignes ou par portion de colonnes.

## 3.1 Alignement global

Aligner $X$ et $Y$ en utilisant la programmation dynamique consiste à généraliser les formules de récurrence utilisées pour l'alignement de séquences.

Pour cela nous avons besoin de quatre tables à deux dimensions $D_R$, $D_C$, $I_R$ et $I_C$ définies comme suit pour $0 \le i \le m_1-1$, $0 \le j \le n_1-1$, $0 \le i \le m_2-1$ et $0 \le j \le n_2-1$ :

$$D_R[i,j] = \textstyle\sum_{p=0}^{j} \textit{Del}(X[i,p])$$

$$D_C[i,j] = \textstyle\sum_{p=0}^{i} \textit{Del}(X[p,j])$$

$$I_R[i,j] = \textstyle\sum_{p=0}^{j} \textit{Ins}(Y[i,p])$$

$$I_C[i,j] = \textstyle\sum_{p=0}^{i} \textit{Ins}(Y[p,j]).$$

En d'autres termes, $D_R[i,j]$ est le score de la suppression du préfixe de longueur $j+1$ de la ligne $i$ de $X$, $D_C[i,j]$ est le score de la suppression du préfixe de longueur $i+1$ de la colonne $j$ de $X$, $I_R[i,j]$ est le score de l'insertion du préfixe de longueur $j+1$ de la ligne $i$ de $Y$ et $I_C[i,j]$ est le score de l'insertion du préfixe de longueur $i+1$ de la ligne $j$ de $Y$. Les tables $D_R$ et $D_C$ peuvent être calculées en temps et espace $O(m_1 \times n_1)$. Les tables $I_R$ et $I_C$ peuvent être calculées en temps et espace $O(m_2 \times n_2)$.

Nous allons aussi utiliser deux tables à quatre dimensions $R$ et $C$ de taille $m_1 \times n_1 \times m_2 \times n_2$ definies comme suit : $R[i,j,k,\ell] = d(X[i,0\,..\,j], Y[k,0\,..\,\ell])$, et $C[i,j,k,\ell] = d(X[0\,..\,i,j], Y[0\,..\,k,\ell])$. En fait $R[i,j,k,\ell]$ contient la distance entre le préfixe de longueur $j+1$ de la ligne $i$ de $X$ et le préfixe de longueur $\ell+1$ de la ligne $k$ de $Y$. De manière similaire $C[i,j,k,\ell]$ contient la distance entre le préfixe de longueur $i+1$ de la colonne $j$ de $X$ et le préfixe de longueur $k+1$ de la colonne $\ell$ de $Y$. Les deux tables $R$ et $C$ peuvent être calculées en temps et espace $O(m_1 \times n_1 \times m_2 \times n_2)$.

Nous utilisons alors une table à quatre dimensions $T$ de taille $(m_1+1) \times (n_1+1) \times (m_2+1) \times (n_2+1)$ définie comme suit : $T[i,j,k,\ell] = \max\{\text{score de } \gamma \mid \gamma \in \Gamma_{X,Y}\}$ où $\Gamma_{X,Y}$ est l'ensemble de toutes les séquences d'opérations d'édition qui transforment $X$ en $Y$, et le score d'un élément $\gamma \in \Gamma_{x,y}$ est la somme des scores de ses opérations d'édition élémentaires.

Les valeurs de la table $T$ peuvent être calculées pour $0 \le i \le m_1-1$, $0 \le j \le n_1-1$, $0 \le k \le m_2-1$ et $0 \le \ell \le n_2-1$ comme présenté FIG. 1 et FIG. 2.

Nous avons les initialisations marginales suivantes : $T[-1,j,k,\ell] = T[i,-1,k,\ell] = (k+1) \times (\ell+1)$ et $T[i,j,-1,\ell] = T[i,j,k,-1] = (i+1) \times (j+1)$ pour $0 \le i \le m_1-1$, $0 \le j \le n_1-1$, $0 \le k \le m_2-1$ et $0 \le \ell \le n_2-1$.

Ensuite un tracé arrière peut être effectué à partir de $T[m_1, n_1, m_2, n_2]$ comme dans le cas de l'alignement de séquences.

L'alignement global de deux motifs rectangulaires de taille respective $M = m_1 \times n_1$ et $N = m_2 \times n_2$ peut être effectué en temps et espace $O(M \times N)$.

$$T[i,j,k,\ell] = \max \begin{cases} T[i-1,j,k,\ell] + D_R[X[i,0\,..\,j]] \\ T[i,j-1,k,\ell] + D_C[X[0\,..\,i,j]] \\ T[i,j,k-1,\ell] + I_R[Y[k,0\,..\,\ell]] \\ T[i,j,k,\ell-1] + I_C[Y[0\,..\,k,\ell]] \\ T[i-1,j,k-1,\ell] + R[i,j,k,\ell] \\ T[i,j-1,k,\ell-1] + C[i,j,k,\ell] \\ T[i-1,j-1,k-1,\ell-1] + C[i-1,j,k-1,\ell] + R[i,j,k,\ell] \\ T[i-1,j-1,k-1,\ell-1] + C[i,j,k,\ell] + R[i,j-1,k,\ell-1] \end{cases} \quad (1)$$

FIG. 1 – Calcul de $T$, avec $0 \leq i \leq m_1 - 1$, $0 \leq j \leq n_1 - 1$, $0 \leq k \leq m_2 - 1$ et $0 \leq \ell \leq n_2 - 1$

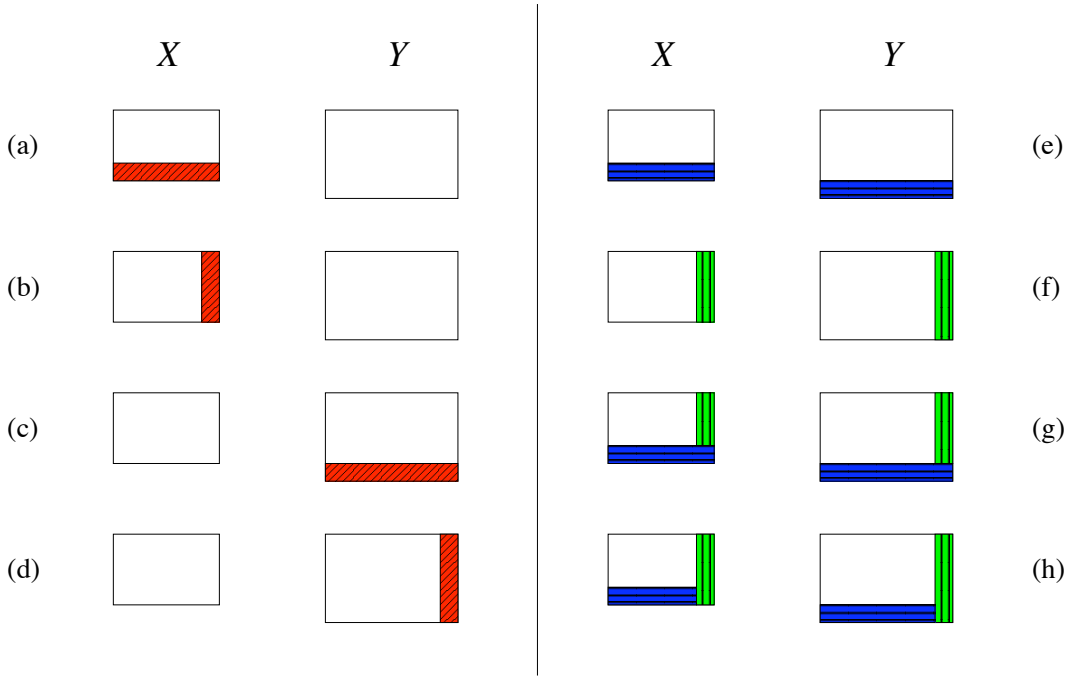



FIG. 2 – (a) : suppression de $X[i,0\,..\,j]$ ; (b) : suppression de $X[0\,..\,i,j]$ ; (c) : insertion de $Y[k,0\,..\,\ell]$ ; (d) : insertion de $Y[0\,..\,k,\ell]$ ; (e) : substitution de $X[i,0\,..\,j]$ par $Y[k,0\,..\,\ell]$ ; (f) : substitution de $X[0\,..\,i,j]$ par $Y[0\,..\,k,\ell]$ ; (g) : substitution de $X[i,0\,..\,j]$ par $Y[k,0\,..\,\ell]$ et substitution de $X[0\,..\,i-1,j]$ par $Y[0\,..\,k-1,\ell]$ ; (h) : substitution de $X[i,0\,..\,j-1]$ par $Y[k,0\,..\,\ell-1]$ et substitution de $X[0\,..\,i,j]$ par $Y[0\,..\,k,\ell]$.

**Exemple**

Avec les motifs suivants

| $X$ | 0 | 1 | 2 |
|---|---|---|---|
| 0 | **A** | **B** | **C** |
| 1 | **D** | **E** | **F** |
| 2 | **G** | **H** | **I** |
| 3 | **J** | **K** | **L** |

et

| $Y$ | 0 | 1 |
|---|---|---|
| 0 | **E** | **C** |
| 1 | **H** | **I** |
| 2 | **K** | **L** |

et en considérant les scores suivants pour les opérations d'édition ($\mathit{Ins}(a) = \mathit{Del}(a) = -1$, $\mathit{Sub}(a, a) = 2$ et $\mathit{Sub}(a, b) = -1$ pour $a, b \in \Sigma$ tels que $a \neq b$), un script possible en remontant depuis $T[3, 2, 2, 1] = 6$, comme présenté FIG. 3.

### 3.2 Alignement local

Pour calculer un alignement local entre deux motifs rectangulaires $X$ et $Y$ on utilise deux tables à quatre dimensions $R_S$ et $C_S$ de taille $m_1 \times n_1 \times m_2 \times n_2$ définies comme suit : $R_S[i, j, k, \ell] = s(X[i, 0\,..\,j], Y[k, 0\,..\,\ell])$, et $C_S[i, j, k, \ell] = s(X[0\,..\,i, j], Y[0\,..\,k, \ell])$. En fait, $R_S[i, j, k, \ell]$ contient la similarité maximale entre un suffixe de $X[i, 0\,..\,j]$ et un suffixe de $Y[k, 0\,..\,\ell]$. De manière similaire $C_S[i, j, k, \ell]$ contient la similarité maximale entre un suffixe de $X[0\,..\,i, j]$ et un suffixe de $Y[0\,..\,k, \ell]$. Ces deux tables peuvent être calculées en utilisant la formule de récurrence habituelle pour les séquences en temps et espace $O(m_1 \times n_1 \times m_2 \times n_2)$.

Ensuite on utilise une table à quatre dimensions $T_S$ de taille $(m_1 + 1) \times (n_1 + 1) \times (m_2 + 1) \times (n_2 + 1)$ définie comme suit :

$T_S[i, j, k, \ell] = \max\{s(X[i'\,..\,i, j'\,..\,j], Y[k'\,..\,k, \ell'\,..\,\ell]) \mid 0 \leq i' \leq i, 0 \leq j' \leq j, 0 \leq k' \leq k \text{ et } 0 \leq \ell' \leq \ell\} \cup \{0\}$.

On note : $r = R_S[i, j, k, \ell]$, $c = C_S[i, j, k, \ell]$, $r' = R_S[i, j - 1, k, \ell - 1]$, $c' = C_S[i - 1, j, k - 1, \ell]$ et $q = \mathit{Del}(X[i, j]) + \mathit{Ins}(Y[k, \ell])$.

Les valeurs de la table $T_S$ peuvent être calculées à l'aide de l'équation présentée FIG. 4.

Nous avons les initialisations marginales :

$$\begin{aligned} T_S[-1, j, k, \ell] &= T_S[i, -1, k, \ell] \\ &= T_S[i, j, -1, \ell] \\ &= T_S[i, j, k, -1] \\ &= 0 \end{aligned}$$

pour $0 \leq i \leq m_1 - 1$, $0 \leq j \leq n_1 - 1$, $0 \leq k \leq m_2 - 1$ et $0 \leq \ell \leq n_2 - 1$.

Les valeurs de la table $T_S$ de taille $(m_1 + 1) \times (n_1 + 1) \times (m_2 + 1) \times (n_2 + 1)$ permettant de détecter des motifs similaires peuvent être calculées en temps et espace $O(m_1 \times n_1 \times m_2 \times n_2)$.

La procédure de tracé arrière doit commencer à la position contenant une valeur maximale plutôt qu'à la position $(m_1 - 1, n_1 - 1, m_2 - 1, n_2 - 1)$.

**Exemple**

Avec le système de score suivant : $\mathit{Sub}(a, a) = 1$, $\mathit{Sub}(a, b) = -1$ et $\mathit{Ins}(a) = \mathit{Del}(a) = -1$, pour $a, b \in \Sigma$ tels que $a \neq b$, FIG. 5 montre sur la gauche les deux motifs à aligner et sur la droite le meilleur alignement local où le symbole **c** sur la troisième ligne du second motif est inséré, tandis que les symboles **ghi** de la troisième ligne du premier motif sont substitués par les symboles **aba** de la quatrième ligne du second motif.

## 4 Application

L'algorithme a été testé sur des dialogues entre parents et enfants de 4 ans lors de la narration d'une histoire enfantine dont le contexte est mentaliste. Chaque énoncé est tout d'abord retranscrit et codé selon la grille de Chanoni [10, 9]. Nous avons ensuite réalisé les alignements de toutes les paires de dialogues distincts, ainsi que les alignements intra-dialogues, afin d'en extraire des régularités dans les comportements verbaux des parents et des enfants.

### 4.1 Retranscription et codage des dialogues

La grille utilisée pour le codage des dialogues est celle de Chanoni [10, 9], conçue spécialement dans le cadre d'une étude psychologique portant sur les interprétations mentalistes des comportements d'un personnage. L'objectif étant de repérer les caractéristiques dialogiques, sémantiques et pragmatiques du discours que proposent les adultes pour expliquer le comportement d'un personnage soumis à une fausse croyance [34], selon l'âge de l'enfant auquel ils s'adressent.

Cette grille est composée de cinq colonnes, chaque colonne comporte entre 2 et 7 codages (6, 3, 7, 2, 2) (voir FIG. 6) :

– Les indices relevés dans la première colonne concernent la nature de l'énoncé. Il peut s'a-

| Opération | Score |
|---|---|
| Substitution de $X[0\,..\,3, 2]$ par $Y[0\,..\,2, 1]$ | 5 |
| Substitution de $X[0\,..\,3, 1]$ par $Y[0\,..\,2, 0]$ | 5 |
| Suppression de $X[0\,..\,3, 0]$ | $-4$ |

FIG. 3 – Exemples de scores pour les opérations d'édition

$$T_S[i,j,k,\ell] = \max \begin{cases} T_S[i-1,j,k,\ell] + \textit{Del}(X[i,j]) \\ T_S[i,j-1,k,\ell] + \textit{Del}(X[i,j]) \\ T_S[i,j,k-1,\ell] + \textit{Ins}(Y[k,\ell]) \\ T_S[i,j,k,\ell-1] + \textit{Ins}(Y[k,\ell]) \\ T_S[i-1,j,k-1,\ell] + (r \text{ si } r \neq 0 \text{ sinon } q) \\ T_S[i,j-1,k,\ell-1] + (c \text{ si } c \neq 0 \text{ sinon } q) \\ T_S[i-1,j-1,k-1,\ell-1] + (c' + r \text{ si } c', r \neq 0 \text{ sinon } q) \\ T_S[i-1,j-1,k-1,\ell-1] + (c + r' \text{ si } c, r' \neq 0 \text{ sinon } q) \\ 0 \end{cases} \tag{2}$$

FIG. 4 – Calcul de $T_S$, avec $0 \leq i \leq m_1 - 1$, $0 \leq j \leq n_1 - 1$, $0 \leq k \leq m_2 - 1$ et $0 \leq \ell \leq n_2 - 1$

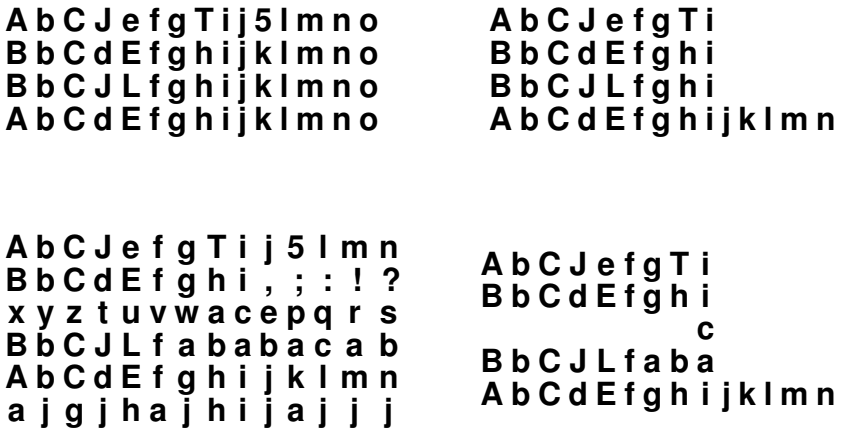



FIG. 5 – Alignement local de motifs 2D.

Dialogue b7

⋮
24 Parent on va la retrouver ta couronne
25 Parent t'inquiète pas
26 Parent donc là ils se cachent
27 Parent ils cherchent
28 Parent qui pourrait avoir pris la couronne
29 Enfant elle dedans, elle est dedans la couronne
30 Parent donc ils suspectent plein de monde,
Cornélius, Céleste, la vieille dame
31 Parent qui a bien pu prendre la couronne ?
32 Enfant la couronne elle est dedans
33 Parent tu crois ? !
34 Enfant oui
35 Parent mais Babar il ne sait pas qu'elle est dedans
36 Parent donc il se dit que c'est une bombe, la couronne
37 Parent ou je ne sais quoi ?
⋮

Grille mentaliste

| | | | | | |
|---|---|---|---|---|---|
| 24 | a | [ | { | ) | ] |
| 25 | A | P | E | ) | ] |
| 26 | A | P | B | ) | ] |
| 27 | a | [ | { | ) | ] |
| 28 | q | [ | { | ) | ] |
| 29 | a | [ | { | ) | ] |
| 30 | A | P | Y | C | J |
| 31 | q | [ | { | ) | ] |
| 32 | a | [ | { | ) | ] |
| 33 | Q | H | K | ) | ] |
| 34 | a | [ | { | ) | ] |
| 35 | A | P | N | O | J |
| 36 | A | P | N | C | J |
| 37 | A | R | N | ) | ] |

FIG. 6 – Codage des dialogues

gir d'une affirmation (a ou A), d'une question (q ou Q). L'état de majuscule (A, Q) correspond à la présence d'un état mental. Est aussi relevé dans cette colonne s'il s'agit d'une demande d'attention sur l'histoire (AH) ou d'une demande d'attention générale (AG).
– La deuxième colonne concerne la référenciation de l'énoncé. L'énoncé peut se réfèrer au personnage (P), à l'interlocuteur (H) ou au locuteur (R). La référenciation n'est codée qu'en association avec la présence d'un état mental.
– La troisième colonne relève les états mentaux. Les partenaires du dialogue peuvent décrire l'émotion (E), la volition (V), la cognition observable (B) ou non observable (N), l'épistémie (K), l'hypothèse (Y) et la surprise (S). Ici la surprise est distinguée des émotions pour ses liens avec la croyance dont elle est issue.
– Les deux dernières colonnes sont consacrées aux justifications par cause / conséquence (C) ou par opposition (O) qui peuvent être appliquées soit pour expliquer l'histoire (J), soit pour expliquer une situation par l'évocation d'un contexte personnel (CP).

Les parenthèses, accolades et autres crochets ([ , , ), ]) sont utilisés pour coder l'absence d'indice dans une colonne donnée.

Par exemple, nous présentons de façon plus précise le codage de l'énoncé figurant ligne 35 du dialogue b7 (voir FIG. 6) : " *mais Babar, il ne sait pas qu'elle est dedans* ". Cet énoncé est une assertion contenant un état mental, c'est pourquoi nous codons A dans la première colonne. Cet état mental, référencé au personnage (" *il ne sait pas* "), est codé P dans la deuxième colonne. L'état mental (" *sait* "), relatif à la cognition non observable, est codé N dans la troisième colonne. Nous relevons par ailleurs une justification par opposition repérée grâce à la conjonction de coordination " *mais* ", que nous codons O dans la quatrième colonne. Le contexte de cette justification est codé dans la cinquième et dernière colonne. Pour cet énoncé, le contexte fait référence à l'histoire en tant que telle (" *qu'elle [la couronne de Babar] est dedans* ") que nous codons J.

### 4.2 Extraction de motifs par programmation dynamique

Nous avons ensuite réalisé les alignements de toutes les paires de dialogues et les alignements intra-dialogues du corpus de dialogues présenté précédemment, afin de repérer des motifs approchés (ou flous) se répétant.

Une matrice de substitution a été spécifiquement construite pour cette application, afin d'exprimer les opérations d'édition élémentaires. Cette matrice peut être vue comme une représentation des liens existants entre les différents indices du codage. Le lien entre deux indices est représenté par une valeur allant de 0 à 10. Lorsque deux

indices sont liés par une valeur faible, cela indique qu'ils ne peuvent pas être substitués l'un par l'autre. A l'inverse, deux indices liés par une valeur élevée peuvent être rapprochés et donc substitués l'un par l'autre.

Pour cette étude nous avons souhaité centrer notre analyse sur les états mentaux et leurs justifications mettant de côté les éléments indiquant la nature grammaticale de l'énoncé. Ainsi, nous avons choisi de lier plus fortement entre elles les catégories référenciation (2ème colonne), états mentaux (3ème colonne), justifications (4ème colonne) et contexte (5ème colonne). Nous avons volontairement distingué la catégorie nature de l'énoncé (1ère colonne) des autres catégories.

Il faut noter que la grille choisie pour tester l'algorithme n'est pas dédiée à la modélisation du dialogue mais a été conçue particulièrement pour une étude psychologique donnée. Elles contient nottament des redondances dont le biais est levé par les valeurs choisies pour la matrice de substitution.

Les données présentées ici sont issues de la comparaison de deux dialogues entre deux mères et leur enfant de 4 ans (voir dialogues b7 et b9, FIG. 7 et FIG. 8).

L'algorithme fait apparaître des motifs en forme de L, se répétant au sein des deux dialogues (b7 et b9). Ces motifs flous peuvent être décomposés en deux parties significatives :

- Le premier ensemble est composé d'une série de lignes (partie verticale du L) qui comportent des indices relatant la présence d'un état mental associé à une référenciation.
- La deuxième partie forme la partie horizontale du L (lignes 35 et 36 pour le dialogue b7 par exemple) pour lesquelles nous notons des énoncés comportant des états mentaux associés à des justifications.

L'intérêt de ce motif réside dans le fait qu'il semble caractériser à la fois une organisation dialogique et un contenu. En effet, le premier ensemble du motif (partie verticale du L) fait apparaître au moins trois énoncés mentalistes au milieu d'une petite dizaine d'énoncés. Il est intéressant de noter qu'il ne s'agit pas d'une reformulation d'un seul état mental, mais de l'utilisation diversifiée d'états mentaux. Dans le dialogue b7, nous relevons la cognition observable (B), l'hypothèse (Y), l'épistémie (K) et la cognition non observable (N). La deuxième partie du motif est composée de deux énoncés complexes qui regroupent chacun une justification associée à une interprétation mentaliste.

Ainsi, notre hypothèse est que les parents prennent soin de proposer plusieurs interprétations mentalistes de la situation, en décrivant et précisant les états mentaux en jeu, tout en prenant soin de les aérer par des énoncés simples (assertions ou questions). Les parents peuvent ensuite énoncer une double justification complexe, précisant les liens de cause, de conséquence ou d'opposition qui déterminent les états mentaux du personnage et ses comportements. Cette préparation de l'enfant à comprendre une justification mentaliste que nous venons de décrire est un fait délicat, voire impossible à observer " à l'oeil nu ". En effet, pour cet exemple, ce motif a été mis en lumière à l'intérieur d'une série de plus de 37 et 86 lignes (respectivement pour les dialogues b7 et b9).

À terme, tous les motifs détectés seront répertoriés et recherchés dans tous les dialogues. Ainsi pour chaque symbole de tous les dialogues, il sera possible de savoir dans quel motif il apparaît et quels sont les dialogues (avec les positions) dans lesquels ces motifs apparaissent. Le but étant d'aider un expert à détecter des motifs significatifs.

## 5 Conclusion et perspectives

Nous avons présenté une méthode très générique pour aligner des motifs en deux dimensions par programmation dynamique en temps et espace quadratique. Cette méthode permet de calculer des alignements globaux et locaux. Nous avons appliqué cette méthode à l'extraction de motifs dans des dialogues entre parents et enfants de quatre ans, annotés par une grille mentaliste.

Une première difficulté consiste à visualiser les alignements ainsi calculés. En effet, le nombre de motifs qui peuvent être mis en évidence est assez conséquent mais nécessitent à chaque fois une expertise humaine afin d'en évaluer la pertinence. De plus, il n'existe pas à ce jour de méthode d'affichage pratique permettant de visualiser et comparer aisément deux motifs approchés. Enfin, quand un motif se répète dans plusieurs dialogues, il est aussi difficile de choisir un motif " consensuel ", permettant de représenter toutes les occurrences.

Du point de vue technique, beaucoup de

Dialogue b7

| | | | | | |
|---|---|---|---|---|---|
| 24 | a | [ | { | ) | ] |
| 25 | A | P | E | ) | ] |
| 26 | A | P | B | ) | ] |
| 27 | a | [ | { | ) | ] |
| 28 | q | [ | { | ) | ] |
| 29 | a | [ | { | ) | ] |
| 30 | A | P | Y | C | J |
| 31 | q | [ | { | ) | ] |
| 32 | a | [ | { | ) | ] |
| 33 | Q | H | K | ) | ] |
| 34 | a | [ | { | ) | ] |
| 35 | A | P | N | O | J |
| 36 | A | P | N | C | J |
| 37 | A | R | N | ) | ] |

Dialogue b9

| | | | | | |
|---|---|---|---|---|---|
| 8 | q | [ | { | ) | ] |
| 9 | A | P | B | ) | ] |
| 10 | q | [ | { | ) | ] |
| 11 | q | [ | { | ) | ] |
| 12 | q | [ | { | ) | ] |
| 13 | A | P | B | ) | ] |
| 14 | a | [ | { | ) | ] |
| 15 | A | P | N | ) | ] |
| 16 | q | [ | { | ) | ] |
| 17 | a | [ | { | ) | ] |
| 18 | a | [ | { | ) | ] |
| 19 | A | P | V | ) | ] |
| 20 | A | P | B | O | J |
| 21 | A | P | S | O | J |
| 22 | a | [ | { | ) | ] |
| 23 | a | [ | { | ) | ] |

FIG. 7 – Alignement de motifs mentalistes

Dialogue b7

24 Parent on va la retrouver ta couronne
25 Parent t'inquiéte pas
26 Parent donc là ils se cachent
27 Parent ils cherchent
28 Parent qui pourrait avoir pris la couronne
29 Enfant elle dedans, elle est dedans la couronne
30 Parent donc ils suspectent plein de monde, Cornélius, Céleste, la vieille dame
31 Parent qui a bien pu prendre la couronne ?
32 Enfant la couronne elle est dedans
33 Parent tu crois ? !
34 Enfant oui
35 Parent mais Babar il ne sait pas qu'elle est dedans
36 Parent donc il se dit que c'est une bombe, la couronne
37 Parent ou je ne sais quoi ?

Dialogue b9

7 Parent même son ami Zéphir la cherche partout avec sa loupe
8 Parent mais où est elle donc passée
9 Parent Babar remarque Cornélius donner un paquet à l'intrus
10 Parent Qu'est ce qui peut y avoir dedans ?
11 Parent mais qui est donc cet individu masqué
12 Parent qui a volé la couronne
13 Parent Babar la démasque !
14 Parent c'est la reine céleste !
15 Parent il se pose bien des questions
16 Parent pourquoi donc la reine Céleste s'est déguisée.
17 Parent Babar va donc chez la vieille dame lui demander
18 Enfant oui
19 Parent la vieille dame ne veut pas qu'il rentre !
20 Parent mais derrière se cachait
21 Parent pour lui une surprise en réalité
22 Parent puis Babar rentra chez lui

FIG. 8 – Motifs mentalistes similaires repérés dans deux dialogues différents

questions peuvent être posées. Est-il possible d'aligner des motifs 2D en temps sous-quadratique comme c'est le cas pour les séquences ? (voir [13]). Est-il possible de développer une heuristique similaire à BLAST [2] pour rapidement aligner un motif 2D contre une banque de motifs 2D ? D'autre part, il est nécessaire d'accompagner la méthode proposée d'une validation statistique pour estimer la pertinence des résultats trouvés. Enfin, il faut noter que cette méthode n'a pour l'instant été appliquée que sur des dialogues de taille modeste, avec une grille d'annotations relativement restreinte. Dans le cas d'un corpus plus conséquent (principalement concernant la longueur des dialogues et le volume des annotations) des méthodes d'approximation devront être envisagées.

Au-delà de l'intérêt scientifique et théorique, cette méthode peut, à terme, non seulement fournir un outil d'aide au diagnostic dans le cadre de situations dialogiques asymétriques mais pourrait plus généralement fournir un outil à la formation à l'entretien. Imaginons un jeune enfant qui vient consulter pour une douleur à l'abdomen. L'enfant et le médecin n'ont ni le même niveau de langage, ni le même niveau de représentation du corps et de la douleur. Pourtant le diagnostic ne se fera que sur la base des propos que l'enfant aura eu à propos de sa douleur. Cette méthode en complément d'un entretien semi-dirigé pourrait permettre de capturer des informations non saisissables d'emblée et pourtant nécessaires au diagnostic. Par ailleurs, cette méthode accompagnée d'une expertise psychologique permet de relever les techniques ou les motifs dialogiques efficaces. Ainsi, la formation aux techniques d'entretiens pourrait s'en trouver améliorée. Nous pourrions penser qu'un jour cette méthode pourrait assister en temps réel le locuteur dans son entretien.

## Références


[1] J. Allen and R. Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15 :143–178, 1980.

[2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. A basic local alignment search tool. *J. Mol. Biol.*, 215 :403–410, 1990.

[3] A. Amir, G. Benson, and M. Farach. An alphabet independent approach to 2-dimensional pattern matching. *SIAM J. Comput.*, 23(2) :313–323, 1994.

[4] A. N. Arslan. A largest common d-dimensional subsequence of two d-dimensional strings. In *FCT*, volume 4639 of *LNCS*, pages 40–51, 2007.

[5] R. A. Baeza-Yates. Similarity in two-dimensional strings. In *COCOON*, volume 1449 of *LNCS*, pages 319–328, 1998.

[6] C. Balkanski and M. Hurault-Plantet. Cooperative requests and replies in a collaborative dialogue model. *International Journal of Human-Computer Studies*, 53(6) :915–968, 2000.

[7] K. Bouzouba and B. Moulin. Les interactions distribuées : Approche basée sur le point de vue personnel des agents. In *MFI*, pages 15–30, 2001.

[8] C. Brassac and S. Pesty. Simuler la conversation : un défi pour les systèmes multiagents. In *Analyse et simulation de conversations. De la théorie des actes de discours aux systèmes multiagents*, chapter 9, pages 317–345. L'interdisciplinaire, 1999.

[9] É. Chanoni. Comment les mères racontent une histoire de fausses croyances à leur enfant de 3 à 5 ans ? *Enfance*, 2004.

[10] É. Chanoni. *Rôle du langage dans le développement de la théorie de l'esprit chez les enfants de 3 à 5 ans : contexte verbal et contexte narratif*. PhD thesis, Université de Lille, 2004.

[11] T. H. Cormen, C. E. Leiserson, R L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, 2001.

[12] M. Crochemore, C. Hancart, and T. Lecroq. *Algorithms on strings*. Cambridge University Press, 2007.

[13] M. Crochemore, G. M. Landau, and M. Ziv-Ukelson. A subquadratic sequence alignment algorithm for unrestricted scoring matrices. *SIAM J. Comput.*, 32(6) :1654–1673, 2003.

[14] J.-L. Dessalles. *Modèle cognitif de la communication spontanée appliqué à l'apprentissage*. PhD thesis, ENST Paris, 1993.

[15] R. Giancarlo and R. Grossi. Suffix tree data structures for matrices. In A. Apostolico and Z. Galil, editors, *Pattern matching algorithms*, chapter 10, pages 293–340. Oxford University Press, 1997.

[16] E. Hansen, D. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th National Conference on AI*, pages 709–715, 2004.

[17] J. Kärkkäinen and E. Ukkonen. Two and higher dimensional pattern matching in optimal expected time. In *SODA*, pages 715–723, 1994.

[18] D. K. Kim, Y. A. Kim, and K. Park. Generalizations of suffix arrays to multidimensional matrices. *Theoret. Comput. Sci.*, 302(1–3) :223–238, 2003.

[19] K. Krithivasan and R. Sitalakshmi. Efficient two-dimensional pattern matching in the presence of errors. *Inf. Sci.*, 43 :169–184, 1987.

[20] J. Lehuen. *Un modèle de dialogue dynamique et générique intégrant l'acquisition de sa compétence linguistique - Le système* COALA. PhD thesis, Université de Caen, 1997.

[21] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille. Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology*, 6(5) :367–381, 2002.

[22] H. Mazouzi. *Ingénierie des protocoles d'interaction : des systèmes distribués aux systèmes multi-agents*. PhD thesis, Université Paris IX, 2001.

[23] J. Moeschler. *Modélisation du dialogue*. Hermès, 1989.

[24] J. C. Na, R. Giancarlo, and K. Park. Online construction of two-dimensional suffix trees in $O(n^2 \log n)$ time. *Algorithmica*, 48(2) :173–186, 2007.

[25] A. Pauchet. *Modélisation cognitive d'interactions humaines dans une cadre de planification multi-agents*. PhD thesis, Université Paris Nord, 2006.

[26] E. Roulet, A. Auchlin, J. Moeschler, C. Rubattel, and M. Schelling. *L'articulation du discours en francais*. Peter Lang, 1985.

[27] D. Sadek, P. Brétier, and F. Panaget. ARTIMIS : Natural dialogue meets rational agency. In *Actes d'International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 1030–1035, Yokohama, Japon, 1997.

[28] D. Sankoff and J. B. Kruskal. *Time warps, string edits, and macromolecules : the theory and practice of sequence comparison*. Cambridge University Press, 1999. 2nd edition.

[29] R. S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the 7th International Conference on Machine Learning*, pages 216–224, 1990.

[30] D. Szer and F. Charpillet. Point-based dynamic programming for DEC-POMDPs. In *Proceedings of the 21st National Conference on AI*, 2006.

[31] FIPA. FIPA request interaction protocol specification. Technical report, Foundation for Intelligent Physical Agents, 2002. Avail. at `http://www.fipa.org`.

[32] A. Trognon and C. Brassac. L'enchaînement conversationnel. In *Cahier de linguistique francaise*, volume 13, pages 76–116. Cazaubon, 1992.

[33] D. Vanderveken. La structure logique des dialogues intelligents. In *Analyse et simulation de conversations. De la théorie des actes de discours aux systèmes multi-agents*, chapter 2, pages 61–100. L'interdisciplinaire, 1999.

[34] H. Wimmer and J. Perner. Beliefs about beliefs : representation and constraining function of the wrong beliefs in young children's understandig of deception. *Cognition*, (13) :103–128, 1983.

[35] T. Winograd and F. Flores. *Understanding Computers and Cognition : A New Foundation for Design*. Ablex, 1986.

# La confiance dans les relations entre agents : essai de modélisation en Logique Modale

R. Demolombe⋆
robert.demolombe@orange.fr

⋆Institut de Recherche en Informatique de Toulouse
Toulouse, France

**Résumé :**
La notion de confiance fait référence à plusieurs notions qui sont analysées de façon informelle, d'abord sur un exemple, puis dans le cas général. On propose ensuite un essai de formalisation en logique modale où l'on distingue les justifications et les motivations de la confiance. On montre que ces justifications peuvent être de nature soit empirique soit analytique, et que ces dernières font appel à la confiance dans d'autres propriétés : la capacité de créer un certain état des choses, la possibilité de déclencher des actions à partir d'intentions, le respect des normes, l'intérêt mutuel de deux agents, ou l'intérêt d'un agent pour l'autre.
La formalisation consiste à expliciter les relations entre les diverses modalités qui servent à définir ces notions. La relation entre antécédent et conséquent est exprimée par une logique des conditionnelles dont l'axiomatique est donnée en annexe.
Enfin, on montre comment ce cadre logique permet de comparer différentes définitions qui ont été exprimées en logique modale [1].

**Mots-clés :** Confiance, Logique Modale

**Abstract:**
The concept of trust refers to several concepts which are informally analyzed first in the context of an example, and then from an abstract point of view. An essay of formalization in modal logic is proposed where trust justifications and trust motivations are clearly distinguished. Justifications may be either empirical or analytical. It is shown that in the latter case they refer to trust into specific properties : the ability to reach some state of affairs, the possibility to trigger actions from intentions, norms fulfillment, mutual interest, or interest to fellow.
The formalization makes explicit the relationships between several modalities which are involved in the definitions of these properties. The entailment relationship is formalized in a minimal conditional logic plus some additional axiom schemas which are given in the annex.
Finally, it is shown how this logical frameworks allows to compare some distinct trust definitions that have been expressed in modal logic.

**Keywords:** Trust, Modal Logic

## 1 Introduction

Dans les relations entre agents la confiance joue un rôle très important, en particulier quand un agent compte sur d'autres agents pour réaliser un but, ou quand il doit faire des choix qui dépendent du comportement futur d'autres agents.

Les exemples abondent autant dans le domaine des relations individuelles que dans les relations sociales (l'économie est un exemple d'actualité assez convaincant), ou que dans les relations entre individus et agents institutionnels (par exemple entre une personne et sa banque).

Dans le domaine de l'informatique, ces relations peuvent utiliser comme intermédiaires des agents logiciels qui agissent pour le compte d'individus ou d'agents institutionnels. Par exemple, quand la gestion d'informations sensibles est confiée à des agents logiciels, se pose le problème de la confiance dans la préservation de la confidentialité, de l'intégrité et de la disponibilité d'accès à ces informations.

La multiplicité des travaux et des conférences consacrés à la confiance [2, 9, 6, 12, 16, 13] a montré qu'il s'agit d'une notion complexe dont la définition ne fait pas vraiment consensus. L'objectif de cet article est d'essayer de préciser les différentes notions qui interviennent dans la confiance. Pour cela, après une analyse informelle (section 2), on utilisera la Logique Modale pour leur donner un sens plus précis (section 3), sans chercher à détailler la formalisation jusqu'à un point qui rendrait la compréhension intuitive trop difficile. La dimension temporelle joue un rôle important dans la confiance et elle sera un peu plus analysée dans la section 4. L'analyse de la section 3 montre que la confiance peut se définir en termes de confiance dans des propriétés plus élémentaires qui sont récapitulées dans la section 5. Après avoir situé notre analyse par rapport aux travaux existants (section 6), on résumera en conclusion les principaux résultats et on proposera plusieurs directions pour des travaux futurs.

## 2 Analyse informelle

Avant d'aborder une analyse générale de la confiance nous allons étudier un exemple qui re-

1. Ce travail a bénéficié d'une aide de l'Agence Nationale de la Recherche portant la référence ANR-06-SETI-006.

présente bien les différentes questions qui seront étudiées ensuite d'un point de vue général.

Considérons un conducteur de voiture qui prévoit un voyage par autoroute, et qui envisage qu'une roue vienne à crever. Dans ce cas il devra se garer sur le bas côté et deux problèmes vont se poser : 1) réparer sa roue et 2) ne pas être accidenté par une voiture qui pourrait l'accrocher. Dans cette situation il aura deux buts : que la roue soit réparée (proposition $\phi$), et que la voiture reste dans l'état non accidentée (proposition $\phi'$).

Il se pose alors la question de savoir s'il peut avoir confiance dans le fait que, dans ces circonstances, sa roue sera réparée (proposition $\Diamond\phi$), et dans le fait que pendant tout ce temps il ne sera pas accidenté (proposition $\Box\phi'$). Supposons qu'il donne une réponse positive à ces deux questions.

Dans ce cas (F1), le conducteur (on l'appelle $i$) croit que si la roue est crevée (proposition $\neg\phi$) et que son but est qu'elle soit réparée (proposition $Goal_i\Diamond\phi$), alors la roue sera réparée.

D'autre part (M1), il croit que si la voiture n'est pas accidentée (proposition $\phi'$) et que son but est qu'elle reste non accidentée (proposition $Goal_i\Box\phi'$), alors elle restera non accidentée.

C'est le fait d'envisager la crevaison qui a **motivé** la question d'avoir, ou non, confiance. Une autre question est de savoir si cette confiance est **justifiée**.

Les justifications de ses croyances peuvent être de différentes natures :

- **Empiriques**. Dans les justifications empiriques on peut encore distinguer deux cas :
  - **Observations**. Par exemple, il est déjà arrivé plusieurs fois au conducteur de crever, et à chaque fois la roue a été réparée et il n'a pas été accidenté.
  - **Réputation**. Par exemple, le conducteur a entendu dire que tout le monde dit que si on crève on peut réparer sans être accidenté.
- **Analytiques**. Par exemple, le conducteur croit (F2) que dans ces circonstances il y aura quelqu'un d'autre (appelons-le $j$) qui est capable de réparer la roue et qui essaiera de réparer la roue. Et il croit aussi (M2) qu'aucun conducteur capable de l'accrocher n'essaiera de l'accrocher.

Ces justifications analytiques peuvent être considérées comme d'autres formes de confiance, qui elles mêmes peuvent avoir des justifications empiriques ou analytiques. Dans le cas des justifications analytiques on peut raffiner l'analyse de la façon suivante.

La confiance exprimée par la croyance (F2) peut être justifiée par le fait (F3) que $i$ croit que dans ces circonstances il y aura quelqu'un ayant l'intention de réparer sa roue qui transforme ses intentions en actions, et qui est capable de la réparer.

La confiance exprimée par la croyance (M2) peut être justifiée par le fait (M3) que $i$ croit que, dans ces circonstances, aucun conducteur capable de causer l'accident et qui transforme ses intentions en action n'aura l'intention de causer l'accident.

La confiance exprimée par la croyance (F3) peut être justifiée par le fait (F4.1) que $i$ croit que dans ces circonstances il y aura quelqu'un, $j$, qui croit qu'il doit réparer la roue de $i$, qui respecte les obligations (c'est-à-dire qui adopte l'intention de faire ce qu'il croit qu'il est obligatoire de faire), qui transforme ses intentions en actions, et qui est capable de réparer la roue. Dans ce cas la confiance de $i$ se fonde, entre autres, sur le fait que $j$ respecte les normes. On dira par la suite qu'un agent est "*obéissant*" s'il adopte l'intention de faire que l'on ait $\phi$ quand il croit qu'il est obligatoire qu'il adopte l'intention de faire que l'on ait $\phi$.

Bien que cette obligation ne soit pas dans le code de la route, il se peut que $i$ croit que $j$ croit que c'est une obligation morale.

Une autre éventualité est que la croyance (F3) est justifiée par le fait que (F4.2) $i$ croit que dans ces circonstances il y aura quelqu'un $j$ qui, en échange de la promesse d'une contrepartie, adoptera l'intention de réparer la roue. Cette personne $j$ peut-être, par exemple, un dépanneur auquel $i$ fait appel, et vis-à-vis duquel $i$ s'engage à payer l'intervention à condition que $j$ fasse la réparation, et $j$ s'engage à faire la réparation à condition que $i$ s'engage à payer l'intervention. Dans ce cas la confiance de $i$ est fondée sur l'intérêt mutuel de $i$ et de $j$.

Enfin, la confiance exprimée par la croyance (F3) peut être justifiée par le fait (F4.3) que $i$ croit que dans ces circonstances il y aura quelqu'un, $j$, qui croit que $i$ a pour but que sa roue soit réparée, et que cela suffit à $j$ pour adopter l'intention de réparer la roue de $i$. Il se peut que $j$ ait connaissance du but de $i$, soit parce que

$i$ fait appel à lui par un geste quelconque, soit parce que $j$ observe que $i$ est en train d'essayer de réparer la roue. Dans ce cas la confiance de $i$ se fonde sur le fait que l'attitude de $j$ est déterminée uniquement par l'intérêt qu'il porte à l'autre ($i$ en l'occurrence).

La confiance exprimée par la croyance (M3) peut être justifiée par le fait (M4.1) que $i$ croit que dans ces circonstances il n'y aura aucun agent $j$ qui croit qu'il est permis d'avoir l'intention de causer l'accident, qui ne fait que ce qui est permis (c'est-à'dire qui n'adopte pas l'intention de faire quelque chose quand il ne croit pas que cela est permis), qui transforme ses intentions en actions, et qui est capable de causer l'accident. On dira par la suite qu'un agent est "*honnête*" s'il n'adopte pas l'intention de faire que l'on ait $\phi$ quand il ne croit pas qu'il est permis de faire que l'on ait $\phi$.

La confiance exprimée par la croyance (M3) peut aussi être justifiée par le fait (M4.2) que $i$ croit que, dans ces circonstances, il n'y aura aucun agent $j$ qui croit qu'en demandant à un autre agent $k$ de causer l'accident moyennant une contre partie, cet agent $k$ s'engagera à causer l'accident à condition que $j$ s'engage à la contre partie, et $j$ s'engagera à la contre partie si $k$ s'engage à causer l'accident. Ici, la confiance de $i$ est fondée sur la croyance qu'aucun couple d'agents $j$ et $k$ n'a pour intérêt mutuel que l'accident ait lieu.

Enfin, la confiance exprimée par la croyance (M3) peut être justifiée par le fait (M4.3) que $i$ croit que dans ces circonstances il n'y aura aucun agent $j$ qui croit possible que $i$ ait pour but que l'accident ait lieu, et que cela suffit à $j$ pour adopter l'intention de causer l'accident, et qui transforme ses intention en action, et qui est capable de causer l'accident. Dans ce cas, comme pour (F4.3), la confiance de $i$ se fonde sur le fait que l'attitude de $j$ est déterminée uniquement par l'intérêt qu'il porte à l'autre.

Nous allons maintenant donner une présentation générale, mais informelle, des différentes sortes de confiance susceptibles d'être justifiées par d'autres formes de confiance. On adoptera dans cette présentation, un langage abrégé et semi-formel. On distinguera deux grands cas : celui où le but de $i$ est de la forme "faire que l'on ait $\phi$", et celui où il est de la forme "maintenir le fait qu'on ait $\phi$".

**Faire que l'on ait $\phi$**

(F1) L'agent $i$ croit que si $\neg\phi$ et $Goal_i\Diamond\phi$, alors dans le futur on aura $\phi$.

Dans le futur on aura $\phi$ s'il existe un agent $j$ tel que ($j$ est capable de faire en sorte que l'on ait $\phi$) ET ($j$ essaie de faire que l'on ait $\phi$).

(F2) L'agent $i$ croit que si $\neg\phi$ et $Goal_i\Diamond\phi$, alors il existe un agent $j$ tel que ($j$ est capable de faire en sorte que l'on ait $\phi$) ET ($j$ essaie de faire que l'on ait $\phi$).

$j$ essaie de faire que l'on ait $\phi$ si ($j$ essaie de faire que l'on ait $\phi$ quand il en a l'intention) ET ($j$ a l'intention de faire que l'on ait $\phi$).

(F3) L'agent $i$ croit que si $\neg\phi$ et $Goal_i\Diamond\phi$, alors il existe un agent $j$ tel que ($j$ essaie de faire que l'on ait $\phi$ quand il en a l'intention) ET ($j$ est capable de faire en sorte que l'on ait $\phi$) ET ($j$ a l'intention de faire que l'on ait $\phi$).

$j$ a l'intention de faire que l'on ait $\phi$ si :

- **Respect des normes** : $j$ croit qu'il est obligatoire que $j$ ait l'intention de faire que l'on ait $\phi$.
  Il est obligatoire que $j$ ait l'intention de faire que l'on ait $\phi$ si ($i$ demande à $j$ d'avoir l'intention de faire que l'on ait $\phi$) ET ($i$ a le pouvoir institutionnel de faire qu'il soit obligatoire que $j$ ait l'intention de faire que l'on ait $\phi$ en demandant à $j$ d'avoir l'intention de faire que l'on ait $\phi$).
- **Intérêt mutuel** : $j$ croit que ($i$ demande à $j$ de s'engager à faire que l'on ait $\phi$ à condition que $i$ s'engage à faire que l'on ait $\psi$) ET ($j$ s'engage à faire que l'on ait $\phi$ à condition que $i$ s'engage à faire que l'on ait $\psi$) ET ($i$ s'engage à faire que l'on ait $\psi$ à condition que $j$ s'engage à faire que l'on ait $\phi$).
- **Intérêt pour l'autre** : $j$ croit que $\neg\phi$ et $Goal_i\Diamond\phi$.
  $j$ croit que $Goal_i\Diamond\phi$ si :
  - $i$ demande à $j$ de faire que l'on ait $\phi$, ou
  - $j$ observe que ($\neg\phi$ ET $i$ essaie de faire que l'on ait $\phi$).

Pour illustrer ces trois possibilités on peut considérer le cas d'une personne $i$ dont le but est d'être transportée en voiture à une certaine destination. Si $i$ est un autostopeur, il a confiance dans le fait qu'en levant le bras il y aura un agent $j$, qui agit "pour l'intérêt de l'autre", qui acceptera de le prendre sans contrepartie. Par contre, si $i$ cherche un taxi, il a confiance dans le fait qu'en levant le bras un chauffeur de taxi $j$ s'arrêtera pour le transporter moyennant un enga-

gement mutuel. On notera que cela présuppose en plus que $i$ et $j$ aient confiance dans le fait que leurs engagements respectifs seront respectés. Ce qui ramène au cas "respect des normes". Dans un contexte similaire, si $i$ est un agent de police, il a le pouvoir institutionnel, en levant le bras, de créer l'obligation pour $j$ de s'arrêter, et il a confiance dans le fait que $j$ respectera l'obligation.

Les protocoles qui peuvent conduire à un engagement mutuel de $i$ et $j$ peuvent être très variés, celui qui a été présenté plus haut n'est donné qu'à titre d'exemple. Dans d'autres contextes il peut être fait appel à une tierce partie de confiance, comme un notaire. Le protocole peut même faire appel à deux autres agents qui représentent $i$ et $j$ et qui ont confiance l'un dans l'autre, comme dans le protocole de la lettre de crédit (voir [1]) utilisé pour le commerce maritime.

**Maintenir le fait qu'on ait** $\phi$

(M1) L'agent $i$ croit que si $\phi$ et $Goal_i\Box\phi$, alors dans le futur on aura toujours $\phi$.

L'agent $i$ croit que dans le futur on aura toujours $\phi$ si aucun agent $j$ capable de faire que l'on ait $\Diamond\neg\phi$ n'essaie de faire que l'on ait $\Diamond\neg\phi$.

(M2) L'agent $i$ croit que si $\phi$ et $Goal_i\Box\phi$, alors il n'existe aucun agent $j$ tel que ($j$ est capable de faire que l'on ait $\neg\phi$) ET ($j$ essaie de faire que l'on ait $\neg\phi$).

L'agent $i$ croit que dans le futur on aura toujours $\phi$ si aucun agent $j$ tel que ($j$ essaie de faire que l'on ait $\neg\phi$ quand il en a l'intention) et ($j$ est capable de faire que l'on ait $\neg\phi$) n'a l'intention de faire que $\Diamond\neg\phi$.

(M3) L'agent $i$ croit que si $\phi$ et $Goal_i\Box\phi$, alors il n'existe aucun agent $j$ tel que ($j$ essaie de faire que l'on ait $\neg\phi$ quand il en a l'intention) ET ($j$ est capable de que l'on ait $\neg\phi$) ET ($j$ a l'intention de que l'on ait $\neg\phi$).

$j$ n'a pas l'intention de faire $\neg\phi$ si :

- **Respect des normes** : $j$ ne croit pas qu'il soit permis que $j$ ait l'intention de que l'on ait $\neg\phi$.
- **Intérêt mutuel** : il est faux que $j$ croit qu'il existe $k$ tel que : ($k$ demande à $j$ de s'engager à faire que $\neg\phi$ à condition que $k$ s'engage à que l'on ait $\psi$) ET ($j$ s'engage à que l'on ait $\neg\phi$ à condition que $k$ s'engage à que l'on ait $\psi$) ET ($k$ s'engage à que l'on ait $\psi$ à condition que $j$ s'engage à que l'on ait $\neg\phi$)
- **Intérêt pour l'autre** : $j$ ne croit pas que $\phi$ et $Goal_i\Diamond\neg\phi$.
  $j$ ne croit pas que $Goal_i\Diamond\neg\phi$ si :
  - $i$ ne demande pas à $j$ de que l'on ait $\neg\phi$, ou
  - $j$ n'observe pas que ($\phi$ ET $i$ essaie de que l'on ait $\neg\phi$).

On notera que, dans le cas "intérêt mutuel", $j$ ne croit pas que quelqu'un ($k$) le "payera" pour faire échouer le but de $i$. Bien que formellement $k$ puisse être $i$ lui-même, il n'est pas vraisemblable que $j$ croie que $i$ puisse "payer" $j$ pour qu'il mette en échec son propre but.

## 3 Essai de formalisation

Nous allons introduire certains opérateurs modaux pour décrire les différentes attitudes mentales que nous venons de voir. La signification de ces opérateurs sera donnée de façon purement intuitive, car ce qui nous intéresse en premier est de montrer que les différentes formes de confiance s'expriment par des relations entre les notions représentées par ces opérateurs.

Le seul opérateur dont l'axiomatique sera partiellement explicitée est l'opérateur qui exprime le fait que "si on a $\phi$, alors on a $\psi$", ou "$\phi$ entraîne $\psi$". On sait que si on utilise l'implication matérielle pour représenter la relation entre $\phi$ et $\psi$ on obtient des conséquences paradoxales qui tiennent au fait que l'implication matérielle est vraie quand l'antécédent est faux. Quand elle est dans le champ d'un opérateur modal on peut éviter ce type de paradoxe, mais c'est au prix de conditions compliquées, qui n'ont rien de naturel, et servent plutôt d'"astuces" pour éviter les paradoxes qu'à exprimer des conditions intuitives. C'est pour ces raisons que nous avons choisi d'exprimer la relation entre antécédent et conséquent avec un opérateur de "conditionnelle" (voir "minimal conditional logic" dans [3] section 10.1) auquel on n'impose que les propriétés dont on a besoin ici. On peut lire intuitivement ces opérateurs de la façon suivante.

$\phi \Rightarrow \psi$ : $\phi$ entraîne $\psi$.
$Bel_i\phi$ : $i$ croit que $\phi$.
$Goal_i\phi$ : $i$ a pour but que l'on ait $\phi$.
$\Box\phi$ : $\phi$ est vraie maintenant et dans tous les instants futurs.
$\Diamond\phi$ : il y a un instant, maintenant ou dans le futur, où $\phi$ est vraie.
$\Diamond\phi \stackrel{\text{def}}{=} \neg\Box\neg\phi$.
$Attempt_i\phi$ : $i$ essaie de faire en sorte que l'on ait $\phi$.

$Int_i\phi$ : $i$ a l'intention de faire que l'on ait $\phi$.
$Obg\phi$ : il est obligatoire que l'on ait $\phi$.
$Perm\phi$ : il est permis que l'on ait $\phi$.
$Perm\phi \stackrel{\text{def}}{=} \neg Obg\neg\phi$.
$Ask_{i,j}\phi$ : $i$ demande à $j$ de faire que l'on ait $\phi$.
$Commit_i(\phi \mid \psi)$ : $i$ s'est engagé à faire que l'on ait $\phi$ à condition que l'on ait $\psi$.
$Obs_i\phi$ : $i$ observe que l'on a $\phi$.

Nous allons maintenant exprimer les différentes attitudes des agents qui ont été présentées dans la section précédente en utilisant ces modalités. Pour ne pas avoir des formules trop complexes nous n'exprimerons pas tous les détails liés à la dimension temporelle. On verra dans la section suivante comment on peut aller plus loin dans ce sens.

**Faire que l'on ait $\phi$**

(F1) $Bel_i(\neg\phi \wedge Goal_i\Diamond\phi \Rightarrow \Diamond\phi)$

On définit la notion de capacité à faire que l'on ait $\phi$ de la façon suivante.
$Able_j\phi \stackrel{\text{def}}{=} Attempt_j\phi \Rightarrow \Diamond\phi$

(F2) $Bel_i(\neg\phi \wedge Goal_i\Diamond\phi \Rightarrow \exists j(Able_j\phi \wedge Attempt_j\phi))$ [2]

Le fait qu'un agent a la propriété de traduire son intention en action est exprimé par la définition suivante.
$PosActive_j\phi \stackrel{\text{def}}{=} Int_j\phi \Rightarrow Attempt_j\phi$

On notera qu'on a : $(Able_j\phi \wedge Attempt_j\phi) \Rightarrow \Diamond\phi$, d'où :

(F3) $Bel_i(\neg\phi \wedge Goal_i\Diamond\phi \Rightarrow \exists j(PosActive_j\phi \wedge Able_j\phi \wedge Int_j\phi))$

Le fait qu'un agent est obéissant est exprimé par la propriété suivante.
$Obey_j\phi \stackrel{\text{def}}{=} Bel_jObgInt_j\phi \Rightarrow Int_j\phi$
On notera que même si on a : $(Bel_jObgInt_j\phi) \wedge Obey_j\phi$, cela garantit seulement que $j$ a l'intention de faire que l'on ait $\phi$. On a alors :

(F4.1) $Bel_i(\neg\phi \wedge Goal_i\Diamond\phi \Rightarrow \exists j(Obey_j\phi \wedge PosActive_j\phi \wedge Able_j\phi \wedge Bel_jObgInt_j\phi))$

Le fait que, du point de vue de $j$, $i$ a le pouvoir, dans l'institution $s$, de créer l'obligation pour $j$ d'avoir l'intention de faire que l'on ait $\phi$ est défini de la façon suivante. [3]
$InstPower_{i,j}\phi \stackrel{\text{def}}{=} Ask_{i,j}\phi \Rightarrow_s Bel_jObgInt_j\phi$
On a alors :

(F5.1) $Bel_i(\neg\phi \wedge Goal_i\Diamond\phi \Rightarrow \exists j(InstPower_{i,j}\phi \wedge Obey_j\phi \wedge PosActive_j\phi \wedge Able_j\phi \wedge Ask_{i,j}\phi))$

Dans l'axiomatique des conditionnelles mentionnée en annexe on peut montrer facilement qu'on a : (F2) implique (F1), (F3) implique (F1), (F4.1) implique (F1), et (F5.1) implique (F1).

On définit le fait que $i$ et $j$ se sont mutuellement engagés, à la demande de $i$, à faire que l'on ait $\phi$ à condition que l'autre fasse que l'on ait $\psi$ de la façon suivante.
$Contract_{i,j}(\phi, \psi) \stackrel{\text{def}}{=}$
$InstPower_{i,j}(Commit_j(\phi \mid Commit_i\psi)) \wedge$
$Obey_j(Commit_j(\phi \mid Commit_i\psi)) \wedge$
$(Commit_j(\phi \mid Commit_i\psi) \Rightarrow$
$Commit_i(\psi \mid Commit_j\phi)) \wedge$
$(Commit_i(\psi \mid Commit_j\phi) \Rightarrow Int_j\phi)$
On a alors :

(F4.2) $Bel_i(\neg\phi \wedge Goal_i\Diamond\phi \Rightarrow \exists j(Contract_{i,j}(\phi, \psi) \wedge PosActive_j\phi \wedge Able_j\phi \wedge Ask_{i,j}Commit_j(\phi \mid Commit_i\psi))$

On a : (F4.2) implique (F1).

On définit le fait que $j$ est disposé à satisfaire le but de $i$ sans contrepartie de la façon suivante :
$ActAltruis_{j,i}\phi \stackrel{\text{def}}{=}$
$Bel_j(\neg\phi \wedge Goal_i\Diamond\phi) \Rightarrow Int_j\phi$
On a alors :

(F4.3) $Bel_i(\neg\phi \wedge Goal_i\Diamond\phi \Rightarrow \exists j(ActAltruis_{j,i}\phi \wedge PosActive_j\phi \wedge Able_j\phi \wedge Bel_j(\neg\phi \wedge Goal_i\Diamond\phi)))$

On a (F4.3) implique (F1).

**Maintenir le fait qu'on ait $\phi$**

(M1) $Bel_i(\phi \wedge Goal_i\Box\phi \Rightarrow \Box\phi)$

(M2) $Bel_i(\phi \wedge Goal_i\Box\phi \Rightarrow \neg\exists j(Able_j\neg\phi \wedge Attemt_j\neg\phi))$

2. Pour éviter un trop grand nombre de parenthèses on utilise la formule : $\neg\phi \wedge Goal_i\Diamond\phi \Rightarrow \exists j(Able_j\phi \wedge Attempt_j\phi)$, à la place de : $(\neg\phi \wedge Goal_i\Diamond\phi) \Rightarrow (\exists j(Able_j\phi \wedge Attempt_j\phi))$.

3. On peut juger cette définition exagérément simplifiée. On devrait avoir plutôt : $InstPower_{i,j,s}\phi \stackrel{\text{def}}{=} Bel_j(Ask_{i,j,s}\phi \Rightarrow_s Obg_sInt_j\phi)$, c'est-à-dire : $j$ croit que $i$ a le pouvoir, dans l'institution $s$, de créer l'obligation $Obg_sInt_j\phi$ en demandant à $j$ de faire que l'on ait $\phi$. Voir [8] pour les actes de communication qui ont des effets institutionnels.

Si un agent capable de que l'on ait $\Diamond\neg\phi$ essaie de que l'on ait $\Diamond\neg\phi$, alors on a $\Diamond\neg\phi$.

Formellement, on a :
$(Able_j\neg\phi \wedge Attemt_j\neg\phi) \Rightarrow \Diamond\neg\phi$.

Un agent $i$, peut accepter, en plus, que si on a $\phi$ et aucun agent capable de faire que l'on ait $\neg\phi$ n'essaie de faire que l'on ait $\neg\phi$, alors on a toujours $\phi$. Ceci revient à accepter que $\neg\phi$ ne peut résulter que de l'action d'un agent capable de faire que l'on ait $\neg\phi$. Cette croyance de $i$ est exprimée par :
(PERSIST1) $Bel_i(\phi \wedge \neg\exists j(Able_j\neg\phi \wedge Attemt_j\neg\phi) \Rightarrow \Box\phi)$

On a (M2) et (PERSIST1) implique (M1).

Le fait qu'un agent est capable de se retenir de faire $\phi$, c'est-à-dire de ne pas essayer de faire que l'on ait $\phi$ s'il n'en a pas l'intention est défini par :
$RefActive_j\phi \stackrel{\text{def}}{=} \neg Int_j\phi \Rightarrow \neg Attempt_j\phi$
Formellement on a :
$\neg Int_j\neg\phi \wedge RefActive_j\neg\phi \rightarrow \neg Attempt_j\neg\phi$.
On a alors :

(M3) $Bel_i(\phi \wedge Goal_i\Box\phi \Rightarrow \neg\exists j(RefActive_j\neg\phi \wedge Able_j\neg\phi \wedge Int_j\neg\phi))$

Un agent $i$, peut accepter, en plus, que si aucun agent qui peut se retenir d'essayer de que l'on ait $\neg\phi$ et qui est capable de que l'on ait $\neg\phi$ n'a l'intention de que l'on ait $\neg\phi$, alors on a toujours $\phi$. Soit :
(PERSIST2) $Bel_i(\phi \wedge \neg\exists j(RefActive_j\neg\phi \wedge Able_j\neg\phi \wedge Int_j\neg\phi) \Rightarrow \Box\phi)$

Cette hypothèse est risquée car il pourrait y avoir des agents capables de faire $\neg\phi$, et qui essaient de faire que $\neg\phi$ bien qu'ils n'en aient pas l'intention. Dans l'exemple de la section précédente, ce pourrait être le cas d'un conducteur $j$ qui accroche la voiture de $i$ involontairement.

On peut noter que (PERSIST2) est équivalent à :

$Bel_i(\phi \wedge \forall j(RefActive_j\neg\phi \wedge Able_j\neg\phi \rightarrow \neg Int_j\neg\phi) \Rightarrow \Box\phi)$

On a (M3) et (PERSIST2) implique (M1).

Le fait que $j$ est honnête est défini par :
$Honest_j\phi \stackrel{\text{def}}{=} \neg Bel_jPermInt_j\phi \Rightarrow \neg Int_j\phi$
On a alors :

(M4.1) $Bel_i(\phi \wedge Goal_i\Box\phi \Rightarrow \neg\exists j(Honest_j\neg\phi \wedge PosActive_j\neg\phi \wedge Able_j\neg\phi \wedge \neg Bel_jPermInt_j\neg\phi))$

Formellement on a :
$\neg Bel_jPermInt_j\neg\phi \wedge Honest_j\neg\phi \rightarrow \neg Int_j\neg\phi$.

Un agent $i$, peut accepter, en plus, que si aucun agent qui est honnête pour $\neg\phi$, qui peut se retenir d'essayer de que l'on ait $\neg\phi$ et qui est capable de que l'on ait $\neg\phi$, ne croit qu'il est permis d'avoir l'intention de faire que $\neg\phi$, alors on a toujours $\phi$. Soit :
(PERSIST3.1) $Bel_i(\phi \wedge \neg\exists j(Honest_j\neg\phi \wedge PosActive_j\neg\phi \wedge Able_j\neg\phi \wedge \neg Bel_jPermInt_j\neg\phi) \Rightarrow \Box\phi)$

Cette hypothèse est risquée car il peut y avoir des agents qui peuvent se retenir d'essayer de que l'on ait $\neg\phi$ et qui sont capables de que l'on ait $\neg\phi$, mais qui ne sont pas honnêtes.

On a (M4.1) et (PERSIST3.1) implique (M1).

(M4.2) $Bel_i(\phi \wedge Goal_i\Box\phi \Rightarrow \neg\exists j(Bel_j(\exists k((Contract_{k,j}(\neg\phi,\psi) \wedge PosActive_j\neg\phi \wedge Able_j\neg\phi \wedge Ask_{k,j}Commit_j(\neg\phi \mid Commit_j\psi))))))$

Un agent $i$ peut accepter en plus une hypothèse similaire à (PERSIST3.1). Soit :
(PERSIST3.2) $Bel_i(\phi \wedge \neg\exists j(Bel_j(\exists k((Contract_{k,j}(\neg\phi,\psi) \wedge PosActive_j\neg\phi \wedge Able_j\neg\phi \wedge Ask_{k,j}Commit_j(\neg\phi \mid Commit_j\psi))))) \Rightarrow \Box\phi)$

On a (M4.2) et (PERSIST3.2) implique (M1).

Le fait qu'un agent $j$ se retient d'adopter l'intention de faire que l'on ait $\phi$ s'il croit qu'il est possible que le but de $i$ est qu'il n'adopte pas l'intention de faire que l'on ait $\phi$ est défini par :
$RefAltruis_{i,j}\phi \stackrel{\text{def}}{=} \neg Bel_j\neg Goal_i\neg Int_j\phi \Rightarrow \neg Int_j\phi$
On a alors :

(M4.3) $Bel_i(\phi \wedge Goal_i\Box\phi \Rightarrow \neg\exists j(RefAltruis_{i,j}\neg\phi \wedge PosActive_j\neg\phi \wedge Able_j\neg\phi \wedge \neg Bel_j\neg Goal_i\neg Int_j\neg\phi))$

Un agent $i$ peut en plus adopter une hypothèse similaire à à (PERSIST3.1). Soit :
(PERSIST3.3) $Bel_i(\phi \wedge \neg\exists j(RefAltruis_{i,j}\neg\phi \wedge PosActive_j\neg\phi \wedge Able_j\neg\phi \wedge \neg Bel_j\neg Goal_i\neg Int_j\neg\phi) \Rightarrow \Box\phi)$

On a (M4.3) et (PERSIST3.3) implique (M1).

La formalisation qui a été présentée jusqu'ici pourrait être raffinée dans plusieurs directions. D'abord en ce qui concerne les quantificateurs qui apparaissent dans les formules telles que (F2) et (M2). Si on s'inspire des logiques modales du premier ordre, la position des quantificateurs par rapport à la modalité $Bel_i$ permet de distinguer le fait que $i$ connaît, ou ne connaît pas, les agents $j$ qui son capables de faire que l'on ait $\phi$ ou de que l'on ait $\neg\phi$.

Par exemple, la formule (F'2) :
(F2) $\exists j Bel_i(\neg\phi \wedge Goal_i\Diamond\phi \Rightarrow (Able_j\phi \wedge Attempt_j\phi))$
exprime que $i$ connaît un $j$ qui satisfait la propriété dans le champs de $Bel_i$, alors que ce n'est pas nécessairement le cas dans (F2).

Le cas de (M2) est plus délicat car il contient implicitement un quantificateur universel. En effet (M2) est équivalent à (M'2) :
(M'2) $Bel_i(\phi \wedge Goal_i\Box\phi \Rightarrow \forall j(Able_j\neg\phi \rightarrow \neg Attemt_j\neg\phi))$
La même transformation que celle que nous avons faite pour passer de (F2) à (M2) conduirait à (M"2) :
(M"2) $\forall j Bel_i(\phi \wedge Goal_i\Box\phi \Rightarrow (Able_j\neg\phi \rightarrow Attemt_j\neg\phi))$
Mais (M"2) est très problématique car elle suppose implicitement que $i$ connaît l'ensemble $I$ de tous les agents capables de faire que l'on ait $\neg\phi$ [4], et il y a de nombreux domaines d'application où ce n'est pas le cas. Par exemple, si $i$ a confié à certains agents logiciels la gestion d'informations confidentielles, il sait que ce sont ces agents qui ont la capacité de communiquer ces informations, et il peut avoir confiance, ou pas, dans le fait qu'aucun d'eux ne les communiquera à des personnes non autorisées. Par contre, si $i$ transmet ses informations via un réseau tel qu'Internet, il n'est pas très réaliste d'envisager que $i$ connaisse tous les agents qui ont la capacité d'altérer l'intégrité de ces informations.

Les différentes modalités que nous avons utilisées ont fait l'objet de nombreux travaux de formalisation, et si certaines, comme la modalité épistémique $Bel_i$ font l'objet de consensus, ce n'est pas le cas pour les modalités de but et d'intention $Goal_i$ et $Int_j$, et encore moins pour les modalités déontiques $Obg$ et $Perm$, ainsi que pour la modalités d'action $Attempt_j$.

Dans notre esprit il est essentiel dans le cadre de la confiance que la modalité $Attempt_j$ exprime la causalité, c'est-à-dire que si après que $j$ ait essayé de faire que l'on ait $\phi$ on a $\phi$, alors on a $\phi$ à cause de ce qu'a fait $j$. Même s'il est bien connu que la formalisation de la causalité pose de gros problèmes, en particulier à cause de la condition contre factuelle, les travaux de G. H. von Wright [18], de I. Pörn [17] ou de N. Belnap et J. Horty [10], peuvent servir de référence.

Malheureusement l'expression "essayer de faire que" laisse entendre que $j$ agit intentionnellement. Ce n'est pas le sens que nous voulons donner à $Attempt_j\phi$, et **on ne suppose pas** que : $Attempt_j\phi \Rightarrow Int_j\phi$. Par exemple, si le conducteur $i$ a confiance dans le fait qu'aucun autre conducteur $j$ ne va l'accrocher, l'ensemble $I$ mentionné plus haut contient non seulement les agents qui pourraient l'accrocher intentionnellement, mais aussi ceux qui pourraient le faire involontairement.

On notera que parmi les propriétés élémentaires qui peuvent faire l'objet de confiance ne figure pas le pouvoir institutionnel. En effet, la conditionnelle qui définit ce pouvoir exprime une norme, définie par une institution, qui relie l'antécédent et le conséquent [11]. Cette relation n'exprime pas la régularité de l'attitude d'un agent, et elle est de nature différente.

Enfin, faut-il envisager un cadre logique unique qui intégrerait une formalisation détaillée de toutes les modalités que nous utilisons. Mis à part le fait qu'un travail d'une telle ampleur constitue un défi considérable, le résultat serait d'une telle complexité que le bénéfice en terme de compréhension intuitive est loin d'être acquis. De plus, on pourrait toujours trouver des points insuffisamment formalisés, et comme le montre très bien J. Ladrière dans [14] il faut accepter des limites à la formalisation. Ici, nous nous sommes limité au niveau qui permet de comparer différents points de vue sur la formalisation de la confiance.

## 4 Dimension temporelle

Jusqu'à présent nous n'avons fait intervenir le temps que pour distinguer le cas où le but d'un agent $i$ est qu'advienne une situation où on a $\phi$, représenté par $\neg\phi \wedge \Diamond\phi$, et le cas où son but est

4. La formule (M"2) soulève aussi la question de savoir si elle est "domain independent", au sens défini dans [4], car, à notre connaissance, il n'y a aucune étude portant sur les formules "domain independent" en logique modale, et il faudrait s'assurer que la sous-formule $Able_j\neg\phi$ peut être considérée comme un prédicat qui définit le champ du quantificateur universel $\forall j$. La même question se pose pour le quantificateur existentiel $\exists j$ dans (F2).

que persiste une situation où on a $\phi$, représenté par $\phi \wedge \Box\phi$.

Il y a une autre distinction qui vaut la peine d'être analysée, car elle a conduit à des définitions différentes de la confiance, et peut-être à des confusions, c'est la distinction entre, d'une part, le cas où survient un événement, par exemple une roue qui crève, qui motive l'adoption à cet instant-là d'un but, et, éventuellement, l'adoption de la confiance dans le fait que le but sera atteint, et, d'autre part, le cas où, comme on l'a décrit dans la section 2, un agent envisage qu'à n'importe quel instant futur pourrait survenir le même événement. Dans ce cas c'est à l'instant présent qu'il doit décider, ou non, d'adopter la confiance dans le fait que son but sera satisfait quel que soit l'instant futur où l'événement aura lieu.

Nous allons voir comment cette distinction s'exprime en logique modale.

On utilisera les notations suivantes.

$Until(\phi)\psi$ : $\psi$ est vraie jusqu'à ce que $\phi$ soit vraie.
$Before(\phi)\psi$ : $\psi$ sera vraie à un certain instant avant que $\phi$ soit vraie.

**But motivé par un événement présent**

**Faire que l'on ait $\phi$**

En pratique $i$ veut que son but soit satisfait dans un certain délai, c'est-à-dire avant qu'une proposition représentée par $\delta$ soit vraie. Par exemple, $i$ veut que la roue soit réparée avant qu'il fasse nuit. D'autre part, le but de $i$ ne peut pas être satisfait à l'instant même où il a été adopté. Donc le but de $i$ persiste jusqu'à ce qu'il soit satisfait ou que le délai soit dépassé.

Si de plus $i$ adopte, à l'instant présent, la croyance dans le fait que dans toute situation telle que la situation présente le but sera satisfait avant $\delta$, on peut dire qu'il a confiance dans le fait que son but sera réalisé. Formellement la situation est représentée par :

$Bel_i(\neg\phi \wedge Until(\phi \vee \delta)Goal_i Before(\delta)\phi \Rightarrow Before(\delta)\phi)$

On notera que dans ce cas la relation entre le but et le fait qu'il sera satisfait s'exprime par une conditionnelle bien que la croyance de $i$ ne soit motivée que par l'événement présent. En effet, si $i$ adopte cette croyance c'est parce qu'il croit qu'il y a une relation de dépendance entre son but et le fait qu'il soit satisfait, et il n'est pas nécessaire qu'il croit que cette relation persiste au delà de l'instant où son but actuel est satisfait.

Ce serait une erreur de représenter la relation entre le but et sa satisfaction par une conjonction, bien qu'à partir de la confiance $i$ puisse **déduire** qu'il a le but **et** qu'il sera satisfait [5] .

**Maintenir le fait qu'on ait $\phi$**

Ici aussi, en pratique, le but de $i$ (ne pas être accidenté) n'est pas illimité dans le temps. Par exemple, son but peut être ne pas être accidenté jusqu'à ce qu'il puisse repartir, c'est-à-dire jusqu'à ce que la roue soit réparée. D'autre part, son but de ne pas être accidenté persiste jusqu'à ce qu'il puisse repartir. Si on représente par $\delta$ la limite qu'il accepte pour son but, on a alors dans le cas général :

$Bel_i(\phi \wedge Until(\delta)Goal_i Until(\delta)\phi \Rightarrow Until(\delta)\phi)$

**But motivé par un éventuel événement futur**

**Faire que l'on ait $\phi$**

Dans ce cas $i$ peut adopter le but à n'importe quel instant dans le futur, et il croit que s'il adopte le but, alors le but sera atteint. Cependant, ici aussi, en pratique $i$ n'envisage pas un futur illimité et sa confiance ne peut pas être illimitée dans le temps. Par exemple, il a confiance dans le fait que sa roue sera réparée si elle crève à n'importe quel instant de son voyage jusqu'à ce qu'il soit arrivé à son terme.

Dans le cas général, la confiance de $i$ s'exprime alors de la façon suivante.

$Bel_i(Until(\delta')(\neg\phi \wedge Goal_i Before(\delta)\phi \Rightarrow Before(\delta)\phi))$

**Maintenir le fait qu'on ait $\phi$**

Pour des raisons similaires, ici aussi le but et la confiance de $i$ ont des limites temporelles. Par exemple, $i$ croit que jusqu'à ce qu'il soit arrivé au terme de son voyage ($\delta'$), s'il est garé le long d'un autoroute et que son but est de ne pas se faire accrocher jusqu'à ce qu'il ait réparé ($\delta$), alors il ne sera pas accroché jusqu'à ce qu'il ait réparé.

5. On notera que si la relation entre l'antécédent et le conséquent était représentée par l'implication matérielle, c'est-à-dire par une formule de la forme : $Bel_i(A \rightarrow C)$, on aurait $Bel_i(A \rightarrow C) \wedge Bel_i A$ logiquement équivalent à $Bel_i(A \wedge C)$, alors que ce n'est pas le cas pour $Bel_i(A \Rightarrow C) \wedge Bel_i A$

Dans le cas général, la confiance de $i$ s'exprime alors de la façon suivante.

$Bel_i(Until(\delta')(\phi \wedge Goal_i Until(\delta)\phi \Rightarrow Until(\delta)\phi))$

## 5 Confiance dans les propriétés élémentaires

De l'analyse de la section 3 on peut conclure que la confiance dans le fait qu'un but soit atteint peut être justifiée par la confiance dans un certain nombre de propriétés élémentaires qui sont récapitulées ci-dessous. Ces propriétés s'expriment toutes sous forme de conditionnelles.

Du point de vue des relations temporelles entre antécédents et conséquents nous avons fait certaines hypothèses. On suppose que les relations entre attitudes mentales sont simultanées (relations entre le fait que $j$ croit, ou ne croit pas, quelque chose et le fait que $j$ adopte, ou non, l'intention de faire que l'on ait $\phi$), et on suppose que le passage d'une intention à une action, ou d'une action à l'effet de l'action, nécessite un délai dans le temps. On a alors :

$Able_j\phi \stackrel{\text{def}}{=} Attempt_j\phi \Rightarrow \Diamond\phi$
$PosActive_j\phi \stackrel{\text{def}}{=} Int_j\phi \Rightarrow \Diamond Attempt_j\phi$
$RefActive_j\phi \stackrel{\text{def}}{=} \neg Int_j\phi \Rightarrow \neg\Diamond Attempt_j\phi$
$Obey_j\phi \stackrel{\text{def}}{=} Bel_j ObgInt_j\phi \Rightarrow Int_j\phi$
$ActAltruis_{j,i}\phi \stackrel{\text{def}}{=} Bel_j(\neg\phi \wedge Goal_i\Diamond\phi) \Rightarrow Int_j\phi$
$Honest_j\phi \stackrel{\text{def}}{=} \neg Bel_j PermInt_j\phi \Rightarrow \neg Int_j\phi$
$RefAltruis_{i,j}\phi \stackrel{\text{def}}{=} \neg Bel_j(\phi \wedge Goal_i\Box\phi) \Rightarrow \neg Int_j\phi$

En général, la confiance est une propriété que $i$ attribue à $j$ pour les instants futurs. Mais, comme on l'a vu, en pratique cette confiance n'est pas illimitée dans le temps. De plus, en général, elle est restreinte à certains contextes, et ceci de deux façons différentes. D'une part, ce n'est que dans certains contextes que $i$ accordera sa confiance à $j$ pour une certaine durée dans le temps, et d'autre part, à un instant futur donné, la propriété attribuée à $j$ peut être restreinte aux cas où certaines conditions sont satisfaites à cet instant-là. Si on appelle $Prop_{i,j}\phi$ la propriété attribuée par $i$ à $j$, on a la forme générale de la confiance représentée par [6] :

$TrustProp_{i,j}\phi \stackrel{\text{def}}{=} Bel_i(BelContext \Rightarrow Until(\delta)(PropContext \Rightarrow Prop_{i,j}\phi))$

Par exemple, dans le cas de la confiance dans la capacité ou dans l'honnêteté de $j$ on a :

$TrustAble_{i,j}\phi \stackrel{\text{def}}{=} Bel_i(BelAContext \Rightarrow Until(\delta)(AContext \Rightarrow (Attempt_j\phi \Rightarrow \Diamond\phi)))$

$TrustHonest_{i,j}\phi \stackrel{\text{def}}{=} Bel_i(BelHContext \Rightarrow Until(\delta)(HContext \Rightarrow (\neg Bel_j PermInt_j\phi \Rightarrow \neg Int_j\phi)))$

## 6 Comparaison avec d'autres travaux

Nous nous restreindrons ici aux travaux qui proposent des formalisations en logique modale. D'une manière générale ils s'accordent sur le fait que la confiance est une attitude mentale qui s'exprime par une croyance d'un agent $i$ au sujet de propriétés attribuées à un autre agent $j$ [2, 9, 12, 6].

Ces propriétés peuvent-elles être étendues à des objets, ou à des systèmes physiques ? Par exemple, il se peut que, dans le cas où $i$ est un conducteur, $i$ déclare qu'il a confiance dans le fait que son régulateur automatique de vitesse se débranchera s'il appuie sur le bouton approprié. Nous pensons qu'en fait cette confiance ne concerne pas les systèmes physiques eux-mêmes. En effet, leurs comportements sont déterminés, d'une part, par les lois de la nature, et celles-ci ne peuvent faire l'objet d'un jugement subjectif tel que la confiance, et d'autre part, par les personnes qui les ont conçus, fabriqués ou utilisés. Ce sont donc, plus vraisemblablement, ces personnes qui peuvent faire, ou pas, l'objet de la confiance de $i$. On est donc ramené au cas précédent.

Les divergences les plus importantes portent sur les notions qui doivent être considérées comme constitutives de la confiance. En particulier, pour C. Castelfranchi et al. [2, 9] le but de $i$ est un élément constitutif de la confiance. Selon cette approche, la croyance de $i$ dans certaines propriétés de $j$ ne peut être considérée comme de la confiance que si cette croyance est motivée par le fait que $i$ a un certain but, et que celui-ci

6. Une notation, plus lourde, mais plus rigoureuse devrait être de la forme : $TrustProp_{i,j}(\phi, BelContext, PropContext, \delta)$.

pourrait être satisfait par $j$ en mettant en oeuvre cette propriété. Cette approche peut être formalisée en logique en exprimant que si $i$ croit que $j$ a cette propriété, alors nécessairement $i$ a un but qui peut être atteint grace à $j$. Cette contrainte peut s'exprimer, sans détailler tous les aspects temporels, par :

$TrustProp_{i,j}\phi \rightarrow Bel_i \Diamond Goal_i \phi$

On notera qu'ici la relation entre l'antécédent est le conséquent est exprimée par l'implication matérielle, et non par une conditionnelle, car la contrainte exprime une disjonction : ou bien on n'a pas $TrustProp_{i,j}\phi$, ou bien on a $Bel_i \Diamond Goal_i \phi$.

Pour d'autre auteurs, en particulier pour A.J.I. Jones [12], le but n'est pas considéré comme une condition nécessaire pour qu'on puisse appeler "confiance" certaines croyances, même s'il accepte que très souvent c'est le cas. Il considère que c'est essentiellement la régularité de l'attitude de $j$ qui constitue la notion de confiance. Il distingue deux types de régularité : la capacité à créer une certaine situation, et le respect de certaines obligations. Il exprime la première par la formule : $Bel_i(\psi \rightarrow E_j\phi)$, et la seconde par : $Bel_i(ObgE_j\phi \rightarrow E_j\phi)$ [7] , où $E_j\phi$ signifie que $j$ a fait en sorte qu'on ait (est la cause de) $\phi$. Bien que ce ne soit pas explicité dans les formules, pour Jones les propriétés attribuées à $j$ ne sont pas satisfaites qu'à l'instant présent, mais toujours dans le futur.

Une autre différence importante porte sur le fait que dans l'approche de Castelfranchi et al. $i$ croit que $j$ satisfera son but en réalisant une action bien déterminée $\alpha$, tandis que pour Jones cette action n'est pas nécessairement constitutive de la confiance. On peut résumer ces deux positions en disant que dans la première $i$ veut que son but soit atteint d'une certaine manière, alors que dans la seconde ce n'est que l'état à atteindre qui compte. Plus brièvement dans la seconde le but n'exprime que le "quoi", et la première exprime en plus le "comment". On peut formellement concilier ces deux approches en explicitant dans la formule $\phi$ qui définit le but, le fait que $j$ a réalisé l'action $\alpha$. Le but serait alors, par exemple, de la forme : $\phi \wedge Done_j\alpha$. Il resterait à préciser la modalité $Done_j\alpha$.

Par exemple, dans le cas du conducteur dont la roue est crevée, il se peut que $i$ veuille que la roue soit réparée sur place, ou, au contraire, que la voiture soit transportée dans un garage pour faire la réparation. A l'inverse, si $i$ approche d'un carrefour où il y a un "stop" pour les voitures qui viennent d'une route perpendiculaire, et que son but est qu'aucune autre voiture ne se trouve au carrefour pendant qu'il passe, il peut avoir confiance dans le fait qu'aucun conducteur $j$ venant de l'autre route n'essaiera de faire qu'il se trouve au milieu du carrefour. Et dans ce cas $i$ ne se préoccupe pas de l'action que pourrait faire $j$ pour se trouver au milieu du carrefour ; par exemple il se pourrait que $j$ soit arrêté et qu'il lui barre la route en faisant l'action de démarrer, ou bien que $j$ soit en train d'approcher du carrefour et qu'il s'abstienne de freiner pour arrêter sa voiture.

## 7 Conclusion

Partant d'une définition de la confiance en termes de croyance dans le fait que le but d'un agent sera satisfait, nous avons montré que ce type de confiance peut être justifié par la confiance dans certaines propriétés attribuées à d'autres agents : la capacité à obtenir un certain état des choses, la possibilité de transformer les intentions en actions, et enfin la possibilité d'adopter certaines intentions. Celle-ci peut être déterminée, soit par le respect des normes, soit par l'intérêt mutuel, soit par l'intérêt pour l'autre. Chacune de ces propriétés a sa duale selon qu'un agent attend d'un autre agent qu'il agisse pour satisfaire son but, ou qu'il s'abstienne d'agir pour l'empêcher de maintenir son but.

Grâce à la formalisation en logique modale on a pu mettre en évidence des distinctions entre des notions comme : obéissance et honnêteté, ou entre celles représentées par : $ActAltruis$ et $RefAltruis$. Ces dernières correspondent aux distinctions qu'ont mis en évidence J. Horty et N. Belnap dans [10] avec les modalités d'action du type "to see to it that" et "to refrain to see to it that".

D'autre part, en explicitant la dimension temporelle, nous avons mis en évidence la distinction entre la confiance qui est motivée par un événement particulier à l'instant présent, et celle qui est motivée par l'occurrence éventuelle de cet événement dans le futur.

Du point de vue de la formalisation, nous avons adopté un opérateur de conditionnelle pour lequel nous avons requis les propriétés les plus

7. Nous avons changé les notations pour les rendre homogènes avec le reste de l'article.

faibles, celles qui suffisent à déduire ce qu'il nous intéressait de déduire dans le cadre de cet article (essentiellement le fait que certains types de confiance permettent de déduire d'autres types de confiance), laissant ainsi la porte ouverte à l'adoption de schémas d'axiomes plus spécifiques si nécessaire. La formalisation de la signification de chaque opérateur modal n'a pas été développée, et nous nous sommes concentrés uniquement sur leurs interrelations.

De nombreux aspects restent à étudier et à formaliser au sujet de la confiance. Nous n'en mentionnerons que trois.

- Il faudrait distinguer des buts de natures différentes. Les buts épistémiques, quand le but de $i$ est de connaître une information (voir [5]), par exemple la situation financière d'une banque, ou une prévision relative à l'évolution du cours d'une action. Les buts "physiques", quand le but est d'atteindre une certaine situation physique de l'état du monde, par exemple qu'une roue soit réparée. Et enfin les buts déontiques, par exemple quand le but de $i$ est de créer une obligation en portant plainte auprès d'un magistrat.
- Dans l'analyse que nous avons faite dans cet article nous n'avons distingué que le cas où $i$ a, ou n'a pas, confiance. Il faudrait pouvoir distinguer des degrés de confiance. Ces nuances sont évidentes quand la justification de la confiance est de nature empirique et s'appuie sur des statistiques. Il reste beaucoup de points à étudier quand la justification n'est pas empirique et qu'on considère des degrés qualitatifs [15, 7].
- Enfin, dans le domaine de l'évolution de la confiance, il nous semble important d'aller au-delà des modifications de probabilités objectives qui peuvent résulter de l'acquisition de nouvelles observations. La confiance est une notion essentiellement subjective qui n'évolue pas toujours de façon rationnelle. Par exemple, un accident de la route très rare, mais spectaculaire, peut avoir un effet très fort sur la confiance même si la probabilité de cet accident reste inchangée. D'autre part, l'évolution de la confiance peut être influencée par des changements de nature normative et qui ne peuvent être pris en compte par des statistiques. Par exemple, le fait de réduire la vitesse maximum autorisée sur autoroute peut augmenter la confiance par rapport au risque d'accidents.

## Annexe

On suppose que l'axiomatique des connecteurs logiques est définie par le Calcul Classique des Propositions.

Pour l'opérateur de conditionnelle on adopte les règles d'inférences et schémas d'axiomes suivants.

(EQUIV) Si $\vdash \phi \leftrightarrow \phi'$ et $\vdash \psi \leftrightarrow \psi'$, alors $\vdash (\phi \Rightarrow \psi) \rightarrow (\phi' \Rightarrow \psi')$
(PROPG) $(\phi_1 \wedge \phi_2 \Rightarrow \phi_3) \rightarrow (\phi_1 \wedge \phi_2 \Rightarrow \phi_1 \wedge \phi_3)$
(TRANS) $(\phi_1 \Rightarrow \phi_2) \wedge (\phi_2 \Rightarrow \phi_3) \rightarrow (\phi_1 \Rightarrow \phi_3)$
(DIST) $(\phi_1 \wedge (\phi_1 \Rightarrow \phi_2) \wedge \psi) \Rightarrow (\phi_2 \wedge \psi)$
(MATIMP) $(\phi \Rightarrow \psi) \rightarrow (\phi \rightarrow \psi)$

Dans cette logique on peut facilement démontrer par induction à partir de (TRANS) et (DIST) que l'on a :

$\phi_1 \wedge (\phi_1 \Rightarrow \phi_2) \wedge (\phi_2 \Rightarrow \phi_3) \wedge \ldots$
$\ldots \wedge (\phi_{n-1} \Rightarrow \phi_n) \Rightarrow \phi_n$

## Références


[1] G. Boella, J. Hulstin, Y-H. Tan, and L. van der Torre. Transaction trust in normative multi agent systems. In *AAMAS Workshop on Trust in Agent Societies*, 2005.

[2] C. Castelfranchi and R. Falcone. Social trust : a cognitive approach. In C. Castelfranchi and Y-H. Tan, editors, *Trust and Deception in Virtual Societies*. Kluwer Academic Publisher, 2001.

[3] B. F. Chellas. *Modal Logic : An introduction*. Cambridge University Press, 1988.

[4] R. Demolombe. Syntactical Characterization of a Subset of Domain Independent Formulas. *Journal of ACM*, 39(1), 1982.

[5] R. Demolombe. To trust information sources : a proposal for a modal logical framework. In C. Castelfranchi and Y-H. Tan, editor, *Proc. of the Workshop on Deception, Fraud and Trust in Agent Societies*, 1998.

[6] R. Demolombe. Reasoning about trust : a formal logical framework. In C. Jensen, S. Poslad, and T. Dimitrakos, editors, *Trust management : Second International Conference iTrust (LNCS 2995)*. Springer Verlag, 2004.

[7] R. Demolombe and C-J. Liau. A logic of graded trust and belief fusion. In C. Castelfranci and R. Falcone, editors, *Proc. of 4th Workshop on Deception, Fraud and Trust*, 2001.

[8] R. Demolombe and V. Louis. Speech acts with institutional effects in agent societies. In L. Goble and J-J. Ch. Meyer, editors, *Deontic Logic and Artificial Normative Systems*. Springer, LNAI 4048, 2006.

[9] R. Falcone and C. Castelfranchi. Trust dynamics : How trust is influenced by direct experiences and by trust itself. In *Proceedings of the 3rd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-04)*, pages 740–747. New York, ACM, 2004.

[10] J.F. Horty and N. Belnap. The deliberative STIT : a study of action, omission, ability, and obligation. *Journal of Philosophical Logic*, 24 :583–644, 1995.

[11] A. J. Jones and M. Sergot. A formal characterisation of institutionalised power. *Journal of the Interest Group in Pure and Applied Logics*, 4(3), 1996.

[12] A.J.I. Jones. On the concept of trust. *Decision Support Systems*, 33, 2002.

[13] A.J.I. Jones and B.S. Firozabadi. On the characterisation of a trusting agent. Aspects of a formal approach. In C. Castelfranchi and Y-H. Tan, editors, *Trust and Deception in Virtual Societies*. Kluwer Academic Publisher, 2001.

[14] J. Ladrière. Les limites de la formalisation. In J. Piaget, editor, *Logique et Connaissance Scientifique*. Editions Gallimard, 1967.

[15] N. Laverny and J. Lang. From knowledge-based programs to graded belief-based programs part ii : Off-line reasoning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2005.

[16] E. Lorini and R. Demolombe. Trust and norms in the context of computer security : a logical formalization. In R. van der Meyden and L. van der Torre, editors, *Deontic Logic in Computer Science*. Springer, LNAI 5076, 2008.

[17] I. Pörn. Action Theory and Social Science. Some Formal Models. *Synthese Library*, 120, 1977.

[18] G. H. von Wright. *Norm and Action*. Routledge and Kegan, 1963.

# Prédire l'intérêt dans la communication événementielle

A. Dimulescu[1] J.-L. Dessalles[1]
{adrian.dimulescu, jean-louis.dessalles}
@telecom-paristech.fr

[1]Telecom ParisTech – LTCI UMR-5141
75013 Paris – FRANCE

**Résumé :**
Cette étude vise à mesurer les variations de l'intérêt suscitées par un événement lorsque certaines dimensions définies sont manipulées. Les résultats sont comparés aux prédictions de la théorie du décalage de complexité, selon laquelle les événements sont d'autant plus intéressants qu'ils sont plus simples qu'attendu.
**Mots-clés** : Complexité, simplicité, intérêt, événement.

**Abstract :**
This study is an attempt to measure the variations of interest aroused by conversational narratives when definite dimensions of the reported events are manipulated. The results are compared with the predictions of the Complexity Drop Theory, which states that events are more interesting when they appear simpler than anticipated.
**Keywords**: Complexity, simplicity, interest, event.

## 1 Introduction

L'activité qui consiste à signaler ou rapporter des événements occupe une part significative du temps humain. Si l'on considère que le temps de conversation représente jusqu'à un tiers du temps d'éveil (Mehl & Pennebaker, 2003) et que les narrations peuvent constituer jusqu'à 40% de ce temps de parole (Eggins & Slade, 1997), nous pouvons passer plus du dixième de notre temps disponible à communiquer des événements dignes d'intérêt, ce qui est considérable. Cette activité présente un enjeu social important, comme l'ont révélé les études sociolinguistiques (p.ex. Labov & Waletzky, 1967; Labov, 1997; Tannen, 1984) : celui ou celle qui est capable d'intéresser son auditoire renforcera de ce fait ses liens sociaux et, inversement, ceux qui suscitent l'ennui en rapportant des événements sans intérêt courent le risque de se retrouver isolés. Malgré son importance, l'activité narrative spontanée n'a été décrite que récemment (voir notamment Norrick, 2000) et, à notre connaissance, n'a pas fait l'objet de modélisation.

L'analyse de la sélection des événements rapportés au cours des interactions spontanées et des paramètres qui contrôlent leur intérêt nous a permis de développer un modèle fondé sur la complexité (Dessalles, 2006, 2008b). Notre objectif ici est de tester cette théorie en analysant les préférences des sujets concernant les variantes de petites histoires.

Dans ce qui suit, nous rappelons les éléments principaux de la théorie du décalage de complexité, puis nous décrivons notre expérience. Nous terminons par une discussion sur la portée des résultats obtenus.

## 2 La théorie du décalage de complexité

Même si les facteurs de l'intérêt narratif des événements n'ont pas été répertoriés, il y a eu des tentatives dans des domaines connexes comme le journalisme ou l'étude de la mémoire. Selon les travaux classiques du journalisme (Galtung et Ruge, 1965), la récence, la proximité, le caractère exceptionnel ou négatif des événements et le nombre d'individus affectés ont une influence décisive sur la valeur de l'information (*newsworthiness*). Un moyen indirect d'évaluer les facteurs qui jouent sur l'intérêt narratif consiste à observer les paramètres qui influent sur la mémorisation des événements. Les individus mémorisent plus efficacement les comportements qui leur paraissent incohérents ou qui s'écartent des stéréotypes (Stangor et McMillan 1992), notamment les actions atypiques (Woll et Graesser 1982 ; Shapiro et Fox 2002). Nous suggérons qu'une seule propriété, l'*inattendu*, résume l'ensemble de ces caractéristiques. L'autre dimension de l'intérêt des événements, l'intensité émotionnelle (Rimé, 2005), ne sera pas considérée ici.

Selon la théorie du décalage de complexité (*Complexity Drop Theory*, CDT), les événements dignes d'intérêt sont ceux qui sont *plus complexes à produire qu'à décrire*. Formellement[1] :

$$U(s) = C_w(s) - C(s)$$

$U(s)$ mesure l'inattendu de la situation $s$. $C(s)$ est sa complexité, c'est-à-dire la taille de la description la plus concise de $s$, autrement dit la quantité d'information minimum permettant de désigner $s$ sans ambigüité (Li & Vitányi, 1993). $C_w(s)$ est la complexité de *production*, mesurée par la quantité d'information minimum permettant de produire $s$ à partir du monde $w$ tel qu'il connu ou imaginé et en tenant compte de ses contraintes.

La notion de complexité apparaît de plus en plus comme fondamentale en sciences cognitives (Chater, 1999 ; Chater & Vitányi, 2003). Par exemple, la reconstitution des parties cachées obéit à un principe de complexité minimale. Il en est de même pour l'induction (Solomonoff, 1978) et pour la formation d'hypothèse (Chaitin, 2004). Le reproche qui est parfois adressé aux théories utilisant la complexité est que cette notion reste abstraite, puisqu'on ne peut pas la calculer en général. Ce reproche est infondé dans de nombreux cas d'intérêt pratique (Delahaye & Zenil, 2007), et particulièrement dans le cas de la complexité cognitive (Dessalles 2008b). La suite du présent article va permettre d'illustrer la fécondité du concept de complexité en testant ses prédictions concernant l'intérêt des événements.

## 3 Description de l'expérience

Pour tester les prédictions de la CDT, nous avons constitué un corpus de 18 histoires en français, pour partie inspirées d'événements rapportés sur un site Web communautaire (viedemerde.fr) sur lequel les utilisateurs décrivent des petits événements de la vie quotidienne dans un style informel. Les histoires ont été proposées à 95 participants par l'intermédiaire d'un questionnaire sur le Web. Notre objectif était de démontrer que, derrière la richesse et la variété phénoménologique apparente de ces histoires, un déterminisme simple est à l'œuvre. Voici un exemple d'histoire, suivi des trois options possibles.

> Les policiers du commissariat d'Antibes ont découvert dans l'après midi dans la baie des Anges les corps flottants de deux femmes aux élégantes tenues vestimentaires. Apparemment les deux noyades se sont passées à peu près en même temps. De plus, les deux femmes [...].
>
> a: avaient chacune un tatouage rouge représentant le dragon Tsuba-Kasai sur le bras droit
> b: avaient chacune un tatouage rouge sur le bras droit
> c: avaient chacune un tatouage sur le bras droit

On ne saurait être trompé par le fait que les réponses attendues pour ce test semblent souvent « évidentes ». L'explication ne l'est pas. Cette situation rappelle celle des études sur la syntaxe, où l'explication scientifique d'une intuition largement partagée se révèle très difficile à dégager.

Toutes les histoires ont été présentées dans un ordre aléatoire à tous les participants. La plupart des participants étaient des ingénieurs et des étudiants travaillant dans les domaines autres que les sciences cognitives. Pour chaque histoire, nous avons isolé un paramètre considéré comme important pour la pertinence de l'histoire et avons défini trois options qui affecteraient graduellement l'intérêt. Chaque participant ne voyait que deux options pour chaque histoire, de sorte que le but global du test ne soit pas transparent. Le test était proposé sous la forme d'une application Web. Les participants devaient compléter un passage manquant pour chaque histoire en cliquant sur une option parmi deux choisies aléatoirement parmi les trois possibles.

À la fin de l'expérience, nous avons donc obtenu pour chaque histoire un ensemble de préférences deux-à-deux entre les trois versions possibles. Pour comparer ces résultats aux prédictions de CDT, il nous fallait agréger les préférences en une hiérarchie. En suivant (Saaty, 1994), nous avons calculé le rang $r_i$ de la version $i$ à partir de sa dominance sur les autres versions, représentée par les poids $w_{ij}$ =

[1] Pour un tutoriel sur CDT, voir le site www.unexpectedness.eu.

win($i$)/win($j$), où win($i$) est le nombre de réponse où la version $i$ a été choisie. Pour ne pas diviser par zéro, un vote a été accordé par défaut à chaque option. Par exemple, si 30 participants ont eu à choisir entre les options $i$ et $j$ d'une histoire donnée et que 23 ont choisi la première version tandis que les 7 participants restants choisissent la seconde, nous avons $w_{ij} = 24/8 = 3$. Notez que $w_{ij} = 1/w_{ji}$. Nous voulons favoriser le rang d'une version d'histoire qui a été préférée aux autres versions, en tenant compte du rang des versions délaissées et de l'amplitude de la dominance. Un calcul possible est $r_i = k \Sigma_j w_{ij} r_j$, que l'on peut réécrire comme $R = kWR$, ce qui revient à trouver un vecteur propre de la matrice $W$ (Farkas 2007). Les rangs sont ensuite normalisés pour que leur somme soit 100. Par exemple, les options de l'histoire précédente sur le double suicide obtiennent les rangs suivants :

| Préférences | | Rangs |
|---|---|---|
| a-b | 17/14 | a 40 |
| b-c* | 34/5 | b 49 |
| a-c* | 23/9 | c 11 |

Pour chaque histoire, les versions sont ordonnées de la plus intéressante à la moins intéressante selon CDT. Les préférences identifiées par * ont été trouvées statistiquement significatives ($p < .05$) sur un test binomial standard visant à détecter une préférence marquée pour une version particulière.

## 4 Analyse

### 4.1 Coïncidences

La théorie CDT permet de prédire la fascination qu'exercent les coïncidences, ainsi que la manière dont cette fascination dépend des paramètres de l'histoire (Dessalles 2008a). Dans une coïncidence comme celle qui est rapportée dans l'histoire du double suicide, l'intérêt résulte de l'observation de deux situations analogues $s_1$ et $s_2$. Nous pouvons écrire $C(s_1\&s_2) \leq C(s_1) + C(s_2|s_1)$, où la complexité conditionnelle $C(s_2|s_1)$ mesure la taille de la description la plus concise de $s_2$ lorsque $s_1$ est disponible. Si les deux situations sont indépendantes, nous pouvons écrire par ailleurs : $C_w(s_1\&s_2) = C_w(s_1) + C_w(s_2)$. Si l'on considère que les deux événements ont la même complexité : $C_w(s_1) \approx C_w(s_2)$ et que, séparément, ils ne sont pas inattendus : $C_w(s_1) = C(s_1)$, alors :

$$U(s_1\&s_2) \geq C(s_1) - C(s_2|s_1)$$

Le modèle CDT prédit ainsi qu'une coïncidence est d'autant plus intéressante que les événements coïncidents sont complexes et qu'ils sont fortement analogues, puisque l'analogie a pour effet de diminuer $C(s_2|s_1)$ (Cornuéjols, 1996). Cette prédiction est globalement vérifiée dans l'histoire suivante.

> J'ai acheté une petite Peugeot 106 ColorLine, que j'ai payée 2000 euros. Je l'avais essayée la veille, elle roulait très bien. Je tourne la clé, je démarre, je sors de la propriété de l'ancien propriétaire de la voiture quand, venant de gauche sans regarder, une autre [...] me rentre dedans.
>
> a: Peugeot 106 ColorLine
> b: Peugeot 106
> c: Peugeot

La contribution de la marque de la voiture à l'inattendu peut faire l'objet d'un calcul précis. Pour estimer $C_w(s_1) = C(s_1)$, on peut considérer que le « choix » de la voiture impliquée dans l'accident demande une complexité $\log_2 N$, où $N$ est le nombre de toutes les voitures de la région. Le calcul de $C(s_2|s_1)$ peut utiliser la marque $f$ de la voiture, qui est disponible gratuitement dans $s_1$. Il suffit alors de $\log_2 n_f$ pour discriminer la voiture impliquée, si $n_f$ est le nombre de voitures de la marque $f$. Ainsi :

$$U(s_1\&s_2) \geq \log_2 N - \log_2 n_f$$

La prédiction est donc que la caractéristique $f$ commune aux deux véhicules augmentera d'autant plus l'intérêt qu'elle sera précise. C'est bien ce qui a été observé.

| Préférences | | Rangs |
|---|---|---|
| a-b* | 24/8 | a 61 |
| b-c* | 32/4 | b 32 |
| a-c* | 28/5 | c 7 |

Dans l'histoire du double suicide mentionnée plus haut, chaque option offre des caractéristiques qui augmentent $C(s_1)$ mais,

puisqu'elles sont communes aux deux situations coïncidentes, ces caractéristiques laissent $C(s_2|s_1)$ inchangée. La prédiction du modèle se trouve globalement vérifiée dans cette histoire, puisque l'option (c) y est largement dominée. Noter toutefois que l'option (a) n'obtient pas le meilleur rang, contrairement à la prédiction de CDT (voir partie discussion).

### 4.2 Déviation quantitative

La CDT prévoit que les situations atypiques sont intéressantes parce qu'il est facile de les singulariser. Nous supposons que la situation atypique est vue comme appartenant à un cadre de référence $r$ et qu'elle s'écarte du prototype de $r$ de $k$ écarts-types pour la caractéristique $f$. Si le « monde » est comparé dans ce cas à une loterie, alors $C_w(s/r) = \log_2 N$, où $N$ est le nombre de situations correspondant à $r$. La complexité pour décrire $s$ à partir de $r$ respecte l'inégalité : $C(s/r) \leq C(f/r) + C(s/r\&f)$. Nous supposerons pour simplifier que $f$ n'est pas conceptuellement liée à $r$ : $C(f/r) = C(f)$. Les situations couvertes par $r$ peuvent être rangées selon les valeurs de $f$ avec une complexité additionnelle négligeable (noter que la complexité se rapporte à la taille des algorithmes, pas à leur temps d'exécution). Si $s$ est extrême pour $f$, elle apparaîtra parmi les premiers éléments de ce tri. Nous pouvons écrire $C(s/r\&f) \approx \log_2 N - A(k)$, où la fonction $A$ dépend seulement de la distribution statistique de $r$ le long de $f$ (pour une distribution gaussienne, $A(k) \sim k^2$). Nous obtenons finalement :

$$U(s/r) \geq A(k) - C(f)$$

L'histoire suivante teste la prédiction.

> La Police d'Amiens a effectué mercredi une saisie [...] d'héroïne au numéro 13 de la rue Fafet.
>
> a: de 10 kilos
> b: de 5 kilos
> c: de 2 kilos

Pour cette histoire, nous devons supposer que les participants ont une idée des saisies typiques de drogue telles qu'elles sont rapportées dans les nouvelles, même si cette idée ne porte que sur l'ordre de grandeur. Il leur suffit de comprendre que les options proposées impliquent des valeurs différentes de $k$. Dans ce cas, la prédiction est qu'ils trouveront l'histoire plus intéressante lorsque la saisie est plus importante.

| Préférences | | Rangs |
|---|---|---|
| a-b* | 28/13 | a 54 |
| b-c | 20/11 | b 28 |
| a-c* | 21/8 | c 18 |

L'expérience montre bien que la hiérarchie (a) > (b) > (c) est respectée.

### 4.3 Déviation qualificative

Certains événements se produisent même si l'on pouvait penser qu'ils étaient presque impossibles. Cette propriété signifie simplement que la complexité $C_w(s)$ nécessaire pour produire un tel événement $s$ est très élevée. Elle s'élève à $C(H)$, où $H$ est le scénario causal le plus simple qui explique $s$ en tenant compte du fonctionnement du monde connu (voir la partie discussion). Selon CDT, l'inattendu vaut :

$$U(s) = C(H) - C(s)$$

Considérons l'histoire suivante.

> Je marchais tranquillement dans la rue quand un parfait inconnu s'arrête devant moi, me regarde et [...] avant de continuer sa route.
>
> a: me donne une gifle phénoménale
> b: me donne une gifle
> c: me demande l'heure

Dans cette histoire, expliquer pourquoi quelqu'un demande l'heure (option (c)) peut être aussi simple que « parce qu'il a oublié sa montre », tandis qu'une explication d'une gifle immotivée exigerait un scénario autrement plus compliqué, et par conséquent une complexité $C(H)$ beaucoup plus élevée. D'autre part, la complexité $C(s)$ peut être calculée en utilisant un cadre de référence $r$ et une caractéristique $f$ de $s$ : $C(s) \leq C(r) + C(f/r) + C(s/r\&f)$. Dans l'histoire précédente, $r$ correspond à une scène typique dans la rue. Pour les options (a) et (b), $f$ correspond au fait d'être giflé par un étranger. Si une telle description peut être considérée comme capturant une situation unique, alors

$C(s|r\&f) = 0$. En revanche, $C(s|r\&f)$ a une valeur significative pour l'option (c) (demander l'heure), puisque les scènes où une personne demande l'heure dans la rue sont nombreuses et ainsi difficiles à distinguer. L'inattendu sera ainsi élevé pour les options (a) et (b) et proche de zéro pour (c).

| Préférences | | Rangs |
|---|---|---|
| a-b | 22/18 | a 53 |
| b-c* | 25/4 | b 41 |
| a-c* | 27/3 | c 6 |

L'expérience montre que, conformément à CDT, l'option (c) est rejetée.

### 4.4 Proximité

Des événements sont plus intéressants s'ils se sont produits à proximité de l'observateur ou à proximité d'un élément déjà connu. La proximité peut être vue comme une relaxation de l'analogie. Dans l'analogie, $C(s_2|s_1)$ est diminuée par la présence d'éléments communs entre $s_1$ et $s_2$. La proximité diminue également la quantité d'information requise pour définir la situation $s_2$ à partir de $s_1$, en utilisant cette dernière comme point de référence à partir duquel il est plus facile de localiser $s_2$. La CDT prévoit ainsi que des événements indépendants se produisant à proximité l'un de l'autre seront ensemble plus intéressants que si leurs occurrences étaient éloignées.

Dans le domaine temporel, si une classe d'événements se produit avec la densité $D_t$, la CDT prévoit que la contribution à l'inattendu d'une occurrence de l'événement observée à distance $t_1$ est[2] :

$$U = -\log_2 (D d)$$

L'intérêt de l'événement décroît comme le logarithme de la distance temporelle. L'histoire qui suit offre un exemple de proximité temporelle inattendue.

> Cela fait un an que je dois changer de portable chez SFR. Je me décide enfin à y aller même s'il faut que je paye une partie car je n'ai pas assez de points carrés rouges. J'ai acheté le portable à 13h00. [...] je reçois un message de SFR : "Changez de mobile, SFR vous offre 15 000 points carrés rouges."
>
> a: A 13h10
> b: A 14h00
> c: Deux semaines plus tard

Dans cette histoire, on s'attend à des offres promotionnelles de la part de l'opérateur environ tous les deux mois. La production qu'une telle offre arrive à la minute précise $t_2$ où elle est arrivée demande une complexité égale à $C_w(t_2) = \log_2 86400 = 16.4$ bits. La description de l'événement peut utiliser le moment $t_1$ de l'achat du mobile comme point de référence. Ainsi, $C(t_2|t_1) = \log_2 10 = 3.3$ bits. Ce contraste contribue à hauteur de 13 bits à l'inattendu (noter que la valeur serait la même avec une résolution temporelle différente). Les trois options de l'histoire précédente produisent un inattendu temporel de 13 bits, 10.5 bits et 2 bits respectivement. Les préférences des participants confirment nettement cet effet.

| Préférences | | Rangs |
|---|---|---|
| a-b* | 29/4 | a 79 |
| b-c* | 28/6 | b 16 |
| a-c* | 31/3 | c 5 |

L'histoire suivante illustre l'influence de la proximité sociale.

> Deux semaines après le vol de ma voiture, la police m'apprend qu'une voiture qui a des chances d'être la mienne est en vente sur Internet. Ils me présentent l'annonce, le numéro de portable est identifié comme étant [...].
>
> a: celui de mon collègue de bureau
> b: celui d'un collègue de mon frère
> c: celui de quelqu'un qui habite le même quartier

L'inattendu, ici, vient du fait que le voleur se trouve être plus simple qu'attendu. Le fait qu'une personne inconnue $P$ soit le voleur a une complexité $C_w(P)$. On peut l'estimer, en comparant la désignation du voleur à une loterie, par le logarithme $\log_2 N$ de la population de la région que le sujet prend comme référence, probablement la ville. Si $P$ se révèle être quelqu'un qui vit dans un voisinage de taille $n$, alors la contribution à l'inattendu est $U = \log_2 N - \log_2 n - C(d)$, où $d$

[2] Voir les détails sur http://www.unexpectedness.eu/NextDoor.html

est le concept de voisinage considéré. Ce calcul extensionnel représente un dernier recours, car un calcul sur le graphe social fournit souvent une plus petite valeur pour $C(P)$. Dans ce cas-là, la complexité de $P$ est l'information minimum requise pour atteindre le nœud de $P$ dans le graphe social. Dans l'histoire du voleur, l'expression « mon collègue » suggère que la complexité de $P$ est zéro une fois que le concept du collègue est installé, car $P$ vient en premier dans la liste. Les résultats confirment clairement la prévision : l'option (a) (mon collègue) est sensiblement préférée à (b) (un collègue de mon frère), et (a) et (b) sont préférées à (c) (quelqu'un vivant dans mon voisinage).

| Préférences | Rangs |
|---|---|
| a-b* 25/6 | a 65 |
| b-c* 24/10 | b 22 |
| a-c* 26/7 | c 13 |

### 4.5 Rencontres fortuites

Les rencontres fortuites fournissent la matière pour de nombreuses conversations. L'inattendu qu'elles produisent s'écrit :

$$U = C(l) - C(P)$$

où $C(l)$ est la complexité du lieu de la rencontre et $C(P)$ celle de la personne rencontrée (Dessalles, 2008a). Il est donc important, pour l'intérêt de l'histoire, que la personne rencontrée soit simple. L'histoire suivante a été proposée aux participants pour tester ce phénomène.

> Je passais dans une rue en plein milieu de Paris quand j'ai entendu quelqu'un m'appeler: c'était un gars [...] quand il était gamin. Nous avons échangé nos adresses. Ca m'a fait super plaisir.
>
> a: que j'ai babysitté pendant 2 ans
> b: que j'ai babysitté pendant 2 mois
> c: que j'ai babysitté quelques soirs

Une période de fréquentation plus longue place l'individu $P$ plus haut dans la liste des personnes que l'on connaît personnellement. La complexité $C(P)$ peut être estimée par le logarithme du rang dans cette liste. Les résultats confirment cette prédiction.

| Préférences | Rangs |
|---|---|
| a-b* 25/11 | a 58 |
| b-c 20/12 | b 26 |
| a-c* 25/7 | c 16 |

Noter que la préférence de (b) à (c) est faible, ce qui s'explique par le fait que les deux options ne modifient pas sensiblement le degré d'accointance passée avec $P$.

### 4.6 Structure

L'une des prévisions les plus immédiates de la CDT est que l'occurrence de structures remarquables augmente l'intérêt. Le « monde » exige les mêmes efforts pour produire une structure typique non remarquable $s_r$ et pour produire une structure remarquable $s$ : $C_w(s_r) = C_w(s)$. Mais les structures remarquables sont simples et ainsi inattendues :

$$U(s) = C(s_r) - C(s)$$

Dans l'histoire suivante, l'option (a) (numéro 4444) économise trois instanciations, car le chiffre 4 est simplement copié. L'inattendu vaut donc $U_a = 3 \times \log_2 10 = 10$ bits. Dans l'option (b), pour la même raison, l'inattendu s'élève à au moins $U_b = 6.6$ bits, tandis qu'il vaut zéro pour l'option (c).

> C'est marrant, j'ai trouvé ça sur Internet: la ville de St-Chéron compte [...] habitants.
>
> a: 4444
> b: 4000
> c: 3856

| Préférences | Rangs |
|---|---|
| a-b* 35/1 | a 92 |
| b-c* 21/11 | b 4 |
| a-c* 30/2 | c 3 |

Les résultats confirment ces prévisions. La structure la plus simple (a) a été fortement préférée à (b) et à (c). L'option (b) a été préférée à (c), bien que moins de manière significative.

## 5 Discussion

La plupart des résultats confirment les prédictions de la CDT. Il y a quelques légères

différences, que nous allons commenter. Auparavant, nous devons répondre à la question récurrente concernant la non-calculabilité théorique de la complexité.

### 5.1 Mesurer la complexité

La non-calculabilité théorique de la complexité de Kolmogorov est un résultat facile à démontrer. Nous évitons cette difficulté en considérant deux restrictions. D'abord, nous définissons la complexité, non comme un minimum objectif, mais comme la longueur de la description la plus concise *disponible*. Ceci peut mener à une sous-estimation de l'inattendu et expliquer pourquoi quelques sujets peuvent ne pas saisir l'intérêt que d'autres sujets trouvent dans une histoire. Notre deuxième restriction est que la complexité cognitive est calculée sur une machine spécifique, « la machine cognitive », c.-à-d. l'ensemble des outils cognitifs dont les humains sont censés disposer. Ceci présuppose que le modèle cognitif que nous employons soit rendu explicite dans chaque cas. Dans les exemples commentés précédemment, la plupart des calculs ont été faits en utilisant des hypothèses minimales en ce qui concerne ces capacités cognitives. Comme nous l'avons montré, le calcul de la complexité cognitive est parfaitement possible dans la plupart des cas. Trois aspects de la complexité sont, cependant, externes au modèle : la complexité conceptuelle, la complexité structurelle et la complexité causale.

La *complexité conceptuelle* intervient quand des prototypes et des propriétés caractéristiques, notés *r* et *f* dans nos exemples, sont impliqués. Une méthode pour évaluer $C(r)$ ou $C(f)$ serait d'utiliser des graphes ontologiques ou des distances dans des corpus comme Wikipedia. Nous sommes en train d'explorer cette possibilité.

La *complexité structurelle* peut être approchée en utilisant des outils de compression, tels que gzip ou bzip2 (Cilibrasi et Vitányi, 2007). Ces outils permettent de reproduire la hiérarchie des options dans l'histoire de St Chéron (en moyennant sur 1000 mesures pour compenser les limitations des compresseurs pour les chaînes courtes). Bien que les valeurs obtenues sous-estiment l'inattendu, de tels outils peuvent être utiles quand la complexité doit être évaluée par un système automatisé.

La *complexité causale* $C(H)$ est la complexité du scénario le plus simple qui permet de produire la situation dans le monde considéré. Elle est équivalente à la parcimonie d'explication (Feldman, 2004 ; Chaitin, 2004). La complexité d'un scénario causal dépend de la capacité du sujet d'imaginer des causes plausibles par abduction. La complexité de génération $C_w(s)$ est alors calculée de manière récursive à partir de la complexité de génération de ses causes (Dessalles, 2008b). Comme modéliser l'abduction se révèle problématique dès que le sens commun est impliqué (Magnani, 2001), nous sommes à la recherche d'une méthode indirecte. Une technique prometteuse consiste à employer une distance informationnelle telle que la distance normalisée Google (NGD) (Cilibrasi et Vitányi, 2007). Le Web contient la trace textuelle d'innombrables événements non inattendus. La co-occurrence statistique des termes décrivant un événement *s* doit donc être négativement corrélée avec la complexité $C_w(s)$. Dans l'histoire de la gifle, il est censé être plus difficile de produire l'occurrence d'un homme qui en gifle un autre dans la rue qu'un événement dans lequel l'homme demande simplement l'heure. Une application directe de la formule NGD donne une distance (normalisée entre 0 et 1) entre « rue » et « gifler » de 0.74 qui est sensiblement plus élevée que la distance entre « rue » et « demander » (0.15). Par comparaison, « rue » est proche de « bâtiment » (0.08) et éloignée d'un concept abstrait comme « configurer » (0.84). Ces distances pourraient offrir des estimations fiables de la complexité (Cilibrasi et Vitányi, 2007), bien qu'il faille encore travailler pour rendre ce type de calcul plus robuste en fonction du choix du moteur de recherche et des corpus de textes utilisés (Lindsey *et al.*, 2007).

### 5.2 Résultats inattendus

La plupart des écarts entre les résultats et les prévisions peuvent être attribués à deux

causes. L'une vient du caractère artificiel de la situation expérimentale, qui diffère de la situation idéale dans laquelle les sujets ne sauraient pas qu'ils sont testés. Par exemple, il est étonnant que trois participants jugent plus intéressant que l'offre promotionnelle de l'opérateur arrive deux semaines après l'achat du mobile, plutôt que juste dix minutes après. La consigne demandait de choisir « l'alternative qui rend l'histoire aussi intéressante que possible », en proposant une récompense (une clé USB) pour le participant qui ferait les choix considérés comme intéressants par les autres participants. Il est possible que certains participants aient pu mal interpréter la consigne.

L'autre source d'écart par rapport aux prédictions vient du caractère peu vraisemblable de certaines options, parfois mentionné de manière explicite dans les commentaires que nous avons recueillis. Par exemple :

`Pour la deuxième fois je coche un truc moins intéressant mais plus crédible.`

ou encore :

`la réponse la plus "choc" n'est pas forcément celle qui rend l'auditoire le plus attentif - ça en fait trop.`

Ce phénomène a certainement joué dans l'histoire du double suicide, dans laquelle le tatouage le plus détaillé n'a pas été retenu. Pour l'histoire de la saisie de drogue, nous avons eu le commentaire suivant :

`5 kilos, c'est une affaire internationale. 2 kilos, peut-être les gens n'en auront pas entendu parler quand je leur raconterai. Donc ca m'intéresse plus.`

Certaines réponses peuvent résulter d'erreurs ou de réflexions complexes, comme le révèle le commentaire suivant à propos de l'histoire de la gifle, pour laquelle le participant a préféré « demander l'heure » :

`C'est intéressant parce qu'on se demande ce qui pousse la personne à raconter ca. Ce type doit être vraiment spécial pour que cet événement soit marquant. Mais qu'est-ce qu'il a de spécial? On veut en savoir plus.`

La seule vraie surprise de l'expérience est venue de l'histoire suivante.

> J'ai dû payer 400 euros une amende vieille d'un an à cause du trésor public qui n'a pas envoyé les rappels à la bonne adresse. Au moment du prélèvement, j'avais exactement [...] euros sur mon compte.
>
> a: 400
> b: 401
> c: 419

| Préférences | | Rangs |
|---|---|---|
| a-b | 13/20 | a 33 |
| b-c* | 29/5 | b 56 |
| a-c* | 24/9 | c 11 |

Les participants ont préféré 401 à 400 euros, alors que CDT prévoit *a priori* que 400 est plus simple. Une explication possible est que 400 évoque un nombre arrondi, et donc une classe de situations, tandis que 401 est à la fois simple et unique (comme pour 1001 nuits *vs*. 1000 nuits). Nous projetons de tester cette hypothèse plus avant.

## 6 Perspectives

On peut penser à une variété d'explications séparées pour les différents types d'histoire : une explication pour des analogies, une pour la proximité, une pour la structure, et ainsi de suite. L'expérience décrite en cet article conforte l'idée inverse qu'un principe cognitif unique et simple est à l'œuvre : la baisse de complexité permet de reproduire de manière correcte toutes les prédictions sans recourir à des hypothèses *ad hoc*. En particulier, le sentiment d'improbabilité qui est souvent mentionné en lisant les histoires de notre test n'est pas une dimension séparée d'intérêt. La probabilité subjective $p$ peut être déduite de l'inattendu $U$ (Dessalles, 2006 ; 2008b) par la formule :

$$p = 2^{-U}$$

La difficulté principale avec notre approche expérimentale de l'intérêt narratif est que les histoires dans ce type de test se situent à mi-chemin entre la fiction et la non-fiction. Nous travaillons actuellement à une nouvelle conception de l'expérience dans laquelle les

jugements d'intérêt seront plus conformes à ceux d'une mise en situation réelle. Une autre perspective est de rendre des calculs de complexité automatiques, en utilisant des distances issues de mesures de co-occurrence sur le Web. Les applications pratiques de ces recherches sont diverses, et vont du calcul de l'intérêt médiatique (*newsworthiness*) aux moteurs de recherche événementiels.

## Références


Chaitin, G. J. (2004). On the intelligibility of the universe and the notions of simplicity, complexity and irreducibility. In Hogrebe & Bromand (Eds.), *Grenzen und Grenzüberschreitungen, XIX*, 517-534. Berlin: Akademie Verlag.

Chater, N. (1999). The search for simplicity: A fundamental cognitive principle?. *The Quaterly Journal of Experimental Psychology*, *52* (A), 273-302.

Chater, N. & Vitányi, P. (2003). Simplicity: a unifying principle in cognitive science?. *Trends in cognitive sciences*, *7* (1), 19-22.

Cilibrasi, R. & Vitányi, P. (2007). The Google similarity distance. *ACM Transactions on Knowledge and Data Engineering*. http://www.cwi.nl/~paulv/papers/amdug.pdf

Cornuéjols, A. (1996). Analogie, principe d'économie et complexité algorithmique. In *Actes des 11èmes Journées Françaises de l'Apprentissage*. Sète. http://www.lri.fr/~antoine/Papers/JFA96-final-osX.pdf

Delahaye, J-P. & Zenil, H. (2007). On the Kolmogorov-Chaitin Complexity for short sequences. In C.S. Calude (Ed.), *Randomness and Complexity: From Leibniz to Chaitin*. Singapore: World Scientific. http://arxiv.org/abs/0704.1043

Dessalles, J-L. (2006). A structural model of intuitive probability. In D. Fum, F. Del Missier & A. Stocco (Eds.), *Proceedings of the seventh International Conference on Cognitive Modeling*, 86-91. Trieste, IT: Edizioni Goliardiche. http://www.dessalles.fr/papiers/pap.cogni/Dessalles_06020601.pdf

Dessalles, J-L. (2008a). Coincidences and the encounter problem: A formal account. In B. C. Love, K. McRae & V. M. Sloutsky (Eds.), *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, 2134-2139. Austin, TX: Cognitive Science Society. http://www.dessalles.fr/papiers/pap.conv/Dessalles_08020201.pdf

Dessalles, J-L. (2008b). *La pertinence et ses origines cognitives - Nouvelles théories*. Paris: Hermes-Science Publications. http://pertinence.dessalles.fr

Eggins, S. & Slade, D. (1997). *Analysing casual conversation*. London: Equinox.

Farkas, A. (2007). The analysis of the principal eigenvector of pairwise comparison matrices. *Acta Polytechnica Hungarica*, *4* (2). http://bmf.hu/journal/Farkas_10.pdf

Feldman, J. (2004). How surprising is a simple pattern? Quantifying 'Eureka!'. *Cognition*, *93*, 199-224.

Galtung, J. & Ruge, M. H. (1965). The structure of foreign news: The presentation of the Congo, Cuba and Cyprus crises in four foreign newspapers. *Journal of International Peace Research*, *2* (1), 64-91.

Labov, W. (1997). Some further steps in narrative analysis. *Journal of Narrative and Life History*, *7* (1-4), 395-415. http://www.ling.upenn.edu/~wlabov/sfs.html

Labov, W. & Waletzky, J. (1967). Narrative analysis: Oral versions of personal experience. In J. Helm (Ed.), *Essays on the verbal and visual arts*, 12-44. Seattle, WA: University of Washington Press. http://www.clarku.edu/~mbamberg/LabovWaletzky.htm

Lindsey, R., Veksler, V. D., Grintsvayg, A. & Gray, W. D. (2007). Be wary of what your computer reads: The effects of corpus selection on measuring semantic relatedness. In *Proceedings of the 8th International Conference on Cognitive Modeling*. Ann Arbor, MI. http://www.cogsci.rpi.edu/cogworks/publications/271_ICCM_final.pdf

Li, M. & Vitányi, P. (1993). *An introduction to Kolmogorov complexity and its applications*. New York: Springer Verlag, ed. 1997.

Magnani, L. (2001). *Abduction, reason and science - Processes of discovery and explanation*. New York: Kluwer Academic.

Mehl, M. R. & Pennebaker, J. W. (2003). The sounds of social life: A psychometric analysis of students' daily social environments and natural conversations. *Journal of Personality and Social Psychology*, *84* (4), 857-870.

Norrick, N. R. (2000). *Conversational narrative: storytelling in everyday talk*. Amsterdam: John Benjamins Publishing Company.

Rimé, B. (2005). *Le partage social des émotions*. Paris: PUF.

Saaty, T. L. (1994). How to make a decision: The analytic hierarchy process. *Interfaces*, *24* (6), 19-43.

Shapiro, M. A. & Fox, J. R. (2002). The role of typical and atypical events in story memory. *Human Communication Research*, *28* (1), 109-135.

Solomonoff, R. J. (1978). Complexity-based induction systems: Comparisons and convergence theorems. *IEEE transactions on Information Theory*, *24* (4), 422-432. http://world.std.com/~rjs/solo1.pdf

Stangor, C. & McMillan, D. (1992). Memory for expectancy-congruent and expectancy-incongruent information: a review of the social and social developmental literatures. *Psychological Bulletin*, *111* (1), 42-61.

Tannen, D. (1984). *Conversational style - Analyzing talk among friends*. Norwood: Ablex Publishing Corporation.

Woll, S. B. & Graesser, A. C. (1982). Memory discrimination for information typical and atypical of person schemata. *Social Cognition*, *1*, 287-310.

# Un point de vue épistémique sur la fusion de croyances : l'identification du monde réel

Patricia Everaere* Sébastien Konieczny° Pierre Marquis°
patricia.everaere@univ-lille1.fr konieczny@cril.fr marquis@cril.fr

*Laboratoire d'Informatique Fondamentale de Lille
Université des Sciences et Technologies de Lille
59655 Villeneuve d'Ascq Cedex – France

°Centre de Recherche en Informatique de Lens
Université d'Artois
62307 Lens Cedex – France

**Résumé :**
La fusion de croyances est souvent décrite comme un processus permettant de définir une base de croyances qui représente au mieux les croyances d'un groupe d'agents (un profil de bases de croyances). La base résultant de ce processus peut être considérée comme une synthèse du profil donné. Dans cet article, nous proposons un autre point de vue, de type épistémique, sur la fusion de croyances. Sous ce point de vue, le but de la fusion de croyances est d'approcher au mieux le monde réel. Nous étudions des généralisations du Théorème du Jury de Condorcet du point de vue de la fusion de croyances. Nous montrons que si l'on fusionne les croyances d'un nombre suffisamment grand d'agents fiables, il est possible d'assurer (en probabilité) l'identification du monde réel. Nous montrons que quelques (mais pas tous les) opérateurs de fusion de la littérature ont un bon comportement pour ce problème de suppression de l'incertitude.

**Mots-clés :** Fusion de croyances, théorème du jury de Condorcet

**Abstract:**
Belief merging is often described as the process of defining a belief base which best represents the beliefs of a group of agents (a profile of belief bases). The resulting belief base can be viewed as a synthesis of the input profile. In this paper another view, called the epistemic view, of what belief merging aims at is considered. Under this view the purpose of belief merging is to best approximate what is the true state of the world. We study generalizations of Condorcet's Jury Theorem from the belief merging perspective. Roughly we show that if we merge the beliefs of sufficiently many reliable agents then we can ensure to identify the true state of the world. We show that some (but not all) merging operators from the literature are suited to the truth tracking issue.

**Keywords:** Belief Merging, Condorcet's Jury Theorem

## 1 Introduction

De nombreux problèmes issus de domaines variés de l'informatique, comme par exemple les bases de données distribuées et les systèmes multi-agents, nécessitent de synthétiser des informations cohérentes provenant de plusieurs sources d'information. Réaliser une telle synthèse est en général difficile car, en particulier, les informations provenant de différentes sources peuvent se contredire les unes les autres. Lorsque les informations disponibles sont des croyances représentées par des formules de la logique propositionnelle, ce problème est appelé fusion de croyances (propositionnelles). Dans cet article, nous nous intéressons au cas purement logique, i.e., nous supposons que les bases de croyances sont des ensembles de formules propositionnelles (voir par exemple [2, 20, 12, 11, 9]). De nombreux opérateurs de fusion ont été introduits dans un tel cadre. Les propriétés logiques de ces opérateurs de fusion ont été largement étudiées et sont décrites dans de nombreux travaux [20, 14, 12]. Ainsi, [12] présente un ensemble de postulats permettant de caractériser la famille des opérateurs de fusion contrainte. D'autres opérateurs ont été introduits dans des cadres plus généraux, comme par exemple en logiques pondérées (logique possibiliste ou basée sur des fonctions ordinales conditionnelles) [3, 16, 4] ; ils se révèlent très utiles lorsque des informations supplémentaires sont disponibles (en particulier, lorsque toutes les croyances ne sont pas également sûres). Dans ces cadres plus généraux, le problème de la fusion devient proche de l'agrégation de préordres (préférences), très étudiée en théorie du choix social [1, 19].

Jusqu'à présent, les opérateurs de fusion de croyances existants sont aussi considérés comme adaptés à la fusion de buts. Cela a du sens, puisque dans les deux cas la fusion cherche à synthétiser les informations représentées dans le profil initial de bases propositionnelles. Même s'il peut sembler étrange à première vue que des concepts si différents (croyances ou buts) puissent être traités de façon identique du point de vue de l'agrégation, la pertinence des opérateurs de fusion contrainte [13] pour la fusion propositionnelle (que ce soit de buts ou de croyances) n'a pas encore été remise en cause par l'adjonction de nouveaux postulats

qui permettraient de séparer les deux types de fusion.

Dans cet article, nous introduisons un nouveau point de vue sur la fusion de croyances, qui est différent du point de vue synthétique habituel et qui permet de justifier une telle distinction entre fusion de croyances et fusion de buts : le point de vue épistémique.

**Vue synthétique :** Sous le point de vue synthétique, comme évoqué auparavant, la fusion a pour objectif de caractériser une base qui représente au mieux les croyances du profil initial. C'est la vue considérée dans les travaux précédents concernant la fusion de croyances.

**Vue épistémique :** Sous le point de vue épistémique, le but d'un processus de fusion est d'estimer au mieux le monde réel, donc de supprimer autant que possible l'incertitude du groupe d'agents à son sujet.

Dans le cas général, aucun agent n'a une connaissance parfaite du monde réel, les croyances sont entachées d'incertitude :

1. Typiquement, un agent ne sait pas quel modèle de sa base correspond au monde réel,
2. Il n'est même pas certain que le monde réel se trouve réellement parmi les modèles de sa base.[1]

La fusion de croyances sous le point de vue épistémique peut être considérée comme un moyen de lever l'incertitude sur le monde réel au niveau du groupe.

Il est remarquable que la recherche du monde réel soit un moyen de différencier la fusion de croyances de la fusion de buts. En effet, alors que la recherche du monde réel peut être attendue lors de la fusion de croyances dans de nombreuses circonstances, le concept de recherche du monde réel n'a pas de sens pour des buts. Il est en effet assez clair que la notion de "vrai but", équivalent dans le cadre des buts du monde réel, ne signifie rien.

L'intuition derrière la vue épistémique de la fusion de croyances est que si les agents sont indépendants et dignes de confiance, alors on peut espérer d'un opérateur de fusion qu'il soit capable d'identifier en probabilité le monde réel, en s'appuyant sur un nombre suffisamment grand d'agents.

Le problème de l'identification du monde réel (*Truth Tracking*) a été étudié depuis des siècles en choix social et en sciences politiques, afin de justifier les fondations des élections démocratiques ou les décisions prises par des jurys. Le principal résultat théorique est le théorème du jury de Condorcet [7]. Ce théorème établit que si un jury est composé d'individus indépendants et dignes de confiance, et qu'ils ont à trouver la véritable réponse oui/non à une question, alors la probabilité que la décision prise par le jury à la majorité soit la bonne tend vers 1 quand la taille du jury tend vers l'infini.

Dans ce travail, nous formalisons le problème de l'identification du monde réel dans une perspective de fusion de croyances. Nous montrons que certains opérateurs de fusion de croyances peuvent être utilisés pour identifier l'état du monde en considérant suffisamment d'agents indépendants et fiables. Plus précisément, la contribution de ce papier est principalement comme suit : nous présentons une généralisation du théorème du jury de Condorcet au cas incertain (i.e., lorsque chaque base peut contenir plusieurs modèles). Nous introduisons un postulat d'identifiabilité à la limite du monde réel, noté **TT**. Nous présentons certains opérateurs de fusion contrainte satisfaisant ce postulat et montrons que d'autres opérateurs ne le satisfont pas. En conséquence, nous concluons que certains, mais pas tous les opérateurs de fusion de la littérature sont adaptés à l'identification du monde réel. Nous présentons également des résultats expérimentaux sur la vitesse de convergence vers le monde réel lorsque l'opérateur de fusion considéré est $\Delta^{d_D,\Sigma}$. Dans le plupart des cas, le nombre d'agents nécessaires pour garantir que la base fusionnée identifie le monde réel avec une probabilité élevée n'est pas très grand. Cela montre la faisabilité pratique de la recherche du monde réel en utilisant $\Delta^{d_D,\Sigma}$.

Le reste de l'article est organisé ainsi : nous donnons quelques préliminaires formels en section 2. Nous rappelons le théorème du jury de Condorcet, et certaines de ses généralisations en section 3. En section 4, nous présentons de nouvelles généralisations du théorème du jury de Condorcet en présence d'incertitude, utiles pour la fusion de croyances. En section 5, nous montrons que certains opérateurs de fusion contrainte de la littérature (mais pas tous) satis-

[1] Si l'on suppose que les agents savent que le monde réel est un modèle de leur base, on peut parler de "connaissances"– cette hypothèse est la seule différence entre croyances et connaissances. Or, la *fusion de connaissances* n'est pas intéressante, puisque la seule méthode raisonnable pour fusionner des connaissances est évidemment la conjonction.

font **TT**. En Section 6, nous présentons les résultats empiriques que nous avons obtenus et les analysons. Finalement nous concluons en section 7.

## 2 Préliminaires

Nous considérons un langage propositionnel $\mathcal{L}$ défini à partir d'un ensemble fini de variables propositionnelles $\mathcal{P}$ et des connecteurs usuels.

Pour tout sous-ensemble $c$ de $\mathcal{P}$, $|c|$ dénote le nombre d'éléments de $c$. Une interprétation (ou état du monde) est une application de $\mathcal{P}$ dans $\{0, 1\}$. L'ensemble de toutes les interprétations est notée $\mathcal{W}$. Le véritable état du monde est notée $\omega^\star$. Une interprétation $\omega$ est un modèle d'une formule $\phi \in \mathcal{L}$ si et seulement si elle la rend vraie au sens usuel. $[\phi]$ représente l'ensemble de modèles de la formule $\phi$, i.e., $[\phi] = \{\omega \in \mathcal{W} \mid \omega \models \phi\}$.

Une *base* $K$ dénote l'ensemble des croyances d'un agent, c'est un ensemble fini de formules propositionnelles, interprété conjonctivement. Nous identifions $K$ avec la conjonction de ses éléments. Toute base $K$ représente un ensemble $[K]$ d'états du monde.

Un *profil* $E$ représente les croyances d'un groupe de $n$ agents impliqués dans le processus de fusion. Dans cet article, les agents expriment parfois un seul monde possible. Dans ce cas un profil est un vecteur de bases complètes. Pour éviter des notations trop lourdes, nous identifions chaque base complète avec son modèle et écrivons ces profils $E_c = \langle \omega_1, \ldots, \omega_n \rangle$. Partout ailleurs les agents expriment des ensembles de mondes possibles, ainsi $E$ est représenté comme un vecteur de bases $E = \langle K_1, \ldots, K_n \rangle$, comme c'est souvent l'usage en fusion propositionnelle.

## 3 Theorème du jury de Condorcet et extensions

Nous considérons un profil de $n$ agents où chaque agent $i$ vote pour une alternative, par exemple un état du monde $\omega_i$. Parmi tous les mondes possibles se trouve le véritable monde $\omega^\star$. Les hypothèses du théorème du jury de Condorcet sont que les agents sont à la fois indépendants et fiables. Les agents sont *indépendants* si connaître le monde réel et le monde donné par n'importe quel agent ne donne aucune information supplémentaire sur le monde donné par tout autre agent (cela signifie que les choix des agents sont indépendants conditionnellement au monde réel, au sens Bayésien standard [17]). Comme de nombreuses notions de fiabilité vont être considérées dans la suite, nous appelons la première *R1-fiabilité* :

– La *R1-fiabilité* $p_i$ d'un agent $i$ est la probabilité que $i$ donne le monde réel, i.e., $p_i = P(\omega_i = \omega^\star)$. **(R1)**
– Un agent $i$ est *R1-fiable* si sa R1-fiabilité est strictement plus élevée que $0.5$.

La règle majoritaire donne simplement comme résultat l'interprétation qui reçoit une majorité stricte de votes. Formellement, nous définissons une notion de score d'un monde par rapport à un profil de bases complètes :

$$s(\omega) = |\{\omega_i \in E_c \mid \omega_i = \omega\}|.$$

**Définition 1 (Majorité)** *Etant donné un profil $E_c$ de $n$ bases complètes, la règle de la* majorité $m$ *est définie par :*

$$m(E_c) = \omega \text{ si } s(\omega) > n/2.$$

Rappelons à présent le théorème du jury de Condorcet. Dans ce théorème, deux alternatives sont considérées seulement et chaque agent vote pour une des deux (pas d'abstention possible) :

**Théorème 1 ([7])** *Considérons deux mondes possibles $\mathcal{W} = \{\omega, \omega^\star\}$ et un profil $E_c$ de bases complètes issues d'un ensemble de $n$ agents indépendants et R1-fiables où tous les agents partagent la même R1-fiabilité $p$. La probabilité que la règle de la majorité sur ce profil retourne le monde réel $\omega^\star$ tend vers $1$ lorsque $n$ tend vers l'infini, i.e. :*

$$P(m(E_c) = \omega^\star) \xrightarrow[n\to\infty]{} 1.$$

Ce théorème est une conséquence de la loi faible des grands nombres. Schématiquement, il établit que si les individus d'un jury sont suffisamment fiables (i.e., que leurs décisions sont meilleures que celles obtenues par tirage au sort à pile ou face) et indépendants, alors la fiabilité du jury augmente avec sa taille et converge vers 1.

Il est assez clair que les deux hypothèses de ce théorème sont assez fortes. D'abord, il est inhabituel que les agents d'un jury soient totalement indépendants : ils ont souvent des connaissances partagées, une culture similaire, sont attentifs

aux mêmes leaders d'opinion, etc. De plus, en général, tous les agents n'ont pas exactement la même fiabilité : il y a souvent des agents plus compétents que d'autres. Des extensions du theorème du jury de Condorcet montrent qu'il est possible de relâcher certaines des hypothèses du théorème sans changer la conclusion. Ainsi, la conclusion du théorème est toujours vraie lorsque les opinions des individus ne sont pas totalement indépendantes [8]. Concernant la fiabilité, il est suffisant de supposer que la fiabilité moyenne des individus est supérieure à 0.5 [10].

Une autre limitation importante du théorème du jury de Condorcet est qu'il ne considère que deux alternatives seulement. Un résultat récent [15] permet d'étendre ce théorème à un nombre fini quelconque d'options. Afin de présenter ce résultat, nous devons d'abord rappeler la définition de la règle de la pluralité :

**Définition 2 (Pluralité)** *Soit un profil $E_c$ de bases complètes, la règle de la* pluralité *$pl$ est définie par :*

$$pl(E_c) = \{\omega \mid \forall \omega' \in \mathcal{W} \;\; s(\omega) \geq s(\omega')\}.$$

L'hypothèse de fiabilité **(R1)** doit être étendue si l'on considère plus de deux alternatives. List et Goodin [15] ont défini la notion suivante de fiabilité. Soit un ensemble $\mathcal{W} = \{\omega_1, \ldots, \omega_{k-1}, \omega^\star\}$ de $k$ mondes possibles :
– Un agent est *R2-fiable* si la probabilité qu'il vote pour $\omega^\star$ est strictement supérieure à la probabilité qu'il vote pour n'importe quel autre monde.[2] **(R2)**
List et Goodin ont montré que :

**Théorème 2 ([15])** *Soit $\mathcal{W} = \{\omega_1, \ldots, \omega_{k-1}, \omega^\star\}$ un ensemble de $k$ mondes possibles et soit un profil $E_c$ de bases complètes issues d'un ensemble de $n$ agents indépendants et R2-fiables. La probabilité que la règle de la pluralité sur ce profil retourne le monde réel $\omega^\star$ tend vers $1$ lorsque $n$ tend vers l'infini, i.e.,*

$$P(pl(E_c) = \{\omega^\star\}) \xrightarrow[n\to\infty]{} 1.$$

Ce théorème est une généralisation du théorème du jury de Condorcet. On peut noter que la règle de la pluralité est utilisée (et non la règle de la majorité) et que l'hypothèse de fiabilité demande seulement que la probabilité de voter pour le monde réel soit strictement supérieure à la probabilité de voter pour un autre monde, et donc la probabilité de voter pour le monde réel peut être inférieure à 0.5.[3] Voir [15] pour une discussion de ce résultat et ses conséquences philosophiques, et pour plus de détails sur le théorème du jury de Condorcet.

## 4 Un théorème du jury avec incertitude

Dans tous les travaux précédents autour du théorème du jury de Condorcet, les agents choisissent toujours exactement une alternative unique. Cette hypothèse n'est pas adaptée au cadre de la fusion de croyances, dans lequel chaque agent présente typiquement une base de croyances qui peut contenir de nombreux modèles (et imposer aux agents de donner des bases de croyances complètes serait très restrictif, puisque cela nierait la nature incertaine des croyances des agents). Ainsi, à partir de maintenant, nous supposons que chaque agent $i$ a une base de croyances $K_i$ qui peut avoir plusieurs modèles parmi un ensemble fini $\mathcal{W} = \{\omega_1, \ldots, \omega_{k-1}, \omega^\star\}$.

Le théorème du jury de Condorcet peut être étendu au cas où chaque agent peut voter pour plusieurs alternatives. Tout d'abord, nous avons à définir une notion de fiabilité adaptée à cette situation :

– La *R3-fiabilité* $p_i$ d'un agent $i$ est la probabilité que le monde réel $\omega^\star$ fasse partie des modèles de sa base de croyances $K_i$, i.e., $p_i = P(\omega^\star \models K_i)$. **(R3)**
– Un agent $i$ est *R3-fiable* si $p_i > 0.5$.

De même, la notion de score d'une interprétation doit être étendue de la façon suivante :

$$s_a(\omega) = |\{K_i \in E \mid \omega \models K_i\}|.$$

Nous avons établi le résultat suivant :

**Proposition 1** *Soit $p^\star \in [0, 1[$. On considère un profil $E$ issu d'un ensemble de $n$ agents indépendants ayant tous la même R3-fiabilité $p >$*

[2] Formellement, soit $p_1, \ldots, p_{k-1}, p^\star$ (respectivement) la probabilité qu'un agent vote pour $\omega_1, \ldots, \omega_{k-1}, \omega^\star$ (respectivement). Alors $\forall i \in 1 \ldots k-1 \; p^\star > p_i$.

[3] Clairement, si l'on considère seulement deux états du monde possibles, les hypothèses du théorème de List-Goodin sont équivalentes à celles du théorème du jury de Condorcet, et les deux théorèmes sont identiques dans ce cas.

*$p^\star$. La probabilité que le score du monde réel $\omega^\star$ soit supériuer à $p^\star n$ tend vers $1$ lorsque $n$ tend vers l'infini, i.e,*

$$P(s_a(\omega^\star) > p^\star n) \xrightarrow[n\to\infty]{} 1.$$

**Preuve:** Pour un agent $i$, la probabilité que $\omega^\star \models K_i$ est $p$ ; donc la probabilité que $\omega^\star \not\models K_i$ est $q = 1 - p$. En supposant que les agents sont indépendants, la probabilité que $\omega^\star$ soit un modèle de $k$ parmi $n$ bases vaut $\binom{n}{k} p^k q^{n-k}$. Dans la suite de la preuve, nous utilisons le théorème de Bienaimé-Tchebitchev : *soit $v$ une variable aléatoire de moyenne $m$ et de variance $\sigma^2$. Etant donné $t$, un réel positif, la probabilité que la variable $v$ s'écarte de $m$ d'au moins $t\sigma$ est inférieure à $\frac{1}{t^2}$ :*

$$P(\mid v - m \mid \geq t\sigma] < \frac{1}{t^2} \qquad (1)$$

Soit $v_i$ la variable aléatoire égale à 1 si $\omega^\star \models K_i$, et à 0 sinon. Comme chaque $v_i$ suit une distribution de Bernoulli avec une probabilité de succès $p$, la moyenne et la variance de $v_i$ sont égales à $p$ et $pq$, respectivement. Alors le nombre de bases du profil $E$ parmi les $n$ ayant $\omega^\star$ comme modèle est égale à $v = \Sigma_{i=1}^{n} v_i$. Comme $v$ suit une distribution binomiale de paramètres $n$, $p$, la moyenne de $v$ est égale à $m = np$, et sa variance est égale à $\sigma^2 = npq$.

La probabilité que le nombre de bases ayant $\omega^\star$ comme modèle soit inférieur à $np^\star$ est donnée par $P(v \leq p^\star n)$.

Or, $\mid v - m \mid \geq m - p^\star n$ si et seulement si $v - m \geq m - p^\star n \geq 0$ ou (exclusif) $v - m \leq p^\star n - m < 0$. Donc $P(\mid v - m \mid \geq m - p^\star n) = P(v - m \geq m - p^\star n \geq 0) + P(v - m \leq p^\star n - m < 0)$.

Donc on a $P(\mid v - m \mid \geq m - p^\star n) \geq P(v - m \leq p^\star n - m < 0)$. Comme enfin $v - m \leq p^\star n - m < 0$ si et seulement si $v \leq p^\star n < m$, on obtient :

$$P(v \leq p^\star n < m) \leq P(\mid v - m \mid \geq m - p^\star n)$$

En prenant $t = \frac{m - p^\star n}{\sigma}$ ($t > 0$ si $p > p^\star$), et en utilisant l'inéquation (1), on obtient :

$$P(\mid v - m \mid \geq m - p^\star n) < \frac{\sigma^2}{(m - p^\star n)^2}.$$

En remplaçant $m$ et $\sigma$ par leur valeurs, comme on a toujours $np^\star < np$ lorsque $p^\star < p$, par transitivité, il s'ensuit que :

$$P(v \leq np^\star) < \frac{pq}{n(p - p^\star)^2}.$$

Ce qui conduit à :

$$P(v \leq p^\star n) \xrightarrow[n\to\infty]{} 0.$$

Autrement dit :

$$P(s_a(\omega^\star) > p^\star n) \xrightarrow[n\to\infty]{} 1.$$

□

Ce résultat donne (à la limite et en probabilité) un minorant du score obtenu par le monde réel à partir du moment où les agents ont la même R3-fiabilité. Il assure (toujours à la limite et en probabilité) pour certaines règles de vote que le monde réel fait partie de l'ensemble des mondes retournés par la règle. Considérons par exemple les règles de vote suivantes :

**Définition 3 (**$M$ **et** $Q_p$**)** *Soit $E$ un profil issu d'un ensemble de $n$ agents.*
- *La règle de la* majorité *$M$ est définie par : $M(E) = \{\omega \mid s_a(\omega) > n/2\}$.*
- *Plus généralement, étant donné $k \in ]0, 1[$, la règle du $k$-*quota *$Q_k$ est définie par : $Q_k(E) = \{\omega \mid s_a(\omega) > kn\}$.*

La règle de la majorité $M$ étend au cas où les agents expriment des croyances incertaines la règle de majorité précédente $m$.

On obtient le corollaire suivant à la proposition précédente :

**Corollaire 1** *Soit $E$ un profil issu d'un ensemble de $n$ agents indépendants.*
- *Si tous les agents sont R3-fiables et ont la même R3-fiabilité $p$, alors le monde réel est capturé à la limite par la règle de la majorité, i.e.,*

$$P(\omega^\star \in M(E)) \xrightarrow[n\to\infty]{} 1.$$

- *Si tous les agents ont la même R3-fiabilité $p > k$, alors le monde réel est capturé à la limite par la règle du $k$-quota, i.e.,*

$$P(\omega^\star \in Q_k(E)) \xrightarrow[n\to\infty]{} 1.$$

On peut souligner que cette proposition ne porte que sur la présence du monde réel dans le résultat du processus de vote. En aucun cas elle n'exclut qu'un autre monde soit présent dans ce résultat. Bien sûr, cela pose un problème pour la détermination du monde réel. En particulier, si chacun des agents $i$ donne tous les mondes possibles ($[K_i] = \mathcal{W}$), alors toute fusion raisonnable du profil $E$ correspondant (en particulier toute fusion réalisée via un opérateur vérifiant le postulat **(IC2)**, [12]) donnera comme résultat tous les mondes possibles (par exemple $Q_k(E) = \mathcal{W}$ quel que soit $k$), ce qui ne donne aucune information sur le monde réel.

Le problème est dû à la notion de R3-fiabilité, qui n'est pas assez forte pour l'identification du monde réel. Intuitivement, demander aux agents de choisir le monde réel avec une probabilité élevée est nécessaire mais pas suffisant puisque cela n'empêche pas les agents de donner (comme modèles de leurs bases) d'autres mondes possibles. Par exemple, un agent $i$ dont la base est toujours une tautologie ($[K_i] = \mathcal{W}$), qui ne donne aucune information, est considéré complètement R3-fiable (i.e., sa R3-fiabilité $p_i$ est égale à 1), ce qui est inadapté. Ainsi, une notion de fiabilité plus forte est nécessaire. La notion de R4-fiabilité qui suit répond à cette demande :

- L'*incompétence* $q_i$ d'un agent $i$ est la probabilité maximale qu'un monde différent de $\omega^\star$ appartienne à l'ensemble de modèles de sa base, i.e., $q_i = max_{\omega_j \neq \omega^\star}(\{P(\omega_j \models K_i)\})$. La *compétence* d'un agent est $c_i = 1 - q_i$.
- Un agent est *compétent* si $c_i > 0.5$.
- Un agent $i$ est *R4-fiable* si sa R3-fiabilité $p_i$ est strictement supérieure à son incompétence $q_i$ : $p_i > q_i$. **(R4)**

Intuitivement, alors que la R3-fiabilité exprime la faculté d'un agent à avoir des croyances compatibles avec le monde réel, la notion de compétence traite de la quantité d'incertitude sur ses croyances. Nous pensons que la R3-fiabilité et la compétence sont des notions à la fois naturelles et importantes pour caractériser l'idée intuitive d' "agent fiable" dans le cadre de la fusion de croyances. Dans le cas spécifique où $\mathcal{W}$ est réduit à deux alternatives seulement, un agent est compétent si et seulement s'il est R3-fiable ; à l'opposé, dans le cas général, ces notions sont bien distinctes. Il est facile de montrer que la notion de R4-fiabilité étend les notions précédentes de fiabilité :

**Proposition 2** – *Quand on considère seulement des profils $E_c$ de bases complètes, les fiabilités de type R4 et R2 sont équivalentes.*
– *Quand on considère seulement des profils $E_c$ de bases complètes et un ensemble $\mathcal{W}$ d'interprétations constitué de deux éléments $\{\omega, \omega^\star\}$, les fiabilités de type R4, R3, R2 et R1 sont équivalentes.*

Les hypothèses supplémentaires suivantes sur l'ensemble d'agents considérés vont permettre de dériver des résultats intéressants :

- Un ensemble d'agents est *homogène* si pour chaque monde possible $\omega_j$, la probabilité $P(\omega_j \models K_i)$ que ce monde soit un modèle de la base $K_i$ fournie par chacun des agents de l'ensemble est la même pour tous les agents $i$ de l'ensemble. En particulier le monde réel $\omega^\star$ a la même probabilité d'apparaître comme modèle chez chacun des agents.
- Un ensemble d'agents homogène est *uniforme* si pour chaque monde possible $\omega_j$ sauf le monde réel $\omega^\star$, les probabilités $P(\omega_j \models K_i)$ coincident.

**Proposition 3** *Soit $E$ un profil issu d'un ensemble uniforme de $n$ agents indépendants. Alors la base $K$ la plus probable (à l'équivalence logique près) dans $E$ est $[K] = \{\omega^\star\}$.*

**Preuve:** Soit $p$ la probabilité pour un agent de choisir une base $K$ telle que le monde réel $\omega^\star$ vérifie $\omega^\star \models K$, et soit $q$ la probabilité pour un agent de choisir une base $K$ telle qu'un monde quelconque $\omega$ différent de $\omega^\star$ vérifie $\omega \models K$. Soit $\mathcal{W} = \{\omega_1, \ldots, \omega_{k-1}, \omega^\star\}$ l'ensemble de tous les mondes possibles ($\mathcal{W}$ contient $k$ éléments). Grâce à l'hypothèse d'indépendance, la probabilité pour un agent de choisir une base $K$ telle que $[K] = \{\omega^\star\}$ est égale à $p(1-q)^{k-1}$. Plus généralement, la probabilité pour un agent de choisir une base $K$ est égale à $p^a(1-p)^b q^c (1-q)^d$ avec $a, b, c, d$ des entiers positifs tels que $a + b = 1$ et $a + b + c + d = k$. Sous l'hypothèse de R4-fiabilité de l'ensemble d'agents, on a $p > 0.5$ et $q < 0.5$, donc $p > (1-p)$ et $q < (1-q)$. En conséquence, $p^a(1-p)^b q^c (1-q)^d$ est maximal pour $a = 1, b = 0, c = 0, d = k-1$, seulement. Ainsi, la base $K$ la plus probable (à l'équivalence logique près) dans $E$ est $[K] = \{\omega^\star\}$. □

Ce résultat n'est néanmoins pas suffisant pour obtenir directement le théorème du jury sous incertitude qui suit :

**Théorème 3** *Soit $\mathcal{W} = \{\omega_1, \ldots, \omega_{k-1}, \omega^\star\}$ un ensemble de mondes possibles et soit $E$ un profil issu d'un ensemble homogène de $n$ agent indépendants et R4-fiables. Alors $\forall i \in \{1, \ldots, k-1\}$,*

$$P(s_a(\omega^\star) > s_a(\omega_i)) \xrightarrow[n\to\infty]{} 1.$$

**Preuve:** Soit $(s_a(\omega_1), \ldots, s_a(\omega_{k-1}), s_a(\omega^*))$ un vecteur de variables aléatoires où $s_a(\omega_i) = j$ ($i \in \{1, \ldots, k-1\}$) (resp. $s_a(\omega^*) = j$) signifie que le score $s_a(\omega_i)$ (resp. $s_a(\omega^*)$) est égal à $j$ ($j \in \{0, \ldots, n\}$). Chacune de ces variables $s_a(\omega_i)$ ($i \in \{1, \ldots, k-1\}$) (resp. $s_a(\omega^*)$) suit une distribution binomiale de paramètres $q_i$ et $n$ (resp. $p$ et $n$). Ainsi, on a $\forall j \in \{0, \ldots, n\}$ :

$$P(s_a(\omega_i) = j) = \binom{n}{j} q_i^j (1 - q_i)^{n-j}$$

et

$$P(s_a(\omega^*) = j) = \binom{n}{j} p^j (1 - p)^{n-j}.$$

La moyenne de chaque $s_a(\omega_i)$ ($i \in \{1, \ldots, k-1\}$) est $nq_i$, sa variance est $nq_i(1 - q_i)$, la moyenne de $s_a(\omega^*)$ est $np$ et sa variance est $np(1-p)$.

La loi faible des grands nombres appliquée à $s_a(\omega_i)$ ($i \in \{1, \ldots, k-1\}$) et $s_a(\omega^*)$ donne que $\forall \epsilon > 0$ :

$$P(\mid \frac{s_a(\omega_i)}{n} - q_i \mid \geq \epsilon) \longrightarrow_{n\to\infty} 0, \qquad (2)$$

$$P(\mid \frac{s_a(\omega^*)}{n} - p \mid \geq \epsilon) \longrightarrow_{n\to\infty} 0. \qquad (3)$$

Soit $q = max_{i \in \{1,\ldots,k-1\}} q_i$ et $\epsilon_1 = \frac{p-q}{2}$. Comme chaque agent est $R4$-fiable, on a forcément $q_i < p$ pour chaque $i \in \{1, \ldots, k-1\}$, donc $q < p$. En conséquence, on a $\epsilon_1 > 0$. Grâce aux assertions (2) et (3), on obtient pour tout $i \in \{1, \ldots, k-1\}$,

$$P(\mid \frac{s_a(\omega_i)}{n} - q_i \mid \geq \epsilon_1) \longrightarrow_{n\to\infty} 0$$

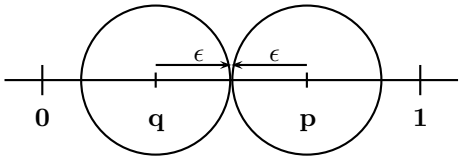
$$et\ P(\mid \frac{s_a(\omega^*)}{n} - p \mid \geq \epsilon_1) \longrightarrow_{n\to\infty} 0.$$

Ce qui permet facilement de déduire que :

$$P(\frac{s_a(\omega_i^*)}{n} > q_i + \epsilon_1) \longrightarrow_{n\to\infty} 0, \qquad (4)$$

et

$$P(\frac{s_a(\omega^*)}{n} < p - \epsilon_1) \longrightarrow_{n\to\infty} 0. \qquad (5)$$




Cette figure illustre la preuve à venir. L'idée est que lorsque la loi faible des grand nombres est utilisable, les valeurs prises se concentrent autours de la moyenne avec une probabilité importante. Schématiquement, la probabilité que toutes les valeurs soient dans une sphère centrée en la moyenne et de rayon $\epsilon_1$ tend vers 1. Ainsi, comme $p > q$, la probabilité que les deux sphères ont une intersection non vide tend vers 0 à l'infini.

Supposons à présent que $\frac{s_a(\omega_i)}{n} \leq q_i + \epsilon_1$ et que $\frac{s_a(\omega^*)}{n} \geq p - \epsilon_1$. Alors, comme $\forall i \in 1 \ldots k-1, q_i + \epsilon_1 \leq p - \epsilon_1$, on obtient :

$$\frac{s_a(\omega_i)}{n} \leq q_i + \epsilon_1 \leq p - \epsilon_1 \leq \frac{s_a(\omega^*)}{n}.$$

Ainsi l'événement $\frac{s_a(\omega_i)}{n} > \frac{s_a(\omega^*)}{n}$ peut se produire uniquement si $\frac{s_a(\omega_i)}{n} > q_i + \epsilon_1$, ou $\frac{s_a(\omega^*)}{n} < p - \epsilon_1$. Et donc on a :

$P(\frac{s_a(\omega_i)}{n} > \frac{s_a(\omega^*)}{n}) = P(\frac{s_a(\omega_i)}{n} > q_i + \epsilon_1$ et $\frac{s_a(\omega_i)}{n} > \frac{s_a(\omega^*)}{n}) + P(\frac{s_a(\omega^*)}{n} < p - \epsilon_1$ et $\frac{s_a(\omega_i)}{n} > \frac{s_a(\omega^*)}{n}) - P(\frac{s_a(\omega_i)}{n} > q_i + \epsilon_1$ et $\frac{s_a(\omega^*)}{n} < p - \epsilon_1$ et $\frac{s_a(\omega_i)}{n} > \frac{s_a(\omega^*)}{n})$

Comme $P(\frac{s_a(\omega^*)}{n} < p - \epsilon_1$ et $\frac{s_a(\omega_i)}{n} > \frac{s_a(\omega^*)}{n}) \leq P(\frac{s_a(\omega^*)}{n} < p - \epsilon_1)$ et $P(\frac{s_a(\omega_i)}{n} > q_i + \epsilon_1$ et $\frac{s_a(\omega_i)}{n} > \frac{s_a(\omega^*)}{n}) \leq P(\frac{s_a(\omega_i)}{n} > q_i + \epsilon_1)$, on obtient :

$P(\frac{s_a(\omega_i)}{n} > \frac{s_a(\omega^*)}{n}) \leq$

$$P(\frac{s_a(\omega^*)}{n} < p - \epsilon_1) + P(\frac{s_a(\omega_i^*)}{n} > q_i + \epsilon_1).$$

Finalement, grâce aux assertions (4) et (5), on obtient :

$$P(\frac{s_a(\omega_i)}{n} \geq \frac{s_a(\omega^*)}{n}) \xrightarrow[n\to\infty]{} 0,$$

ou, de façon équivalente

$$P(s_a(\omega_i) \geq s_a(\omega^*)) \xrightarrow[n\to\infty]{} 0.$$

□

Ce théorème montre immédiatement l'aptitude de la méthode de vote dite par approbation [6], à identifier le monde réel (à la limite et en probabilité). Avec cette méthode, les agents sont autorisés à voter pour un nombre quelconque de mondes, et ensuite on choisit comme résultat les mondes avec les scores les plus élevés :

**Définition 4 (Approbation)** *Etant donné un profil de bases $E$, la règle de vote* par approbation $av$ *est définie par :*

$$av(E) = \{\omega \mid \forall \omega' \in \mathcal{W} \;\; s_a(\omega) \geq s_a(\omega')\}.$$

## 5 Identifier le monde réel par fusion de croyances

L'aptitude d'un opérateur de fusion à identifier le monde réel peut être modélisée par un nouveau postulat, appelé postulat **T***ruth* **T***racking* :

**TT** Soit $E$ un profil issu d'un ensemble homogène de $n$ agents indépendants et R4-fiables. Soit $\omega^\star$ le monde réel.

$$P([\Delta(E)] = \{\omega^\star\}) \xrightarrow[n\to\infty]{} 1.$$

Ce postulat est vérifié par un opérateur de fusion si l'opérateur est capable d'identifier le monde réel (à la limite et en probabilité) en s'appuyant sur un ensemble homogène d'agents indépendants qui sont plus R3-fiables que incompétents. Clairement, ses conditions d'application sont fortes puisqu'il n'est pas faciles de pouvoir garantir en pratique que l'on pourra disposer des croyances d'un ensemble homogène et suffisamment arbitrairement grand d'agents indépendants et R4-fiables. Il en est d'autant plus intéressant puisqu'il assure que les opérateurs ne le vérifiant pas ne sont pas de bons candidats pour identifier le monde réel.

On peut à présent étudier le comportement par rapport à ce postulat de certains opérateurs de fusion de croyances bien connus. Nous rappelons d'abord la définition des opérateurs de fusion basés sur une distance [12].

**Définition 5 (fusion à base de distance)** *Soit $d$ une pseudo-distance entre interprétations et soit $f$ une fonction d'agrégation. L'opérateur de fusion $\Delta^{d,f}(E)$ est défini par :* $[\Delta^{d,f}_\mu(E)] = \min([\mu], \leq_E)$
*où le pré-ordre $\leq_E$ sur $\mathcal{W}$ induit par $E$ est défini par :*
– $\omega \leq_E \omega'$ *si et seulement si* $d(\omega, E) \leq d(\omega', E)$, *avec*
– $d(\omega, E) = f_{K\in E}(d(\omega, K))$, *où*
– $d(\omega, K) = min_{\omega'\models K} d(\omega, \omega')$.

On notera $d_D$ la distance drastique ($d_D(\omega, \omega') = 0$ si $\omega = \omega'$ et 1 sinon), et $d_H$ la distance de Hamming [11].

Le résultat suivant est une conséquence directe du théorème 3 :

**Proposition 4** $\Delta^{d_D,\Sigma}$ *satisfait* **TT**.

Puisque $\Delta^{d_D,\Sigma}$ est un opérateur de fusion contrainte [12], cette proposition montre qu'il n'y a pas de conflit entre **TT** et les postulats **(IC0-IC8)** qui caractérisent les opérateurs de fusion contrainte. Néanmoins, tous les opérateurs de fusion contrainte ne satisfont pas **TT**. Par exemple :

**Proposition 5** $\Delta^{d_H,Gmax}$ *ne satisfait pas* **TT**.

**Preuve:** On considère quatre mondes possibles $\mathcal{W} = \{00, 01, 10, 11\}$, et on suppose que le monde réel est $\omega^\star = 11$. On considère la probabilité suivante $p$ sur les mondes possibles, partagées par tous les agents (hypothèse d'homogénéité) : $p(11 \models K) = a > 0.5$, $p(00 \models K) = b$, $p(01 \models K) = c$, $p(10 \models K) = 0$, avec $a + b + c = 1$, $b > 0$, et $c > 0$. Par construction, pour chaque monde $\omega$ de $\mathcal{W}$ sauf 01, la probabilité qu'un profil $E$ de $n$ agents indépendants contienne une base $K$ telle que $d_H(\omega, K) = 2$ tend vers 1 lorsque $n$ tend vers l'infini ; par ailleurs, la probabilité qu'un profil $E$ de $n$ agents indépendants contienne une base $K$ telle que $d_H(01, K) = 2$ est toujours égale à 0 (en effet on a toujours $d_H(01, K) < 2$). En conséquence, $P([\Delta^{d_H,Gmax}(E)] = \{01\}) \xrightarrow[n\to\infty]{} 1$, ce qui contredit **TT**. □

## 6 Résultats expérimentaux

Les résultats obtenus dans les sections précédentes au sujet de l'identification du monde réel $\omega^\star$ sont des résultats *à la limite*. Aucun d'entre eux ne donne d'information sur l'identification du monde réel d'un point de vue pratique, comme par exemple un majorant sur le nombre de bases à partir de laquelle l'identification est atteinte avec une probabilité élevée. Or, clairement, si le nombre de bases nécessaires à cette identification est très élevé, le problème de l'identification du monde réel, lorsqu'il est envisageable a priori (e.g. lorsque les agents considérés sont des experts indépendants) reste sans résolution pratique possible.

Afin d'approfondir cette question, nous avons conduit des expérimentations en utilisant $\Delta^{d_D,\Sigma}$, un opérateur de fusion contrainte qui satisfait **TT** et qui est simple à implémenter. Le protocol expérimental utilisé a été le suivant. Nous avons étudié la vitesse de convergence vers le monde réel en utilisant $\Delta^{d_D,\Sigma}$, en fonction de la R3-fiabilité des agents $p$ et de leur incompétence $q$ (supposée identique pour tous les agents). Nous avons considéré des ensembles d'interprétations de tailles variables (jusqu'à $2^{15}$), nous avons fixé le monde réel $\omega^\star$ comme l'interprétation associant chaque variable propositionnelle à 0, et nous avons engendré des profils $E$ à partir de $n$ agents indépendants, avec une R3-fiabilité $p$ et une incompétence $q$, pour différentes valeurs de $n$. Pour chaque valeur $n$, nous avons construit 1000 profils $E$. Pour chaque $E$, nous avons calculé $\Delta^{d_D,\Sigma}(E)$ et testé si $[\Delta^{d_D,\Sigma}(E]) = \{\omega^\star\}$. La proportion des 1000 profils pour lesquels $[\Delta^{d_D,\Sigma}(E]) = \{\omega^\star\}$ donne une estimation de la probabilité de succès de l'identification du monde réel.

La figure 1 donne la probabilité que $[\Delta^{d_D,\Sigma}(E]) = \{\omega^\star\}$ étant donné le nombre $n$ d'agents, lorsque $p = 0.3$, et $|\mathcal{W}| = 2^7$ mondes, pour différentes valeurs $q$, La figure 2 présente des expérimentations similaires, mais pour $p = 0.9$.

Un point intéressant est à observer sur la figure 1 : même si $p = 0.3$ est relativement faible (ici les agents considérés ne sont pas R3-fiables), la vitesse de convergence est élevée : pour obtenir $[\Delta^{d_D,\Sigma}(E)] = \{\omega^\star\}$ avec une probabilité supérieure à 90%, 1250 agents sont nécessaires pour une incompétence $q = 0.25$, 300 agents sont nécessaires pour $q = 0.2$ et seulement 60 agents

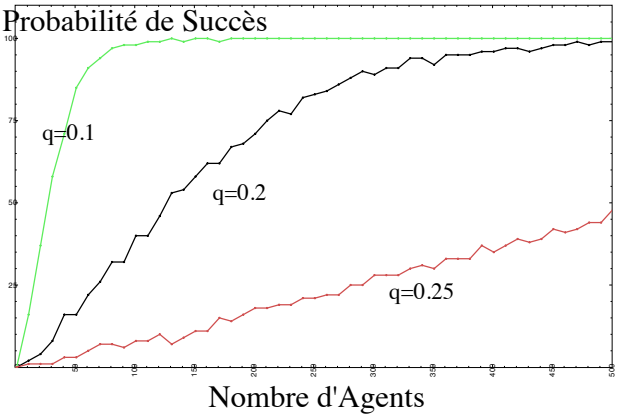



FIG. 1 – Vitesse de convergence (7 variables, p=0.3)

sont nécessaires pour $q = 0.1$.

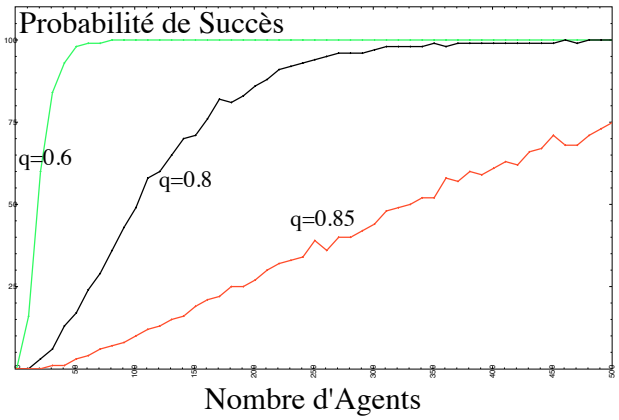



FIG. 2 – Vitesse de convergence (7 variables, p=0.9)

Ces résultats sont proches de ceux donnés à la figure 2 pour $p = 0.9$. Les valeurs correspondantes sont 800 pour $q = 0.85$, 230 pour $q = 0.8$, et 40 pour $q = 0.6$.

Empiriquement, il apparaît que le "niveau de R4-fiabilité" des agents, i.e., l'écart $p - q$ a plus d'impact sur la vitesse de convergence de l'identification du monde réel en utilisant $\Delta^{d_D,\Sigma}$ que le fait que les valeurs $p$ et $q$ sont plutôt proches de 1 ou de 0.

La figure 3 donne la probabilité de succès de l'identification du monde réel étant donné le nombre de variables propositionnelles (et donc le nombre d'interprétations) avec $p = 0.70$ et $q = 0.40$.

Il n'est pas étonnant que la complexité de d'identifier le monde réel augmente avec le nombre de mondes possibles. Cependant, le

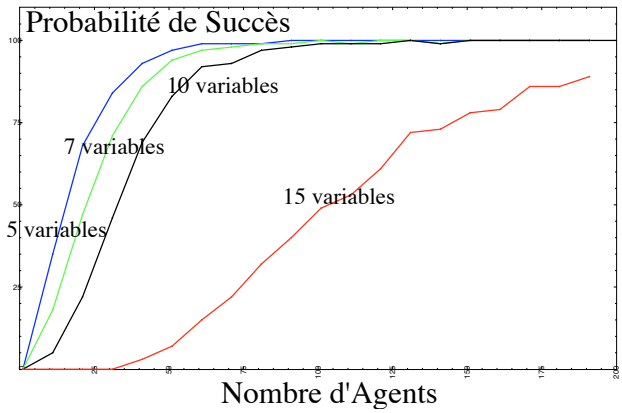



FIG. 3 – Vitesse de convergence (p=0.7, q =0.4)

nombre d'agents qui doivent être considérés pour atteindre l'objectif avec une probabilité élevée n'est pas si grand, si on le met en balance avec le nombre d'interprétations possibles. Par exemple, on peut observer sur la figure 3 que, lorsque 10 variables sont considérées, moins de 50 agents sont suffisants pour assurer que $[\Delta^{d_D,\Sigma}(E)] = \{\omega^\star\}$ avec une probabilité supérieure à $90\%$, en dépit du fait qu'un unique monde doit être discriminé parmi 1024 et que la R3-fiabilité et la compétence des agents ne sont pas très élevées. Tous ces résultats montrent la faisabilité pratique de la recherche de la vérité en utilisant $\Delta^{d_D,\Sigma}$.

## 7 Conclusion

Dans ce travail, nous avons considéré la fusion de croyances selon un point de vue original que nous avons appelé point de vue épistémique. Cette approche permet d'évaluer la capacité des opérateurs de fusion de croyances à trouver le monde réel. Nous avons démontré une généralisation du théorème du jury de Condorcet lorsque les croyances des agents sont incertaines. Nous avons également défini un postulat correspondant à l'identifiabilité à la limite du monde réel pour la fusion de croyances et montré que certains opérateurs de fusion de croyances satisfont ce postulat. Finalement, nous avons étudié la vitesse de convergence offerte par $\Delta^{d_D,\Sigma}$.

Le problème de l'identification du monde réel a aussi été étudié dans le cadre plus spécifique de l'agrégation de jugement [5]. Dans [18] les auteurs étudient les performances d'un opérateur $\Delta^{d_H,\Sigma}$ dans cet optique, et montrent que ses performances sont bonnes (typiquement meilleures que celles d'autres procédures d'agrégation de jugement) pour des agents assez fiables.

Dans nos travaux futurs, nous pensons étudier le problème de l'identification du monde réel pour d'autres opérateurs de fusion de croyances, comme ceux basés sur la distance de Hamming. Nos premières expériences suggèrent que $\Delta^{d_H,\Sigma}$ satisfait également **TT**.

## Remerciements

Les auteurs remercient les relecteurs anonymes pour la qualité de leur travail. Ils ont bénéficier du support du projet ANR PHAC (ANR-05-BLAN-0384).

## Références

[1] K. J. Arrow. *Social Choice and Individual Values*. Wiley, New York, second edition, 1963.

[2] C. Baral, S. Kraus, and J. Minker. Combining multiple knowledge bases. *IEEE Transactions on Knowledge and Data Engineering*, 3(2) :208–220, 1991.

[3] S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Possibilistic merging and distance-based fusion of propositional information. *Annals of Mathematics and Artificial Intelligence*, 34(1–3) :217–252, 2002.

[4] S. Benferhat, S. Lagrue, and J. Rossit. An egalitarist fusion of incommensurable ranked belief bases under constraints. In *Proceedings of AAAI'07*, pages 367–372, 2007.

[5] L. Bovens and W. Rabinowicz. Democratic answers to complex questions. *Synthese*, 150 :131–153, 2006.

[6] S. J. Brams and P. C. Fishburn. *Approval voting*. Springer Verlag, 1983.

[7] Marquis de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Imprimerie Royale, Paris, 1785.

[8] D. Estlund. Opinion leaders, independence and condorcet's jury theorem. *Theory and Decision*, 36(2) :131–162, January 1994.

[9] P. Everaere, S. Konieczny, and P. Marquis. Quota and gmin merging operators. In *Proceedings of IJCAI'05*, pages 424–429, 2005.

[10] O. Guillermo, B. Grofman, and S. Feld. Proving a distribution-free generalization of the condorcet jury theorem. *Mathematical Social Sciences*, 17(1) :1–16, February 1989.

[11] S. Konieczny, J. Lang, and P. Marquis. $DA^2$ merging operators. *Artificial Intelligence*, 157 :49–79, 2004.

[12] S. Konieczny and R. Pino Pérez. Merging information under constraints : a logical framework. *Journal of Logic and Computation*, 12(5) :773–808, 2002.

[13] S. Konieczny and R. Pino Pérez. On the frontier between arbitration and majority. In *Proceedings of KR'02*, pages 109–118, 2002.

[14] P. Liberatore and M. Schaerf. Arbitration (or how to merge knowledge bases). *IEEE Transactions on Knowledge and Data Engineering*, 10(1) :76–90, 1998.

[15] C. List and R. Goodin. Epistemic democracy : Generalizing the condorcet jury theorem. *Journal of Political Philosophy*, 9(3) :277–306, 2001.

[16] T. Meyer. On the semantics of combination operations. *Journal of Applied Non-Classical Logics*, 11(1-2) :59–84, 2001.

[17] Judea Pearl. *Causality : models, reasoning, and inference*. Cambridge University Press, 2000.

[18] G. Pigozzi and S. Hartmann. Judgment aggregation and the problem of truth-tracking. In *Proceedings of TARK'07*, pages 248–252, 2007.

[19] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Aggregating partially ordered preferences : impossibility and possibility results. In *Proceedings of TARK'05*, pages 193–206, 2005.

[20] P. Z. Revesz. On the semantics of arbitration. *International Journal of Algebra and Computation*, 7(2) :133–160, 1997.

# Une logique modale pour raisonner sur la cohérence et la complétude de réglementations

C. Garion* S. Roussel† L. Cholvy†
garion@isae.fr roussel@cert.fr cholvy@cert.fr

†ONERA-Toulouse DTIM
BP 54025
31055 Toulouse Cedex 4 – FRANCE

* Université de Toulouse
ISAE/DMIA
BP 54032
31055 Toulouse Cedex 4 – FRANCE

**Abstract:**
Nous nous intéressons dans ce papier aux réglementations qui peuvent exister dans des systèmes multi-agents pour réguler les comportements des agents. Plus précisément, nous travaillons sur deux propriétés des réglementations, la cohérence et la complétude. Après avoir donné une définition de ces deux notions, nous proposons un cadre permettant de compléter de façon cohérente une réglementation incomplète. Nous considérons dans l'article que les réglementations sont exprimées dans une logique déontique du premier ordre.

**Mots-clés :** logique déontique, raisonnement par défaut, complétude, cohérence, réglementations

**Keywords:** deontic logic, default reasoning, completness, consistency, regulations

## 1 Introduction

Dans une société d'agents, une réglementation est un ensemble de phrases, ou normes, qui régule le comportement des agents en exprimant ce qui est obligatoire, permis et interdit et ceci sous quelles conditions.

Par exemple, une réglementation qui s'applique dans la plupart des pays de l'UE est : « *il est interdit de fumer dans les lieux publics, sauf dans des endroits spécifiques. Dans de tels endroits, il est permis de fumer* ». Un autre exemple de réglementation est celle qui définit les permissions, interdictions (et parfois les obligations) des différents utilisateurs d'un ordinateur pour lire, écrire ou exécuter des fichiers.

Les réglementations sont des moyens de réguler les comportements des agents de telle façon qu'ils puissent coexister. Mais pour être efficace, elles doivent être *cohérentes* et, dans la plupart des cas, *complètes*.

La cohérence est une propriété des réglementations sur laquelle de nombreux travaux ont déjà porté. Par exemple, dans le cas de politiques de confidentialité, la cohérence permet d'éviter les cas où l'utilisateur a à la fois la permission et l'interdiction de savoir quelque chose [2]. Plus généralement, d'après [4] qui étudie la cohérence de réglementations générales, une réglementation est cohérente si il n'y a pas de situation possible qui conduit un agent à des *contradictions normatives* ou des *dilemmes* également appelés *conflits contradictoires* dans [20] (un comportement donné est à la fois prescrit et non prescrit, ou interdit et non interdit). Suivant cette définition, la cohérence de politiques de sécurité a été étudiée dans [5].

La complétude des réglementations, quant à elle, a été beaucoup moins étudiée. [2] propose une définition de complétude entre deux politiques de confidentialité (pour chaque information, l'utilisateur doit soit avoir la permission de la posséder, soit l'interdiction de la posséder), définition qui a été adaptée dans [8] pour les politiques de sécurité multi-niveaux.

Plus récemment, nous avons étudié la notion de complétude pour des réglementations particulières, les politiques d'échange d'informations dans un système multi-agents [6]. Une définition de l'incomplétude de telles politiques a été proposée et une méthode de raisonnement avec de telles politiques a été définie. L'approche prise dans ce travail était prometteuse et nous l'avons étendue pour des réglementations plus générales dans [7]. Le langage formel utilisé dans ces deux papiers est la logique du premier ordre (FOL) suivant les idées développées dans [4]. En particulier, les notions déontiques (obligation, permission, interdiction) sont représentées en utilisant des symboles de prédicats. Cela amène à effectuer une partition compliquée du langage entre les symboles de prédicats déontiques et les symboles de prédicats représentant les propriétés des objets, ce qui est critiquable. De plus, les notions déontiques sont

classiquement représentées en utilisant une logique modale [19, 14]. C'est pourquoi nous utilisons dans ce papier une logique modale du premier ordre [12] pour exprimer les réglementations d'une manière plus élégante. Notre objectif est donc de reformuler le travail de [7] dans un cadre modal du premier ordre.

Le papier est organisé comme suit. La section 2 présente le formalisme logique utilisé pour exprimer des réglementations, les définitions de cohérence et de complétude de réglementations. La section 3 s'intéresse au problème du raisonnement avec une réglementation incomplète. En suivant les idées qui ont amené à la logique des défauts [17] pour le raisonnement par défaut, nous présentons des défauts qui peuvent être utilisés dans le but de compléter des réglementations incomplètes. Dans la section 4, nous présentons un exemple particulier de réglementation, une politique d'échange d'informations. Enfin, la section 5 est dévolue à une discussion et une présentation des futures extensions de ce travail.

## 2 Réglementations

Le formalisme de base utilisé pour représenter des réglementations est SDL (*Standard Deontic Logic*), une logique modale propositionnelle [3]. En suivant les techniques présentées dans [12], nous étendons SDL en FOSDL (*First-Order Standard Deontic Logic*) pour pouvoir exprimer des réglementations plus complexes impliquant plusieurs agents.

### 2.1 Langage

L'alphabet de FOSDL est composé des ensembles de symboles non logiques suivants : un ensemble $\mathcal{P}$ de symboles de prédicats, un ensemble $\mathcal{F}$ de symboles de fonctions et une modalité $O$ représentant l'obligation. L'ensemble des fonctions d'arité 0 est appelé *ensemble des constantes* et est noté $\mathcal{C}$. Nous définissons également les symboles logiques suivants : un ensemble $\mathcal{V}$ de symboles de variables, $\neg$, $\vee$, $\forall$, ( et ). On appelle *terme* une variable ou l'application d'un symbole de fonction à un ou plusieurs termes.

Nous utiliserons des lettres romaines majuscules comme symboles de prédicats, des lettres romaines minuscules comme symboles de fonctions et $\{x_1, \ldots, x_i, \ldots\}$ comme symboles de variables.

**Définition 1** *Les formules de FOSDL sont définies récursivement comme suit :*

- *si $t_1, \ldots, t_n$ sont des termes et $P$ un symbole de prédicat d'arité $n$, alors $P(t_1, \ldots, t_n)$ est une formule de FOSDL.*
- *si $\varphi$ est une formule de FOSDL, alors $O\varphi$ est une formule de FOSDL.*
- *si $\psi_1$ et $\psi_2$ sont des formules de FOSDL et $x_1$ un symbole de variable, alors $\neg\psi_1$, $\psi_1 \vee \psi_2$, $\forall x_1\ \psi_1$ sont des formules de FOSDL.*

Si $\psi_1$, $\psi_2$ et $\psi_3$ sont des formules de FOSDL et $x_1$ est un symbole de variable, nous définissons également les abréviations suivantes : $\psi_1 \wedge \psi_2 \equiv \neg(\neg\psi_1 \vee \neg\psi_2)$, $\psi_1 \otimes \psi_2 \otimes \psi_3 \equiv (\psi_1 \wedge \neg\psi_2 \wedge \neg\psi_3) \vee (\neg\psi_1 \wedge \psi_2 \wedge \neg\psi_3) \vee (\neg\psi_1 \wedge \neg\psi_2 \wedge \psi_3)$, $\psi_1 \rightarrow \psi_2 \equiv \neg\psi_1 \vee \psi_2$, $\psi_1 \leftrightarrow \psi_2 \equiv (\neg\psi_1 \vee \psi_2) \wedge (\psi_1 \vee \neg\psi_2)$, $\exists x_1\ \psi_1 \equiv \neg\forall x_1\ \neg\psi_1$.

Les modalités pour la permission, notée $P$, et l'interdiction, notée $F$ sont définies à partir de $O$ comme suit :

$$F\varphi \equiv O\neg\varphi$$
$$P\varphi \equiv \neg O\varphi \wedge \neg O\neg\varphi$$

Notons que notre définition de la permission ne correspond pas à la définition usuelle de SDL. Pour cette dernière, quelque chose est permis si sa négation n'est pas obligatoire. Cependant, il a été montré par des juristes [13] que les cas où la permission est bilatérale (permission de faire et permission de ne pas faire) sont les seuls valides. Si une permission n'est pas bilatérale, alors elle implique l'obligation[1]. Notre définition de la permission bilatérale correspond à la notion d'*optionalité*[15] (quelque chose est optionnel ssi ni lui ni sa négation ne sont obligatoires).

Une formule de FOSDL sans modalités est dite *objective*. Les termes et formules de FOSDL sans symboles de variables sont dits *de base*. L'ensemble des termes de base de FOSDL est appelé univers de Herbrand $HU$. Une formule de FOSDL sans connecteur $\vee$, $\wedge$, $\otimes$, $\rightarrow$ ou $\leftrightarrow$ est appelée un *littéral*.

Enfin, nous appelons une *substitution de base* toute fonction $\chi : \mathcal{V} \rightarrow HU$. Si $\varphi(x)$ est une formule de FOSDL avec une variable libre $x$, $\varphi(\chi(x))$ est la formule $\varphi$ dans laquelle les occurrences de $x$ ont été remplacées par $\chi(x)$.

[1] Par exemple, si fumer est autorisé, alors ne pas fumer est également autorisé. Sinon, cela signifierait que fumer est obligatoire.

## 2.2 Sémantique

La sémantique des logiques modales propositionnelles est classiquement définie en utilisant des modèles de Kripke. Les modèles sont définis par un *cadre* $\langle \mathcal{W}, \mathcal{R} \rangle$, où $\mathcal{W}$ est un ensemble de mondes et $\mathcal{R}$ une relation d'accessibilité entre les mondes, et une relation $\Vdash$ entre les mondes et les lettres propositionnelles. Dans le cas du premier ordre, nous définissons des modèles en utilisant un cadre *augmenté* et une interprétation du premier ordre au lieu de $\Vdash$.

La sémantique des langages du premier ordre est fondée sur un ensemble de symboles (les *objets du discours*), appelé le *domaine*. Le domaine représente les objets sur lesquels les prédicats vont porter par opposition aux termes qui sont des notions purement mathématiques. Dans le cas d'une logique modale du premier ordre, nous devons choisir entre des cadres augmentés à domaine constant ou à domaine variable. Dans le premier cas, le domaine est fixé pour tous les mondes de $\mathcal{W}$, dans le second cas, chaque monde de $\mathcal{W}$ peut avoir son propre domaine. Nous choisissons ici un domaine constant. Comme les normes que nous étudions ne concernent que des éléments fixes, ce choix est assez intuitif[2].

**Définition 2** *Soient $\mathcal{W}$ un ensemble de mondes, $\mathcal{R}_O$ une relation sur $\mathcal{W}^2$ et $\mathcal{D}$ un ensemble non vide de symboles représentant le domaine, alors $\langle \mathcal{W}, \mathcal{R}_O, \mathcal{D} \rangle$ est appelé un cadre.*

Pour définir un modèle, nous devons définir une interprétation du premier ordre, ce qui est fait classiquement.

**Définition 3** *Une interprétation $\mathcal{I}$ dans un cadre $\langle \mathcal{W}, \mathcal{R}_O, \mathcal{D} \rangle$ est une application telle que :*

- *pour tout symbole de fonction $n$-aire $f$ dans $\mathcal{F}$ et tout monde $w \in \mathcal{W}$, $\mathcal{I}(f, w)$ est une fonction $\mathcal{D}^n \rightarrow \mathcal{D}$* indépendante du monde $w$ *;*
- *pour tout symbole de prédicat $n$-aire $P$ dans $\mathcal{P}$ et tout monde $w \in \mathcal{W}$, $\mathcal{I}(P, w)$ est une relation sur $\mathcal{D}^n$.*

[2]Notons toutefois que les domaines variables peuvent être utiles. Par exemple, dans une logique modale du premier ordre doxastique, un agent peut apprendre l'existence d'un objet particulier, ou un nouvel objet peut apparaître.

Notons que nous imposons une condition particulière sur l'interprétation des fonctions : l'interprétation d'une fonction $f$ est la même dans tous les mondes possibles (c'est possible grâce à l'utilisation d'un domaine constant). Cette restriction nous permet d'éviter des détails techniques compliqués[3], comme par exemple l'utilisation de l'abstraction pour les prédicats. Voir [12] pour plus de détails.

**Définition 4** *Un modèle $\mathcal{M}$ est une structure $\langle \mathcal{W}, \mathcal{R}_O, \mathcal{D}, \mathcal{I} \rangle$ où $\langle \mathcal{W}, \mathcal{R}_O, \mathcal{D} \rangle$ est un cadre et $\mathcal{I}$ une interprétation sur $\langle \mathcal{W}, \mathcal{R}_O, \mathcal{D} \rangle$.*

Enfin, nous utilisons une classe de cadres qui modélisent le comportement de l'opérateur $O$ en contraignant la relation d'accessibilité $\mathcal{R}_O$.

**Définition 5** *Un modèle de FOSDL est un modèle $\langle \mathcal{W}, \mathcal{R}_O, \mathcal{D}, \mathcal{I} \rangle$ tel que $\mathcal{R}_O$ est sérielle.*

Pour pouvoir définir une relation de satisfaisabilité entre modèles et formules, nous devons définir la notion de *valuation* qui associe les variables du langage aux éléments de $\mathcal{D}$ :

**Définition 6** *Soit $\mathcal{D}$ un domaine. Une valuation sur $\mathcal{D}$ est une fonction complète $\mathcal{V} \rightarrow \mathcal{D}$. Une valuation $\sigma'$ est une valuation $x$-variante d'une valuation $\sigma$ si $\sigma$ et $\sigma'$ sont identiques sauf en $x$.*

*Soient $t$ un terme et $\mathcal{V}(t)$ l'ensemble des variables de $t$, $\sigma(t)$ est le terme $t$ dans lequel chaque $x_i$ de $\mathcal{V}(t)$ a été remplacé par $\sigma(x_i)$.*

La relation de satisfaisabilité $\models$ est définie comme suit :

**Définition 7** *Soient $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}_O, \mathcal{D}, \mathcal{I} \rangle$ un modèle de FOSDL, $w$ un monde de $\mathcal{W}$ et $\sigma$ une valuation sur $\mathcal{D}$. Alors :*

- *si $P$ est un symbole de prédicat $n$-aire et $t_1, \ldots, t_n$ sont des termes, alors $\mathcal{M}, w \models_\sigma P(t_1, \ldots, t_n)$ ssi $\langle \mathcal{I}(\sigma(t_1), w), \ldots, \mathcal{I}(\sigma(t_n), w) \rangle \in \mathcal{I}(P, w)$.*
- *si $\psi$ est une formule de FOSDL, alors $\mathcal{M}, w \models_\sigma \neg\psi$ ssi $\mathcal{M}, w \not\models_\sigma \psi$.*

[3]Le problème principal est d'être capable de caractériser la signification d'une formule telle que $OF(c)$ où $c$ est une constante : est que cela signifie que « il est obligatoire que l'objet représenté par *c dans le monde courant* a la propriété $F$ » ou « il est obligatoire que *dans chaque monde*, l'objet représenté par $c$ a la propriété $F$ ».

– *if $\psi_1$ et $\psi_2$ sont des formules de FOSDL, alors $\mathcal{M}, w \models_\sigma \psi_1 \vee \psi_2$ ssi $\mathcal{M}, w \models_\sigma \psi_1$ or $\mathcal{M}, w \models_\sigma \psi_2$.*
– *if $O\varphi$ est une formule de, $\mathcal{M}, w \models_\sigma O\varphi$ ssi pour tout $v \in \mathcal{W}$ tel que $w\mathcal{R}_O v$, $\mathcal{M}, v \models_\sigma \varphi$.*
– *si $\psi$ est une formule de FOSDL, $\mathcal{M}, w \models_\sigma \forall x\, \psi$ ssi pour toute valuation $\sigma'$ $x$-variant de $\sigma$, $\mathcal{M}, w \models_{\sigma'} \psi$.*

Soit $\psi$ une formule de FOSDL. Si pour toute valuation $\sigma$ $\mathcal{M}, w \models_\sigma \psi$, on notera $\mathcal{M}, w \models \psi$. Si $\mathcal{M}, w \models \psi$ pour tout $w$ de $\mathcal{W}$, on notera $\mathcal{M} \models \psi$. Enfin, si $\mathcal{M} \models \psi$ pour tout modèle de FOSDL $\mathcal{M}$, on notera $\models \psi$.

### 2.3 Axiomatique

Nous allons maintenant définir un système axiomatique pour FOSDL en suivant l'approche proposée dans [12]. Dans ce qui suit, $\varphi(x)$ est une formule dans laquelle la variable $x$ peut avoir des occurrences libres. On dira qu'une variable libre $y$ est *substituable* à $x$ dans $\varphi(x)$ si il n'y a pas d'occurrence de $x$ dans $\varphi(x)$ dans la portée de $\forall y$ dans $\varphi(x)$.

**Définition 8 (Axiomes)** *Les formules de la forme suivante sont des axiomes :*

| | |
|---|---|
| *(Taut)* | *les tautologies de la logique prop.* |
| *(KO)* | $O(\varphi \rightarrow \psi) \rightarrow (O\varphi \rightarrow O\psi)$ |
| *(DO)* | $O\varphi \rightarrow \neg O \neg \varphi$ |
| *(Bar1)* | $O(\forall x\, \varphi) \rightarrow \forall x\, O\varphi$ |
| *(Bar2)* | $\forall x\, O\varphi \rightarrow O(\forall x\, \varphi)$ |

**Définition 9 (Règles d'inférence)**

*(MP)* $\dfrac{\varphi \quad \varphi \rightarrow \psi}{\psi}$

*(Gen)* $\dfrac{\varphi}{\forall x\, \varphi}$

*(NO)* $\dfrac{\varphi}{O\varphi}$

**Proposition 1 (Validité et complétude)** *Le système précédent est valide et complète par rapport à la sémantique de FOSDL.*

La preuve est donnée dans [12].

On définie une *preuve* de $\varphi$ à partir de l'ensemble de formules $\Sigma$ comme étant une séquence de formules telle que chacune d'entre elles est un axiome, une formule de $\Sigma$, ou produite par l'application d'une règle d'inférence sur une formule précédente.

Dans ce qui suit, $\bot$ représente toute formule qui est une contradiction et $\top$ représente toute formule qui est une tautologie.

### 2.4 Modélisation de réglementations

Dans cette section, nous définissons la notion de réglementation. Dans un premier temps, nous définissons la notion de règle, qui est le composant de base d'une réglementation. Dans cette définition, les règles ont une forme générale, en particulier, elles peuvent être conditionnelles.

**Définition 10** *Une règle est une formule de FOSDL de la forme $\forall \overrightarrow{x}\ l_1 \vee \ldots \vee l_n$ avec $n \geq 1$ telle que :*

1. *$l_n$ est de la forme $O\varphi$ ou $\neg O\varphi$ où $\varphi$ est un littéral objectif*
2. *$\forall i \in \{1, \ldots, n-1\}$, $l_i$ est un littéral objectif*
3. *si $x$ est une variable dans $l_n$, alors $\exists i \in \{1, \ldots, n-1\}$ tel que $l_i$ est un littéral négatif et contient la variable $x$*
4. *$\forall \overrightarrow{x}$ représente $\forall x_1 \ldots \forall x_m$ où $\{x_1, \ldots, x_m\}$ est l'ensemble des variables libres apparaissant dans $l_1 \wedge \ldots \wedge l_{n-1}$.*

Dans cette définition, les contraintes (1) et (2) permettent d'exprimer des règles de la forme « si telle condition est vraie alors quelque chose est obligatoire (resp. permis, interdit) ». La contrainte (3) restreint les règles aux formules à champ restreint[4]. Enfin, les règles sont des *phrases*, i.e. des formules fermées, comme exprimé par la contrainte (4).

Remarquons également que nous restreignons dans la définition des règles la formule qui peut être obligatoire : seuls les littéraux objectifs peuvent être obligatoires ou pas.

On écrira $\forall \overrightarrow{x}\ l_1 \vee \ldots \vee l_{n-1} \vee P\varphi$ comme un raccourci d'écriture pour les deux règles $\{\forall \overrightarrow{x}\ l_1 \vee \ldots \vee l_{n-1} \vee \neg O\varphi, \forall \overrightarrow{x}\ l_1 \vee \ldots \vee l_{n-1} \vee \neg O\neg\varphi\}$.

[4] Les formules à champ restreint sont un fragment décidable des formules domaine-indépendant dont on a prouvé qu'elles étaient les seules formules du premier ordre ayant une signification en modélisation [9]. Remarquons en particulier que par définition du langage de FOSDL, toutes les variables apparaissant dans $l_n$ sont libres dans $l_n$.

**Définition 11** *Une réglementation est un ensemble de règles.*

Considérons un exemple qui illustrera les notions développées dans les sections 2 et 3.

**Exemple 1** *Nous considérons une réglementation qui régule le comportement d'un conducteur devant un feu tricolore.*

*Le langage utilisé est défini comme suit :*
- *green, orange, red, car, truck, bike, $A$ et $T$ sont des fonctions d'arité nulle, i.e. des constantes.*
- *$x$, $y$, $z$, $i$ et $t$ sont des variables.*
- *$Driver(.)$ est un symbole de prédicat qui indique qu'un terme est un conducteur.*
- *$TL(.)$ est un symbole de prédicat qui indique qu'un terme est un feu tricolore.*
- *$Color(.,.)$ est un symbole de prédicat qui prend pour paramètres un feu tricolore et une couleur et indique la couleur du feu.*
- *$Vehicle(.,.)$ est un symbole de prédicat qui prend pour paramètre un conducteur et le type de véhicule qu'il conduit.*
- *$IFO(.,.)$ est un symbole de prédicat qui prend pour paramètre un conducteur et un feu tricolore et indique que le véhicule conduit par le conducteur est devant un feu.*
- *$Stop(.,.)$ est un symbole de prédicat qui prend un conducteur et un feu tricolore en paramètres et indique que l'agent stoppe devant le feu tricolore.*

*Considérons maintenant les trois règles $(r_0)$ : « Quand un conducteur est devant un feu tricolore qui est rouge, il doit s'arrêter » $(r_1)$ : « Quand un conducteur est devant un feu tricolore qui est orange, il peut s'arrêter » $(r_2)$ : « Quand un conducteur est devant un feu tricolore qui est vert, il ne doit pas s'arrêter ». Ces règles sont modélisées par :*

$$
\begin{array}{ll}
(r_0) & \forall x \forall t\ Driver(x) \wedge TL(t) \wedge \\
& Vehicle(x, car) \wedge Color(t, red) \\
& \wedge IFO(x,t) \rightarrow OStop(x,t) \\
(r_1) & \forall x \forall t\ Driver(x) \wedge TL(t) \wedge \\
& Vehicle(x, car) \wedge Color(t, orange) \\
& \wedge IFO(x,t) \rightarrow PStop(x,t) \\
(r_2) & \forall x \forall t\ Driver(x) \wedge TL(t) \wedge \\
& Vehicle(x, car) \wedge Color(t, green) \\
& \wedge IFO(x,t) \rightarrow FStop(x,t)
\end{array}
$$

## 2.5 Cohérence de réglementations

Nous définissons maintenant une première notion utile pour les réglementations, la *cohérence*. Intuitivement, nous dirons qu'une réglementation est cohérente ssi nous ne pouvons pas dériver de la réglementation d'incohérences comme $OStop(x,t) \wedge FStop(x,t)$ en utilisant le système défini dans 2.3. La cohérence d'une réglementation est évaluée sous des *contraintes d'intégrité*, i.e. un ensemble de formules objectives closes qui représentent par exemple des contraintes physiques ou des contraintes du domaines. Dans ce qui suit, nous noterons un tel ensemble de contraintes d'intégrité $IC$.

Nous définissons d'abord la cohérence d'une réglementation dans un *état du monde* particulier. Intuitivement, les états du monde sont des représentations syntaxiques des interprétations du premier ordre. Ils peuvent également être assimilés à des modèles de Herbrand classiques.

**Définition 12 (état du monde)** *Un état du monde $s$ est un ensemble complet et cohérent de littéraux de base objectifs.*

Un état du monde est une représentation syntaxique d'une interprétation de Herbrand. Donc, pour tout symbole de prédicat $n$-aire $P$, tous termes de base $t_1, \ldots, t_n$ et tout état du monde $s$, soit $P(t_1, \ldots, t_n) \in s$ soit $\neg P(t_1, \ldots, t_n) \in s$. Dans ce qui suit, quand nous décrivons un état du monde, nous omettrons les littéraux négatifs pour plus de lisibilité.

**Définition 13** *Soient $IC$ un ensemble de contraintes d'intégrité et $s$ un état du monde. $s$ est* cohérent *avec $IC$ ssi $s, IC \nvdash \bot$.*

**Définition 14** *Soient $\rho$ une réglementation, $IC$ un ensemble de contraintes d'intégrité et $s$ un état du monde cohérent avec $IC$. $\rho$ est cohérent par rapport à $IC$ dans $s$ ssi $\rho, IC, s \nvdash \bot$.*

**Exemple 2** *Reprenons l'exemple 1. Considérons que $IC$ contient deux contraintes : (1) un feu tricolore a une et une seule couleur et cette couleur peut être verte, orange ou rouge, et (2) un conducteur conduit un et un seul type de véhicule. Ici, $IC = \{\forall t\ TL(t) \rightarrow Color(t, green) \otimes Color(t, orange) \otimes$*

$Color(t, red), \forall x \forall y \forall z \quad Driver(x) \quad \wedge$
$Vehicle(x, y) \wedge Vehicle(x, z) \rightarrow y = z\}$[5].

*Soit $s$ l'état du monde* $\{Driver(A)$, $TL(T), IFO(A, T)$, $Vehicle(A, car)$, $Color(T, red)\}$.

*$s$ est tel que $s, IC \nvdash \bot$. Considérons une réglementation $\rho$ qui contient les trois règles $(r_0)$, $(r_1)$ et $(r_2)$. Dans ce cas, $\rho, IC, s \nvdash \bot$ (parce que le seul littéral déontique qui peut être déduit de $\rho$, $IC$ et $s$ est $OStop(A, T)$). Donc, $\rho$ est cohérent par rapport à $IC$ dans $s$.*

**Définition 15 (cohérence d'une réglementation)** *Soient $\rho$ une réglementation et $IC$ un ensemble de contraintes d'intégrité. $\rho$ est cohérent par rapport à $IC$ ssi pour tous les états du monde $s$ tels que $s, IC \nvdash \bot$ alors $\rho, IC, s \nvdash \bot$.*

## 2.6 Complétude d'une réglementation

Informellement, une réglementation est complète dès qu'elle permet de contraindre le comportement d'un agent dans toute situation. On peut se demander si cette définition a du sens : est-ce qu'une réglementation doit prendre en compte toutes les situations possibles ? Nous suggérons donc de définir une complétude partielle restreinte à deux formules $\varphi$ et $\psi$ : $\varphi$ représente une situation particulière dans laquelle nous voulons évaluer la réglementation et $\psi$ un symbole de prédicat gouverné par la réglementation. Nous voulons qu'une réglementation soit complète pour $\varphi$ et $\psi$ ssi dans toute situation où $\varphi$ est vraie, il est obligatoire (resp. permis, interdit) que $\psi$ soit vraie.

Ceci conduit à la définition suivante :

**Définition 16** *Soient $IC$ un ensemble de contraintes d'intégrité, $\rho$ une réglementation cohérente par rapport à $IC$ et $s$ un état du monde cohérent avec $IC$. Soient $\varphi(\overrightarrow{x})$ et $\psi(\overrightarrow{x})$ deux formules objectives, $\overrightarrow{x}$ représentant les variables libres dans $\varphi$ et $\psi(\overrightarrow{x})$ signifiant que les variables libres de $\psi$ sont un sous-ensemble de $\overrightarrow{x}$. $\rho$ est* $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-complète *par rapport à $IC$ dans $s$ pour $\vdash$ ssi pour toutes les substitutions de base $\chi$ telles que $s \vdash \varphi(\chi(\overrightarrow{x}))$ :*

$$\rho, s \vdash O\psi(\chi(\overrightarrow{x})) \textit{ ou}$$
$$\rho, s \vdash F\psi(\chi(\overrightarrow{x})) \textit{ ou}$$
$$\rho, s \vdash P\psi(\chi(\overrightarrow{x}))$$

**Exemple 3** *Considérons l'état du monde $s_0 = \{Driver(A), TL(T), IFO(A, T), Vehicle(A, Car), Color(T, red)\}$. Considérons $\rho$ et $IC$ définis dans l'exemple 2. $s_0$ est cohérent par rapport à $IC$ et $\rho, s \vdash O(Stop(A, T))$. Prenons $\varphi_0(x, t) \equiv TL(t) \wedge Driver(x) \wedge IFO(x, t)$ et $\psi_0(x, t) \equiv Stop(x, t)$. $s_0, IC \vdash IFO(A, T)$ et $\rho, IC, s_0 \vdash O(Stop(A, T))$. Donc $\rho$ est $(\varphi_0(x, t), \psi_0(x, t))$-complète par rapport $IC$ à $s_0$ pour $\vdash$.*

*Considérons maintenant $s_1 = \{Driver(A), TL(T), IFO(A, T), Vehicle(A, Truck), Color(T, red)\}$. $s_1$ est cohérent avec $IC$. $s_1, IC \vdash IFO(A, T)$ mais $\rho, IC, s_1 \nvdash O\psi_0(A, T)$, $\rho, IC, s_1 \nvdash P\psi_0(A, T)$ et $\rho, IC, s_1 \nvdash F\psi_0(A, T)$. Donc $\rho$ est $(\varphi_0(x, t), \psi_0(x, t))$-incomplète par rapport à $IC$ dans $s_1$ pour $\vdash$. En fait, aucune règle de la réglementation peut être appliquée car le véhicule n'est pas un bus, mais un camion.*

Les définitions précédentes sont généralisées comme suit :

**Définition 17 (complétude d'une réglementation)** *Soient $IC$ un ensemble de contraintes d'intégrité et $\rho$ une réglementation. Soient $\varphi(\overrightarrow{x})$ et $\psi(\overrightarrow{x})$ deux formules objectives avec la même signification que dans la définition 16. $\rho$ est* $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-complète *par rapport à $IC$ pour $\vdash$ ssi pour tout état du monde $s$ cohérent avec $IC$, $\rho$ est $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-complète par rapport à $IC$ dans $s$ pour $\vdash$.*

La complétude est un aspect important des réglementations. Dans une situation donnée, sans comportement stipulé, n'importe quel comportement peut être observé et les conséquences peuvent être très importantes. Pour pouvoir travailler avec une réglementation incomplète, on peut (1) détecter les « trous » de la réglementation et les envoyer aux concepteurs de la réglementation pour qu'ils puissent la corriger ou (2) détecter les « trous » de la réglementation et appliquer sur ces trous des règles de complétion pour les corriger. La première solution peut être très difficile à mettre en œuvre, c'est pourquoi nous choisissons la seconde solution.

[5] L'introduction de l'égalité est faite dans le même esprit que dans [12].

# 3 Raisonner avec des réglementations incomplètes

## 3.1 Défauts pour la complétion de réglementations

Raisonner avec des informations incomplètes est un problème classique en logique et en intelligence artificielle : peut-on inférer quelque chose sur une information qui n'est pas présente dans une base de croyances ? Plusieurs approches on été définies, mais nous nous intéressons à une en particulier, le raisonnement par défaut. Le principe du raisonnement par défaut est simple : si une information n'est pas contradictoire avec les informations qui peuvent être classiquement déduites de la base de croyances, alors on peut déduire une autre information de la base de croyances. Un exemple classique est le suivant : supposons qu'un agent croit que « *tous les oiseaux volent* », que « *les pingouins ne volent pas* » et que « *les pingouins sont des oiseaux* ». La représentation de cet ensemble de formules dans FOL est incohérent (un oiseau qui est également un pingouin vole et ne vole pas en même temps). En fait, la première règle « *tous les oiseaux volent* » est un défaut : « *si a est un oiseau et s'il n'est pas incohérent que a vole, alors a vole* »[6]. Si a est un pingouin, alors « a vole » ne peut pas être déduit et s'il ne peut pas être déduit que a est un pingouin, alors on peut en déduire que a vole.

La logique des défauts est une extension nonmonotone de la logique du premier ordre introduite par Reiter [17] pour formaliser le raisonnement par défaut. Nous suivons ici la présentation de cette logique donnée par Besnard [1].

Un défaut $d$ est une configuration $\frac{P : J_1, \ldots, J_n}{C}$ où $P, J_1, \ldots, J_n, C$ sont des formules fermées du premier ordre. $P$ est appelée le *pré-requis* de $d$, $J_1, \ldots, J_n$ la *justification* de $d$ et $C$ la *conséquence* de $d$. Une théorie de défauts $\Delta = (D, F)$ est composée d'un ensemble de formules closes objectives $F$ (les faits) et d'un ensemble de défauts.

Une théorie des défauts $(D, F)$ peut être représentée par une *forme de surface* $(D', F)$ à condition que

$$D = \{\frac{P(\overrightarrow{a}) : J_1(\overrightarrow{a}), \ldots, J_n(\overrightarrow{a})}{C(\overrightarrow{a})} \ : \ \frac{P(\overrightarrow{x}) : J_1(\overrightarrow{x}), \ldots, J_n(\overrightarrow{x})}{C(\overrightarrow{x})} \in D' \text{ et } \overrightarrow{a} \text{ est un terme de base}\}$$

et chaque élément de $D'$ est de la forme $\frac{P(\overrightarrow{x}) : J_1(\overrightarrow{x}), \ldots, J_n(\overrightarrow{x})}{C(\overrightarrow{x})}$ où $P(\overrightarrow{x}), J_1(\overrightarrow{x}), \ldots, J_n(\overrightarrow{x}), C(\overrightarrow{x})$ sont des formules du premier ordre fermées avec des variables libres apparaissant dans $\overrightarrow{x}$.

En utilisant des défauts, on obtient des *extensions*, i.e. des ensembles de formules qui sont déduites monotoniquement et nonmonotoniquement de $F$. Soit $\Delta = (D, F)$ une théorie des défauts où les défauts ne contiennent que des formules closes, alors une extension de $\Delta$ est un ensemble de formules $E$ vérifiant les conditions suivantes :

1. $F \subseteq E$
2. $Th(E) = E$ où $Th(E) = \{\varphi \ : \ E \vdash \varphi\}$
3. si $\frac{P : J_1, \ldots, J_n}{C}$ est un défaut de $D$, alors si $P \in E$ et $J_1$ est cohérent avec $E$, …, $J_n$ est cohérent avec$E$, alors $C \in E$

Une théorie des défauts peut avoir plusieurs extensions ou aucune extension. Reiter a montré dans [17] que si $F$ est cohérent et si $(D, F)$ a une extensions, alors cette extension est cohérente. Il a également montré que toute théorie des défauts normale et fermée a au moins une extension.

Nous ne nous intéressons pas ici à la croyance d'une formule objective $\psi$ donnée, mais à la dérivation depuis une réglementation donnée du fait que $\psi$ soit obligatoire, permise ou interdite (ces trois cas sont les seules possibles à cause de l'axiome D de $O$). Donc, si la réglementation est incomplète pour une formule objective $\psi$ (i.e. on ne peut en déduire ni $O\psi$ ni $F\psi$ ni $P\psi$), alors elle ne peut être complétée qu'en supposant que $O\psi$ peut être déduite, ou $P\psi$, ou $F\psi$. Cela va nous conduire à la construction des trois ensembles de défauts présentés ci-dessous.

Dans ce qui suit, soient $IC$ un ensemble de contraintes d'intégrité, $\rho$ une réglementation co-

[6] Dans ce cas, l'information qui n'est pas contradictoire avec la base de croyances et la nouvelle information sont identiques.

hérence avec $IC$ est $s$ un état du monde cohérent avec $IC$. Soient $\varphi(\overrightarrow{x})$ et $\psi(\overrightarrow{x})$ les deux formules objectives vérifiant la définition 16.

**Définition 18** *Soient $E_F(\overrightarrow{x})$, $E_P(\overrightarrow{x})$ et $E_O(\overrightarrow{x})$ trois formules objectives telles que leur ensemble respectif de variables libres est dans $\overrightarrow{x}$. On définit un ensemble de configurations comme suit :*

$$(DF_{\varphi,\psi}) \quad \frac{\varphi(\overrightarrow{x}) \wedge E_F(\overrightarrow{x}) \ : \ F\psi(\overrightarrow{x})}{F\psi(\overrightarrow{x})}$$

$$(DP_{\varphi,\psi}) \quad \frac{\varphi(\overrightarrow{x}) \wedge E_P(\overrightarrow{x}) \ : \ P\psi(\overrightarrow{x})}{P\psi(\overrightarrow{x})}$$

$$(DO_{\varphi,\psi}) \quad \frac{\varphi(\overrightarrow{x}) \wedge E_O(\overrightarrow{x}) \ : \ O\psi(\overrightarrow{x})}{O\psi(\overrightarrow{x})}$$

*Une théorie des défauts $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-complète pour $\rho$ et $s$ est une théorie des défauts $\Delta_{\rho,s}(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$ dont la forme de surface est donnée par $(\{DF_{\varphi,\psi}, DP_{\varphi,\psi}, DO_{\varphi,\psi}\}, \rho \cup s)$*

Nous pouvons compléter une réglementation incomplète de telle sorte que $\psi(\overrightarrow{x})$ est interdite ($DF_{\varphi,\psi}$), permise ($DP_{\varphi,\psi}$) ou obligatoire ($DO_{\varphi,\psi}$) en nous appuyant sur $E_F(\overrightarrow{x})$, $E_P(\overrightarrow{x})$ et $E_O(\overrightarrow{x})$. Nous pouvons maintenant définir une nouvelle relation d'inférence $\vdash_*$ en utilisant les règles par défaut.

**Définition 19** *Soit $\gamma$ une formule de FOSDL. $\rho, s \vdash_* \gamma$ ssi $\gamma \in \bigcup E_{\Delta_{\rho,s}(\varphi(\overrightarrow{x}),\psi(\overrightarrow{x}))}$ où $\bigcup E_{\Delta_{\rho,s}(\varphi(\overrightarrow{x}),\psi(\overrightarrow{x}))}$ est l'union de toutes les extensions de $\Delta_{\rho,s}(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$.*

De plus, nous notons $Th_*(E) = \{\varphi \ : \ E \vdash_* \varphi$ et $\varphi$ est fermée$\}$.

Revenons sur notre définition de $\vdash_*$. Classiquement, il y a deux façons de définir $\vdash_*$ :
– *existentiellement*, c'est-à-dire que $\rho, s \vdash_* \gamma$ ssi il y a *une* extension de $\Delta_{\rho,s}(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$ qui contient $\gamma$
– *universellement*, c'est-à-dire que $\rho, s \vdash_* \gamma$ ssi $\gamma$ apparaît dans *toutes* les extensions de $\Delta_{\rho,s}(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$

Malheureusement, ces deux définitions ne nous satisfont pas : si l'on utilise l'inférence existentielle, nous ne pouvons pas détecter de contradictions (par exemple $O\varphi$ serait dans $E_1$ et $O\neg\varphi$ serait dans $E_2$), et si l'on utilise l'inférence universelle, nous n'obtiendrons pas une réglementation complète (s'il y a plus d'une extension, cela signifie que deux règles différentes concernant la même proposition peuvent être appliquées).

Pourquoi alors utiliser l'union de toutes les extensions de $\Delta_{\rho,s}(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$ ? Cela représente en fait l'ensemble qui contient toutes les obligations, permissions et interdictions qui peuvent être déduites de la réglementation (en utilisant des défauts ou pas). Tout agent appliquant la réglementation doit trouver le comportement à adopter dans cet ensemble.

On peut alors se demander si l'utilisation de l'union de toutes les extensions possibles peut mener à une contradiction. Mais, comme nous le verrons dans la section 3.2, une seule extension sera obtenue dans les cas qui nous intéressent.

L'étape suivante est de définir les conditions sous lesquelles la réglementation est complète et cohérente avec cette nouvelle inférence.

### 3.2 Cohérence et complétude de la régulation complétée

Nous étendons tout d'abord les définitions 15, 16 et 17 en utilisant $\vdash_*$ à la place de $\vdash$ dans ces définitions. Pour distinguer les nouvelles notions de cohérence et de complétude des anciennes, nous utiliserons $*$ comme préfixe (par exemple, nous écrirons « $*$-cohérence ») ou écrirons « pour $\vdash_*$ » (par exemple, nous écrirons « cohérent pour $\vdash_*$ »).

Le résultat principal concernant la complétude et la cohérence de la réglementation obtenue en utilisant la théorie des défauts définie précédemment est présenté dans la proposition suivante.

**Proposition 2** *Considérons un ensemble de contraintes d'intégrité $IC$, une réglementation $\rho$ cohérente avec $IC$ et un état du monde $s$ cohérent avec $IC$ et tel que $\rho \cup s$ est cohérent. Soient $\varphi(\overrightarrow{x})$ et $\psi(\overrightarrow{x})$ deux formules objectives vérifiant la définition 16 et $\Delta_{\rho,s}(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$ la théorie des défauts correspondante.*

*Les propositions suivantes sont équivalentes :*

1. *pour tout vecteur $\overrightarrow{a}$ de termes de base, si $s \vdash \varphi(\overrightarrow{a})$, $\rho, s \nvdash O\psi(\overrightarrow{a})$, $\rho, s \nvdash P\psi(\overrightarrow{a})$ et $\rho, s \nvdash F\psi(\overrightarrow{a})$ (i.e. $\rho$ n'est pas $(\varphi(\overrightarrow{a}), \psi(\overrightarrow{a}))$-complète dans $s$), alors $s \vdash E_F(\overrightarrow{a}) \otimes E_P(\overrightarrow{a}) \otimes E_F(\overrightarrow{a})$.*

2. *$\rho$ est cohérent et $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-complète pour $\vdash_*$ dans $s$.*

Cette proposition caractérise les conditions nécessaires et suffisantes pour que les défauts puissent compléter de façon cohérente une réglementation incomplète. Plus précisément, cette proposition signifie que si chaque fois que la réglementation ne permet pas de dériver le comportement attendu alors un seul des $E_i$ est vrai, alors les défauts complètent de façon cohérente la réglementation (un et un seul défaut sera appliqué pour une $\psi(\overrightarrow{a})$ particulière).

**Exemple 4** *Considérons l'exemple précédent et $s_1 = \{Driver(A), TL(T), IFO(A,T), Vehicle(A, truck), Color(T, red)\}$. $\rho$ est incomplète dans $s_1$ pour $\varphi_0(x,t) \equiv Driver(A) \wedge TL(T) \wedge IFO(A,T)$ et $\psi_0(x,t) \equiv Stop(A,T)$ in $s_1$.*

*Posons $E_F(x,t) = Vehicle(x, truck) \wedge Color(t, green)$, $E_P(x,t) = Vehicle(x, truck) \wedge Color(t, orange)$ et $E_O(x,t) = Vehicle(x, truck) \wedge Color(t, red)$, alors $s_1 \vdash E_O(A,T)$. Donc $\rho$ est cohérente et $(\varphi_0(x,t), \psi_0(x,t))$-complète pour $\vdash_*$ dans $s_1$.*

Même si cette condition nécessaire et suffisante est intéressante en théorie, elle n'est pas réellement utile dans des cas concrets : pour vérifier que cette condition est satisfaite, nous devrions détecter chaque « trou » dans la réglementation. Cette détection est une opération que nous voulons éviter. C'est pourquoi nous proposons des conditions plus générales qui sont suffisantes, mais non nécessaires, pour pouvoir compléter la réglementation. Nous présentons ainsi deux corollaires immédiats de la définition précédente.

**Corollaire 1** *Si*

$$s \vdash \forall \overrightarrow{x}\ \varphi(\overrightarrow{x}) \rightarrow E_O(\overrightarrow{x}) \otimes E_F(\overrightarrow{x}) \otimes E_P(\overrightarrow{x})$$

*alors $\rho$ est cohérente et $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-complète par rapport à $IC$ pour $\vdash_*$ dans $s$.*

**Exemple 5** *Considérons $s_2 = \{Driver(A), TL(T), IFO(A,T), Vehicle(A, bike), Color(T, red)\}$. $s_2$ est cohérent avec $IC$. Considérons la réglementation définie dans l'exemple 1.*

*Cette fois, considérons $E_F(x,t) = Color(t, green)$, $E_P(x,t) = Color(t, orange)$ et $E_O(x,t) = Color(t, red)$. $s_2 \vdash E_O(A,T)$. Donc $\rho$ est $*$-cohérente et $*$-complète pour $\varphi_0(x,t)$ et $\psi_0(x,t)$ dans $s_2$. Mais on a également $s_1 \vdash E_O(A,T)$, donc $\rho$ est $*$-cohérente et $(\varphi_0(x,t), \psi(x,t))$-complète pour $\vdash_*$ dans $s_1$. Ces $E_i$ plus généraux permettent d'avoir une réglementation complète pour tout type de véhicule.*

**Corollaire 2** *Si*

$$IC \vdash \forall \overrightarrow{x}\ E_O(\overrightarrow{x}) \otimes E_F(\overrightarrow{x}) \otimes E_P(\overrightarrow{x})$$

*alors $\rho$ est cohérent et $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-complète par rapport à $IC$ pour $\vdash_*$.*

**Exemple 6** $IC \vdash \forall t\ Color(t, red) \otimes Color(t, green) \otimes Color(t, orange)$. *Donc $\rho$ est $*$-cohérente et $(\varphi_0(x,t), \psi_0(x,t))$-complète pour $\vdash_*$.*

*$IC$ spécifie qu'un feu tricolore n'a qu'une et une seule couleur parmi $Red$, $Orange$ et $Green$. S'il y a un $E_i$ pour chaque couleur, nous sommes sûrs que quelque soit la situation, nous pouvons appliquer un et un seul défaut s'il y a un « trou » dans la réglementation.*

Une autre solution serait de prendre des $E_i$ fixés. Par exemple, nous pourrions prendre un $E_i$ égal à $\top$ et les deux autres à $\bot$. On peut distinguer trois cas :

- supposons que $E_F \equiv \top$, $E_P \equiv \bot$ et $E_O \equiv \bot$. Dans ce cas, tout ce qui n'est pas spécifié comme obligatoire ou permis par la réglementation est interdit. Ce comportement strict peut être observé par exemple dans les réglementations qui régissent un système hautement sécurisé où chaque action doit être explicitement autorisé avant d'être exécutée.
- supposons que $E_F \equiv \bot$, $E_P \equiv \top$ et $E_O \equiv \bot$. Nous sommes ici dans la situation opposée : tout ce qui n'est pas obligatoire ou interdit est permis. Ce comportement « tolérant » peut être observé par exemple pour des réglementations pour certains systèmes faiblement sécurisés dans lesquels tout ce qui n'est pas obligatoire ou interdit est implicitement permis.
- supposons que $E_F \equiv \bot$, $E_P \equiv \bot$ et $E_O \equiv \top$. Dans ce cas, chaque action qui n'est pas interdite ou permise doit être exécutée.

# 4 Exemples de réglementations : politiques d'échange d'information

Une politique d'échange d'information est une réglementation qui contraint le comportement des agents d'un système multi-agents en ce qui concerne la communication des informations. Pour décrire de telles politiques, nous avons besoin de cinq symboles de prédicats : $Agent$, $Info$, $Receive$, $Topic$ et $Tell$. $Agent(x)$ signifie que $x$ est un agent, $Info(i)$ signifie que $i$ est une information, $Receive(x, i)$ signifie que l'agent $x$ reçoit l'information $i$. $Topic(i, t)$ signifie que l'information $i$ traite du sujet $t$. $Tell(x, i, y)$ signifie que l'agent $x$ envoie à l'agent $y$ une information $i$. Nous définissons également les constantes $a$, $b$, $i_1$, $EqtCheck$, $ExpRisk$, $Meeting$ et $EqtOutOfOrder$.

La cohérence de telles politiques est définie par la définition 14 et leur complétude est définie en instanciant la définition 16 avec les formules spécifiques suivantes :

$$\begin{aligned}\varphi(x,i,y) \equiv & Agent(x) \wedge Info(i) \wedge Receive(x,i) \wedge \\ & Agent(y) \wedge \neg(x = y)\\ \psi(x,i,y) \equiv & Tell(x,i,y)\end{aligned}$$

Cela conduit à la définition suivante :

**Définition 20** *Soient $IC$ un ensemble de contraintes d'intégrité, $s$ un état du monde cohérent avec $IC$ et $\rho$ une réglementation cohérente dans $s$ par rapport à $IC$. $\rho$ est complète par rapport à $IC$ dans $s$ pour $\vdash$ ssi pour toute substitution de base $\chi$ telle que $s \vdash Agent(\chi(x)) \wedge Info(\chi(y)) \wedge Receive(\chi(x), \chi(i)) \wedge Agent(\chi(y)) \wedge \neg(\chi(x) = \chi(y))$ :*

$$\rho, s \vdash O\,Tell(\chi(x), \chi(i), \chi(y)) \text{ ou}$$
$$\rho, s \vdash F\,Tell(\chi(x), \chi(i), \chi(y)) \text{ ou}$$
$$\rho, s \vdash P\,Tell(\chi(x), \chi(i), \chi(y))$$

Les défauts sont donc les suivants :

$$(DF_{\varphi,\psi}) \quad \frac{\varphi(x,i,y) \wedge E_F(x,i,y) : F\,Tell(x,i,y)}{F\,Tell(x,i,y)}$$

$$(DP_{\varphi,\psi}) \quad \frac{\varphi(x,i,y) \wedge E_P(x,i,y) : P\,Tell(x,i,y)}{P\,Tell(x,i,y)}$$

$$(DO_{\varphi,\psi}) \quad \frac{\varphi(x,i,y) \wedge E_O(,i,y) : O\,Tell(x,i,y)}{O\,Tell(x,i,y)}$$

Les résultats donnés dans la section 3 sont valides. En particulier, nous pouvons toujours définir trois cas :

- $E_F \equiv \top$, $E_P \equiv \bot$ et $E_O \equiv \bot$.
  Cela s'applique à des systèmes multi-agents hautement sécurisés dans lesquels toute action de communication doit être explicitement obligatoire ou permises pour être exécutée.
- $E_F \equiv \bot$, $E_P \equiv \top$ et $E_O \equiv \bot$.
  Ce cas s'applique à des systèmes faiblement sécurisés dans lesquels toute action de communication qui n'est pas explicitement interdite est autorisée.
- $E_F \equiv \bot$, $E_P \equiv \bot$ et $E_O \equiv \top$.
  Dans ce cas, à moins que cela ne soit explicitement mentionné, l'envoi d'information est obligatoire.

Pour illustrer cela, considérons l'exemple d'une entreprise dans laquelle il y a un manager et deux employés. Considérons la politique $\pi_0$ avec une seule règle précisant que « *les managers ne doivent pas informer les employés des informations concernant la vérification des équipements* ». Cette règle est modélisée par[7]

$$\begin{aligned}\forall x \forall i \forall y\ & Manager(x) \wedge Employee(y) \wedge \\ & Receive(x,i) \wedge Topic(i, EqtChk) \rightarrow \\ & O\neg Tell(x,i,y)\end{aligned}$$

Considérons $IC = \emptyset$ (il n'y a pas de contraintes d'intégrité) et l'état du monde $s_0 = \{Agent(a), Agent(b), Manager(a), Employee(b), Info(i_1), Topic(i_1, ExpRisk), Receive(a, i_1)\}$. Dans cette situation, $a$ est un manager et $b$ un employé. $a$ a reçu une information $i_1$ dont le sujet est « risque d'explosion ».

Comme $\pi_0$ ne contient qu'une et une seule règle et $s_0$ est cohérent avec $IC$, $\pi_0$ est cohérente dans $s_0$.

Cependant, nous avons $s_0 \vdash Agent(a) \wedge Info(i_1) \wedge Receive(a, i_1) \wedge Agent(b) \wedge \neg(a = b)$ mais $\pi_0, s_0 \nvdash O(Tell(a, i_1, b))$ et $\pi_0, s_0 \nvdash P(Tell(a, i_1, b))$ et $\pi_0, s_0 \nvdash F(Tell(a, i_1, b))$. Donc $\pi_0$ est incomplète pour $\vdash$.

L'incomplétude vient du fait que la politique contraint le comportement du manager si il ou elle reçoit une information à propos de la vérification des équipements, mais elle ne dit rien par

[7] Les noms de prédicats sont évidents et ne sont pas définis formellement dans le langage.

rapport aux informations concernant les risques d'explosion.

Pour pouvoir compléter la politique précédente, nous pourrions prendre $E_F(x, y, i) = Topic(i, EqtChk)$, $E_P(x, y, i) = \bot$ et $E_O(x, y, i) = Topic(i, ExpRisk)$. Un tel choix oblige le manager à dire à ses employés les informations concernant les risques d'explosion. On peut vérifier que $\pi_0$ est cohérente et complète pour $\vdash_*$ dans $s_0$ pour $\varphi(x, i, y)$ et $\psi(x, i, y)$.

Considérons maintenant que $IC$ contient la contrainte « *une information a un et un seul sujet et ce sujet peut être EqtChk, ExpRisk, Meeting ou EqtOutOfOrder* ». Prenons

$$\begin{aligned} E_F(x, y, i) \equiv & Topic(i, EqtChk) \vee \\ & Topic(i, Meeting) \\ E_P(x, y, i) \equiv & Topic(i, EqtOutOfOrder) \\ E_O(x, y, i) \equiv & Topic(i, ExpRisk) \end{aligned}$$

On peut alors appliquer le corollaire 2 pour conclure que $\pi_0$ est $*$-complète et $*$-cohérente pour $\varphi(x, i, y)$ et $\psi(x, i, y)$.

## 5 Conclusion

Dans cet article, nous nous sommes intéressés à l'analyse de la cohérence et de la complétude de réglementations qui peuvent exister dans des sociétés d'agents pour contraindre leur comportement.

Plus précisément, nous avons défini un cadre logique et montré comment exprimer une réglementation dans ce cadre. Nous avons alors donné une définition de la cohérence et de la complétude d'une réglementation. La définition de la complétude que nous avons donnée est assez générale. Nous nous sommes également intéressés aux réglementations incomplètes et proposé une méthode pour compléter ces réglementations en utilisant des défauts. Nous avons établi plusieurs résultats qui montrent quand ces défauts peuvent compléter de façon cohérente une réglementation.

Bien que ces notions (sauf les défauts) soient déjà présentées dans [6, 7], nous avons étendu ces travaux précédents sur deux points :

– premièrement, nous avons utilisé une logique modale du premier ordre pour représenter les réglementations. Ceci nous a permis de clairement distinguer les propriétés des objets des notions déontiques et de garder l'expressivité de FOL pour les propriétés des objets.
– deuxièmement, l'approche prise dans les papiers précédents pour compléter la réglementation était d'étendre la CWA (*Closed World Assumption*) définie par Reiter pour compléter des bases de données du premier ordre [16]. Nous avons choisi ici d'utiliser le raisonnement par défaut, qui est une solution plus élégante.

La notion de complétude développée ici est en fait une sorte de « complétude locale », dans le sens où nous demandons d'avoir $O(\psi(X))$, $P(\psi(X))$ ou $F(\psi(X))$ seulement pour la proposition $\phi(X)$. Cela se rapproche de la notion de complétude introduite dans le domaine des bases de données par [18] et [10], qui ont remarqué que certaines contraintes d'intégrité exprimées sur une base de données sont des règles à propos de ce que la base de données devrait savoir (où, pour le dire différemment, ce sont des règles portant sur ce qui devrait être déduit de la base de données). Par exemple, la contrainte d'intégrité « tout employé a un numéro de téléphone, un numéro de fax ou une adresse mail » exprime en fait que, pour tout employé contenu dans la base de données, celle-ci connaît son numéro de téléphone, son numéro de fax ou son adresse mail[8]. Comme précisé par Reiter [18], cette contrainte d'intégrité exprime une sorte de complétude locale de la base de données. Les défauts de Reiter peuvent être utilisé pour compléter la base de données dans de tels cas. Par exemple, une des règles pourrait être que si la base de données ne contient pas les informations demandées pour un employé donné, alors on pourrait supposer que le numéro de téléphone de cet employé est celui de son département.

L'étude du lien formel existant entre la complétude introduite dans ce papier et la notion de complétude locale constitue une extension intéressante de ce travail.

De plus, pour pouvoir travailler avec des réglementations plus générales, ce travail doit être étendu, en particulier en considérant des notions comme le temps et l'action. Comme montré dans [11], la question du temps est très importante lorsque l'on modélise des réglementations et nous devront considérer différents types de

[8] Cela n'empêche pas bien sûr que dans le monde réel, un employé de l'entreprise n'a ni numéro de téléphone, ni numéro de fax, ni adresse mail.

temps parmi lesquels, au moins, le temps de validité des normes et les *deadlines* imposées sur les obligations. Dans beaucoup d'exemples présentés ici, les prédicats concernés par les opérateurs déontiques représentent des actions (dire, stopper etc.). L'addition d'un opérateur modal dynamique et/ou temporel pourrait être intéressant. Nous obtiendrions alors une logique multimodale avec une très bonne expressivité.

Enfin, nous avons développé ici un modèle très simple des notions déontique en utilisant SDL et nombre de problèmes classiques en logique déontique ne sont pas traités ici : normes avec exceptions, contrary-to-duties, obligations collectives etc. Une autre extension de ce travail serait de définir une logique qui permettrait de traiter ces problèmes.

## Références


[1] P. Besnard. *An introduction to default logic*. Springer-Verlag, 1989.

[2] P. Bieber and F. Cuppens. Expression of confidentiality policies with deontic logic. In *Deontic logic in computer science : normative system specification*, pages 103–121. John Wiley and Sons, 1993.

[3] B. F. Chellas. *Modal logic, an introduction*. Cambridge University Press, 1980.

[4] L. Cholvy. Checking regulation consistency by using SOL-resolution. In *International Conference on Artificial Intelligence and Law*, pages 73–79, 1999.

[5] L. Cholvy and F. Cuppens. Analyzing consistency of security policies. In *1997 IEEE Symposium on Security and Privacy*, pages 103–112. IEEE, 1997.

[6] L. Cholvy and S. Roussel. Reasoning with incomplete information exchange policies. In K. Mellouli, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 9th European Conference, ECSQARU'07*, number 4724 in Lecture Notes in Articial Intelligence, pages 683–694. Springer-Verlag, 2007.

[7] L. Cholvy and S. Roussel. Consistency and completeness of regulations. In *Proceedings of the third International Workshop on Normative Multiagent Systems (NORMAS'08)*, pages 51–65, 2008.

[8] F. Cuppens and R. Demolombe. A modal logical framework for security policies. In *Lectures Notes in Artificial Intelligence*, volume 1325, page 1997. Springer, 1997.

[9] R. Demolombe. Syntactical characterization of a subset of domain independent formulas. *Journal of the Association for Computer Machinery*, 39(1) :71–94, 1982.

[10] R. Demolombe. Database validity and completeness : another approach and its formalisation in modal logic. In Enrico Franconi and Michael Kifer, editors, *Proc. of the 6th International Workshop on Knowledge Representation meets Databases (KRDB'99)*, pages 11–13. CEUR-WS.org, 1999.

[11] R. Demolombe, P. Bretier, and V. Louis. Norms with deadlines in dynamic deontic logic. In Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso, editors, *Proceedings of ECAI 2006, 17th European Conference on Artificial Intelligence*, pages 751–752. IOS Press, 2006.

[12] M. Fitting and R. L. Mendelsohn. *First-order modal logic*. Kluwer Academic, 1999.

[13] C. Groulier. *Normes permissives et droit public*. PhD thesis, Université de Limoges, 2006. Available on `http://www.unilim.fr/scd/theses/accesdoc.html`. In French.

[14] R. Hilpinen, editor. *Deontic logic*. Reidel Publishing Company, 1971.

[15] P. Mc Namara. Deontic logic. Stanford Encyclopedia of Philosophy, `http://plato.stanford.edu/entries/logic-deontic/`.

[16] R. Reiter. On closed world databases. In J. Minker J.-M. Nicolas H. Gallaire, editor, *Logic and Databases*. Plenum Publications, 1978.

[17] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1,2), 1980.

[18] R. Reiter. What should a database know ? *Journal of Logic Programming*, 14(1,2) :127–153, 1992.

[19] G. H. von Wright. Deontic logic. *Mind*, 60 :1–15, 1951.

[20] E. Vranes. The definition of "norm conflict" in international law and legal theory. *The European Journal of International Law*, 17(2) :395–418, 2006.

# Utiliser des protocoles d'interaction et de la mémoire pour l'apprentissage par renforcement d'actes de communication

S. Hoet* N. Sabouret*
Shirley.Hoet@lip6.fr Nicolas.Sabouret@lip6.fr

*Laboratoire d'Informatique de Paris 6
104, avenue du président Kennedy
75016 Paris – FRANCE

**Résumé :**

Dans ce papier, nous nous intéressons à l'apprentissage par renforcement pour la communication entre agents dans les SMA. L'étude de la littérature nous a permis de mettre en évidence trois principaux problèmes. Tout d'abord il est nécessaire pour l'agent de construire les actes de communication qu'il utilise pour communiquer avec les autres agents. Ensuite il nous faut adapter les actes de communication afin de les modéliser en tant qu'actions pour leur appliquer des techniques d'apprentissage par renforcement. Enfin nous devons résoudre les problèmes liés à la non markovité des systèmes multi-agents afin de pouvoir utiliser les algorithmes usuels d'apprentissage par renforcement utilisé dans le cadre mono-agent. Nous présentons un algorithme d'apprentissage par renforcement qui permet de traiter ces trois problèmes de manière entièrement distribuée. Nous nous appuyons pour cela sur l'utilisation d'agents introspectifs capables de raisonner sur leurs actions et sur leurs états. Puis nous démontrons que nous pouvons assimiler un acte de communication à une action en nous fondant sur la théorie des actes de langage et les travaux sur les PDSMs. Enfin nous montrons comment l'utilisation d'actes de communication peut résoudre le problème de non markovité en étant toutefois complétée par l'utilisation de la mémoire de l'agent.

**Mots-clés :** Apprentissage par renforcement, Acte de language, Système multi-agent

**Abstract:**

In this paper, we propose a reinforcement learning mechanism for multi-agent communication. Existing work underlines three main problems. First, agents need to construct appropriate communicative acts. Second, we have to implement the whole communication process as agent actions, so as to apply classical reinforcement learning algorithms. Last, the non-markovity of MAS raises several issues that have to be solved to guarantee the algorithm convergence. We propose a distributed method to deal with these three problems. This method is based upon introspective agents, that can reason about their own actions and data. We then prove that communicative acts can be seen as actions, according to the speech acts theory, using a Semi-Markovian model of the problem. Last, we show how communicative acts can solve non-markovity issues, using a limited memory of passed actions and observations.

**Keywords:** Reinforcement Learning, Communication, Multi-Agent System

## 1 Introduction

Dans les systèmes multi-agents, la communication directe est une forme d'interaction entre les agents qui consiste en l'envoi d'acte de communication et qui repose sur l'intention de communiquer des agents. En pratique dans les SMA l'agent connaît le contenu et le destinataire du message ainsi que le moment où l'envoyer. Mais si l'on se place dans des SMA ouverts c'est-à-dire où les agents sont hétérogènes, un agent doit pouvoir s'adapter en apprenant à communiquer en fonction de ses besoins sans avoir de connaissances préalables sur ce qu'il peut communiquer ni quand le communiquer.

Il existe de nombreux travaux sur l'apprentissage d'actions pour un agent, notamment des méthodes d'apprentissage par renforcement. L'apprentissage par renforcement est une technique permettant à un agent d'apprendre par essai/erreur à choisir la meilleure action selon la situation dans laquelle il se trouve [SB98]. Il se résout facilement par des algorithmes tel que le Q-Learning [Wat89] qui repose sur le modèle mathématique des processus décisionnels de Markov [Put94]. Or si l'on se réfère à la théorie des actes de langages [Aus62, Sea85] sur laquelle reposent les langages de communication des systèmes multi-agents, un acte de communication est assimilable à une action dans le sens où celui ci produit des effets sur son destinataire. Par conséquent il est envisageable d'utiliser l'apprentissage par renforcement pour apprendre à communiquer. C'est pourquoi dans ce papier nous proposons un algorithme d'apprentissage par renforcement pour les actes de communication.

Cependant cette approche pose des difficultés dans le cadre des SMA. Pour commencer, peu de travaux ont étudiés le problème de l'apprentissage d'actes de communication. Par exemple, [Tan93, Mat98] ont utilisé la communication dans leur modèle d'apprentissage pour traiter des états cachés, mais ils utilisent une communication dirigée (l'agent sait ce qu'il doit com-

muniquer et à qui). L'apprentissage des actes de communication eux mêmes (dans le sens de quoi, quand et à qui) n'a pas été encore étudié dans le cadre des PDMs.

Le second problème dans les PDMs est que les actions sont toutes considérées comme atomiques : elles sont exécutées sur un pas de temps d'exécution et leur effets sont immédiatement disponible à l'agent ou à l'environnement. Au contraire la communication est un exemple typique d'action à durée variable, principalement parce que les SMA sont asynchrones. Quand un agent demande à un autre agent d'exécuter une action, il n'a pas de garantie que celui ci va l'exécuter immédiatement, même si celui ci a répondu qu'il acceptait d'effectuer cette tâche. Pour résoudre ce problème une solution est d'utiliser les Processus Décisionnel Semi-Markoviens (PDSMs) qui permettent de modéliser des PDMs avec des actions à durée variable [Put94, Ber95]. Un avantage des PDSMs, comme l'a montré [BD95], est que les algorithmes classiques d'apprentissage par renforcement peuvent être étendus aux PDSMs.

Cependant, l'utilisation des PDSMs n'apparaît pas comme suffisante, parce que la propriété de Markov n'est plus valide dans un SMA asynchrone. Bien que beaucoup de travaux admettent que cette propriété n'est pas obligatoire pour que l'algorithme d'apprentissage converge vers la solution, il existe des heuristiques pour réduire la "non-markovité" quand l'algorithme ne converge pas. Une solution proposée est de modéliser le problème en utilisant les PDMs Partiellement Observables (PDMPOs[CKL94]), c'est-à-dire des PDMs dans lequel l'agent a accès à seulement une partie du problème. Les PDMPOs permettent de prendre en compte le fait que les autres agents agissent sur l'environnement en représentant leur actions comme des états cachés à l'agent apprenant. Par définition les algorithmes d'apprentissage par renforcement fondé sur le modèles des PDMs ne convergent pas dans le cadre des PDMPOs [SJJ94]. Cependant, [Mcc96, DS03, Dut00] ont montré qu'il est possible de convertir un PDMPO en PDM en utilisant la mémoire de l'agent. Comme nous le montrerons plus tard cette solution est adaptée au cadre de l'apprentissage d'actes de communication.

Dans ce papier, nous proposons un algorithme d'apprentissage d'actes de communication qui permet de prendre en compte l'observabilité partielle du monde et l'asynchronisme lié aux SMA en utilisant de la mémoire et des PDSMs.

La suite de cet article est organisée de la façon suivante. Dans la section 2 nous rappelons brièvement la forme d'un acte de communication ainsi que l'utilisation de la mémoire pour résoudre un PDMPO. Dans la section 3 nous présentons notre algorithme et enfin dans la section 4 nous résumerons nos travaux avant d'évoquer les perspectives de notre travail.

## 2 Existant

### 2.1 Nomenclature

Dans cet article nous notons $\mathcal{A}$ l'ensemble des agents. Chaque agent $Agt \in \mathcal{A}$ est constitué d'un ensemble d'actions et d'un ensemble de données : $Agt = \langle A_{Agt}, D_{Agt} \rangle$. L'ensemble des données $D_{Agt}$ est l'ensemble des couples $\langle var, val \rangle$, où $var$ est le nom de la donnée et $val$ sa valeur. Une variable peut représenter soit une donnée observée par l'agent dans l'environnement, soit une croyance acquise suite à une interaction avec un autre agent.

L'état d'un agent est caractérisé par l'ensemble des données contenu dans $D_{Agt}$, il sera noté $s$ dans la suite de cet article et l'ensemble des états possibles d'un agent sera noté $S_{Agt}$. Nous noterons aussi $V_{Agt}$ l'ensemble des symboles de variables associées.

Une action est un tupple $a = \langle name, \mathcal{P}, \mathbf{E} \rangle$ où :
- $name$ est le nom de l'action. Il est unique pour un agent dans le sens où deux actions du même agent ne peuvent avoir le même nom.
- $\mathcal{P}$ est l'ensemble des préconditions nécessaires pour rendre valide l'action. Ces préconditions sont construites à partir de l'ensemble des données de l'agent $D_{Agt}$. Ces préconditions doivent être vraies pour que l'action puisse être exécutée par l'agent. Nous dirons qu'une action est possible, si l'ensemble de ses préconditions est vérifié, impossible sinon. La syntaxe précise des préconditions n'est pas importante pour la compréhension de cet article. Seul l'ensemble des variables impliquées dans la précondition sera utilisé dans nos algorithmes.
  Nous notons $vars(p)$ l'ensemble des variables impliquées dans la précondition $p \in \mathcal{P}$. Ces variables seront utilisées dans notre algorithme et nos protocoles.
- $\mathbf{E}$ est l'ensemble des effets de l'action, c'est-à-dire les propositions positives ou négatives

construites sur l'ensemble des données de l'agent $D_{Agt}$ qui seront vraies après l'exécution de l'action. Comme pour les préconditions, nous notons $vars(e)$ l'ensemble des variables utilisées pour définir un effet donné $e \in \mathbf{E}$.

Soit $s$ l'état de l'agent, nous notons $A_{Agt}^{(s)}$ l'ensemble de toutes les actions possibles dans cet état. Outre les actions de l'agent $A_{Agt}$, nous construirons un ensemble d'actes de communication qui sera considéré comme un ensemble d'actions possibles pour l'agent dans notre algorithme.

### 2.2 Actes de communication

Dans notre modèle un acte de communication est un message au sens de FIPA [FIP02, FIP97]. L'apprentissage du langage et de l'ontologie étant un problème à part entière nous ne le traiterons pas ici et nous considérerons que les agents utilisent le même langage de contenu et la même ontologie. Pour la compréhension de cet article, nous supposerons que le langage de contenu est une logique propositionnelle d'ordre 0 utilisant les opérateurs $\neg, \cup, \cap, =$ sur deux ensembles d'atomes (les variables $V_{Agt}$ et les actions des agents $A_{Agt}$) et un ensemble de fonctions arithmétiques. Un contenu de message est donc un ensemble (éventuellement vide) de propositions. Dans la plupart des cas, les propositions seront réduites à un atome (c'est-à-dire un symbole appartenant à l'ensemble $A_{Agt} \cup V_{Agt}$). Enfin, pour alléger les notations nous ne représenterons pas les paramètres de contrôle des messages (reply-with, conversation-id).

Un message est donc caractérisé par un tupple $\langle snd, p(rcv, c)\rangle$, avec $snd \in \mathcal{A}$ l'émetteur du message , $rcv \in \mathcal{A}$ son destinataire, $p$ le performatif utilisé et $c$ le contenu du message dont la nature (c'est-à-dire les éléments propositionnels) dépend de $p$.

Dans notre étude, nous nous limiterons à la communication pour demander de l'information sur des états ou demander à un autre agent de faire quelque chose. Dans ce but nous proposons d'utiliser deux performatifs différents : query et request.

**Query.** Les messages de la forme

$$m = \langle snd, query(rcv, v)\rangle$$

permettent à l'agent $snd$ de demander à l'agent destinataire $rcv$ la valeur de la variable $v \in D_{snd}$. Il correspond au message suivant dans la norme FIPA :

$$\langle snd, query{-}ref(rcv, Ref(x).value(rcv, v, x))\rangle$$

où la proposition $value(rcv, var, val)$ est vraie si et seulement si $(var, val) \in D_{rcv}$. L'agent $rcv$ peut alors répondre avec soit :

- $\langle rcv, inform(snd, v = v_0)\rangle$ pour affirmer que $(v, v_0) \in D_{rcv}$, comme il est défini dans FIPA.
- $\langle rcv, unknown(snd, v)\rangle$ pour affirmer que la variable $v$ n'appartient pas à l'ensemble $D_{rcv}$. Cela correspond selon la norme FIPA au message suivant : $\langle rcv, not - understood(snd, \langle snd, m\rangle)\rangle$ avec $m$ le message initialement envoyé.

**Request.** Les messages de la forme

$$m = \langle snd, request(rcv, a)\rangle$$

permettent à l' agent $snd$ de demander à l'agent $rcv$ d'exécuter l'action $a$, représentée par son nom[1]. Sa sémantique suit la définition selon la norme FIPA du performatif ***request***. L'agent $rcv$ peut alors répondre à l'aide d'un des trois messages suivants :

- $\langle rcv, agree(snd, a)\rangle$ pour répondre que l'agent $rcv$ accepte d'exécuter l'action $a$
- $\langle CV, impossible(snd, pe)\rangle$ pour répondre que l'agent $rcv$ ne peut pas exécuter l'action car au moins une des préconditions de cette action est fausse. $pe$ est l'ensemble de ces préconditions échouées. Cela se traduit en norme FIPA par le message suivant : $\langle rcv, refuse(snd, (\langle rcv, a\rangle, pe))\rangle$
- $\langle rcv, not - understood(snd, a)\rangle$ pour répondre que l'agent $rcv$ ne peut pas exécuter l'action $a$ puisqu'il ne connaît pas cette action ($a \notin A_{snd}$). Ce qui correspond au message FIPA suivant : $\langle rcv, not - understood(snd, m)\rangle$ où $m$ est le message initial.

### 2.3 Q-Learning

Le Q-Learning [Wat89] est un algorithme d'apprentissage par renforcement qui repose sur l'utilisation d'une fonction de valeur $Q : S_{Agt} * A_{Agt} \longrightarrow \mathbf{R}$ où $Q(s, a)$ correspond à la récompense attendue par l'agent $Agt \in \mathcal{A}$ lorsqu'il se trouve dans l'état $s$ et qu'il exécute l'action $a$.

[1] Dans ce papier, le nom d'une action comprend à la fois la référence de l'action ainsi que la valeur de ses paramètres. Par conséquent les actions prendre(a) et prendre(b) sont deux actions différentes

La meilleure action pour chaque état $s$ est alors définie par : $a^* = argmax_{a \in A_{Agt}} Q(s,a)$.

Le principe du Q-Learning est de construire itérativement la fonction $Q$ pour chaque couple $(s,a)$ en expérimentant les actions dans l'environnement. A chaque prise de décision (c'est à dire choix d'une action à effectuer), l'agent choisit une action $a$ dans $A_{Agt}^{(s)}$, exécute $a$, reçoit une récompense $r(s,a)$ et observe son nouvel état $s'$. Puis il met à jour la valeur Q(s,a) en fonction de $s'$ et de $r(s,a)$ obtenue selon la formule suivante :

$$Q(s,a) = (1-\alpha_t)Q(s,a)+\alpha_t(r+\gamma \max_{a \in A} Q(s',a'))$$

où $alpha_t \in [0,1]$ est le taux d'apprentissage et $\gamma$ est le facteur d'actualisation (Il permet de modéliser les préférences de l'agent en matière de récompense. Si l'agent préfère une récompense immédiate $\gamma$ sera proche de 0, si au contraire toutes les récompenses sont importantes pour lui $\gamma$ tendra vers 1).

La stratégie de sélection utilisée pour l'action à exécuter est la température de Boltzman où la probabilité de sélectionner l'action $a$ dans l'état $s$ est égale à :

$$P(a_t = a | s_t = s) = \frac{e^{Q(s,a)/T_t}}{\sum_{b \in A} e^{Q(s,b)/T_t}}$$

Avec $T_t$ le paramètre de température qui décroît doucement en fonction du temps t. Lorsque la température au départ est très élevée alors la probabilité de choisir une action $a$ ne dépend pas de $Q(s,a)$. Elle est donc uniforme. Au contraire lorsque la température est plus faible et proche de 0, la probabilité de choisir l'action $a$ dépend de la valeur de $Q(s,a)$. Ainsi une action avec un $Q$ élevé aura plus de chance d'être choisie.

Une des limites du Q-Learning par rapport à notre problème est qu'il est fondé sur le modèle des PDMs dans lesquels chaque action est exécutée sur un pas de temps. Au contraire les SMA sont généralement asynchrones. Ainsi les actes de communication prendront plus d'un pas de temps à s'exécuter.

### 2.4 Utilisation de la mémoire

La mémoire d'un agent consiste pour un agent à se souvenir de ses actions et de ses observations passées. Dans un PDM il n'existe pas de notion de mémoire car la propriété de Markov est vérifiée :

$$\forall s' \in S_{Agt} \qquad P(s_{t+1} = s' | s_t, a_t) = P(s_{t+1} = s' | s_t, a_t, s_{t-1}, .., s_1, a_1)$$

L'état de l'agent ne dépend que de l'état précédent et de la dernière action exécutée. Dans un PDMPO les techniques de reconnaissance d'état doivent utiliser de la mémoire afin de distinguer deux états différents qui correspondent à la même observation selon le point de vue de l'agent [Ber95]. Au lieu de définir son état sur ses seules observations, l'agent utilise aussi sa mémoire qui contient ses actions et ses observations passées. Si l'agent utilise assez de mémoire alors le problème peut redevenir markovien. Cependant l'utilisation de la mémoire peut conduire à une explosion combinatoire due à une augmentation du nombre d'états. La méthode proposée par [Mcc96, DS03, Dut00] est de trouver la taille minimale de la mémoire à utiliser pour que l'algorithme d'apprentissage converge.

L'idée générale est d'utiliser un arbre de décision. Cet arbre a pour feuilles les états possibles de l'agent c'est à dire les informations qu'il doit utiliser pour prendre ses décisions. Chaque branche de l'arbre correspond à une distinction introduite afin de distinguer du noeud père les états cachés. Les branches partant de la racine représentent une distinction sur les observations actuelles de l'agent, puis le second niveau de branche correspond à une distinction sur la dernière action effectuée, le troisième niveau à une distinction sur la dernière observation de l'agent et ainsi de suite comme le montre la figure 1.

L'arbre se construit itérativement :
- Au départ l'agent n'utilise pas de mémoire. Il construit la première série de fils (qui correspondent aux noeuds de profondeur 1) en utilisant les états qu'il observe.
- Après Niter expériences (Niter est fixé expérimentalement), on détermine les N états les plus ambigus. On ajoute alors un cran de mémoire à l'agent qui va lui permettre de mémoriser la dernière action qu'il a effectuée. L'état de l'agent est alors caractérisé par $S + M_1$ et on recommence l'apprentissage, mais on ne prend en compte $M_1$ que dans les états ambigus (dans les autres états on continue le Q-Learning classique avec $S$)
- On itère ce processus. Après $k$ étapes , la mémoire $M_k$ (utilisée pour les noeuds ambigus) contient $k$ éléments : $S$ , action, $S$ , action...

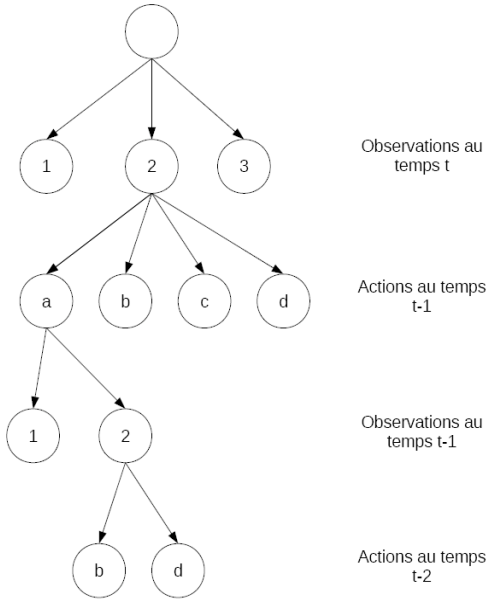



FIG. 1 – Un arbre de décision : Les observations de l'agent sont représentées par des entiers et les actions par des lettres.

Afin de déterminer quand introduire une nouvelle distinction sur la mémoire, [DS03] propose d'utiliser les trois critères suivants :

**Ambiguïté sur l'action optimale :** Un état $s$ est plus ambigu si l'action optimale $a_s^*$ n'est pas clairement définie, car les valeurs des $Q(s,.)$ des deux meilleures actions sont très proches.

**Nombre de mises à jour de la fonction Q :** Un état $s$ est plus ambigu qu'un autre état si sa fonction $Q(s,.)$ est mise à jour plus souvent. En effet un noeud mis à jour souvent correspond à un noeud souvent rencontré donc susceptible de correspondre à plusieurs états cachés.

**Convergence des Q(e,.) :** [DS03] ont noté que la convergence de la fonction Q est plus lente pour un état ambigu que pour un état non ambigu. Par conséquent, les auteurs enregistrent pour chaque action $a$ la dernière variation de la valeur $Q(s,a)$ et l'état $s$ est considéré plus ambigu quand les variations $Q(s,.)$ sont importantes.

Chaque état est classé selon chaque critère et les auteurs considèrent la somme des rangs obtenus pour définir les éléments les plus ambigus. Ensuite les N états les plus ambigus sont alors distingués en ajoutant un cran de mémoire.

L'algorithme se termine soit parce que la mémoire de l'agent a atteint une valeur seuil, soit parce que la différence entre les performances des deux derniers arbres n'est pas significative. La performance d'un arbre correspond à la récompense moyenne obtenue par l'agent lorsqu'il utilise la politique induite par l'arbre. Cette solution semble très intéressante pour traiter des problèmes non-markoviens. Cependant elle nécessite que l'agent connaisse l'ensemble des actions qu'il peut effectuer pour chaque état $s$. Dans notre problème ce n'est pas le cas, l'agent devant construire ses actes de communication au fur et à mesure de ses besoins.

# 3 Apprentissage des actes de communication

Nous présentons dans cette section notre algorithme d'apprentissage d'actes de communication. Notre solution prend en compte les hypothèses d'observabilité partielle et d'asynchronisme liées aux SMA. Premièrement nous montrons comment l'agent peut déterminer les contenus des messages, en utilisant des protocoles simples. Ensuite nous présentons notre mécanisme d'apprentissage par renforcement des actions et des actes de communication de l'agent. Nous montrons alors comment les agents peuvent apprendre à attendre les effets de leurs messages directifs. Enfin nous montrons comment notre algorithme peut être adapté pour traiter l'hypothèse de non-markovité en désambiguïsant les observations de l'agent.

## 3.1 Construction des actes de communication

Dans cet article, nous avons choisi dans un premier temps de nous limiter à la construction de messages d'acte *query* et *request*. La première difficulté de l'apprentissage par renforcement est de déterminer les contenus qui ne sont pas donnés à priori dans un SMA ouvert. Pour cela nous proposons de déterminer deux protocoles d'interaction simples qui reposent sur deux hypothèses. Premièrement les agents sont coopératifs dans le sens où ils effectuent les actions qui leur sont demandées via un message *request* (sous la condition que les préconditions de l'action soient vérifiées) et ils répondent correctement (ils ne mentent pas) lorsqu'ils connaissent la réponse d'un message *query*. Deuxièmement les agents sont capables d'introspection : ils sont capables de raisonner sur leurs actions et leurs données. L'idée générale est alors que l'agent va d'abord " explorer " le SMA à l'aide des

deux protocoles afin de construire un maximum d'actes de communication et tester les messages qu'il en déduit. Le dilemme exploration/exploitation est géré via un facteur $\delta$ qui tend à décroître lorsque l'exploration ne conduit plus à un résultat. Ce facteur est différent de la température utilisée pour déterminer le choix de l'action à utiliser dans un état. Le facteur $\delta$ lui détermine si l'agent doit continuer à construire des actions ou apprendre à utiliser ses actions pour un état donné.

**Construction des messages *Request*.** Pour construire les contenus des messages *request*, nous utilisons le protocole *what-order* illustré sur la figure suivante 2. Le performatif

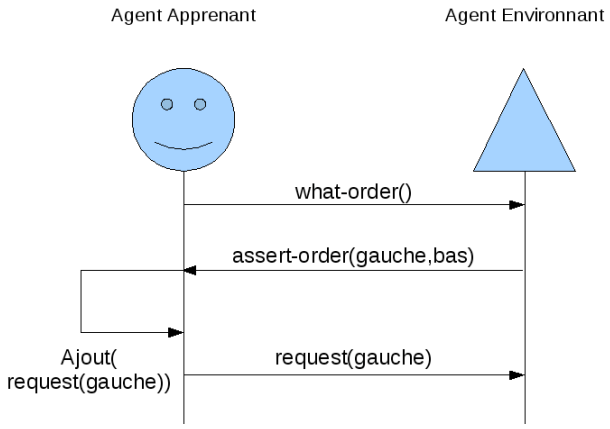



FIG. 2 – Protocole what-order

*what-order*, qui est utilisé avec un contenu vide, a été proposé dans le cadre de travaux antérieurs [CS07, Sab02]. Le message $\langle Agt, what\text{-}order(d, \emptyset)\rangle$ correspond au message suivant selon la nomenclature FIPA : $\langle snd, query(rvc, Ref(x).PossAct(x, rcv))\rangle$ où $PosAct(x, rcv)$ est vraie si $x$ est une action possible de l'agent $rcv$ au moment de la réception du message. Ce message permet à $snd$ de demander à l'agent $rcv$ l'ensemble de ses actions possibles.

La réponse attendue est un message de la forme :

$$m' = \langle d, assert - order(Agt, \langle a_1, .., a_n\rangle)\rangle$$

où $\forall i \quad a_i \in A_{rcv}$. A partir de cette réponse l'agent $snd$ peut construire un ensemble d'actes de communication. $\forall a_i \in contenu(m_rep)$, le message $\langle snd, request(rcv, a_i)\rangle$ est un acte de communication envisageable. Du point de vue de l'implémentation l'agent ajoute simplement ces actions à son ensemble d'actions $A^{(s)}_{Agt}$ ($s$ étant l'état de l'agent $snd$ au moment où le protocole *what-order* a été initié).

**Construction des messages *Query*.** Nous construisons le contenu des messages *query* en utilisant le contenu des messages *impossible* lors de l'échec d'un message *request* :

$$m' = \langle rcv, impossible(snd, pe)\rangle$$

Ce message (comme nous l'avons montré dans la section 2) contient $pe$ l'ensemble des préconditions de l'action demandée qui ont échoué. Chaque précondition $p \in pe$ contient un ensemble de variable $vars(p)$ qui peuvent être utilisées pour construire de nouveaux messages *query* : $\forall p \in pe$, $\forall v \in vars(p)$, le message $\langle snd, query(rcv, v)\rangle$ est un nouvel acte de communication envisageable (l'agent $snd$ peut l'ajouter à l'ensemble des actions $A^{(s)}_{Agt}$). L'utilisation des messages de performatif *impossible* est illustrée sur la figure 3.

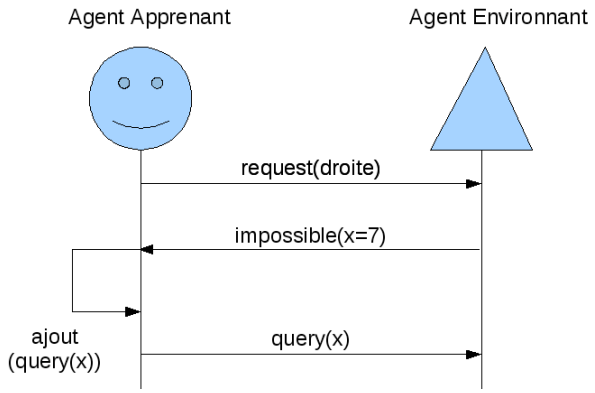



FIG. 3 – Utilisation d'un message de performatif impossible pour construire un nouvel acte de communication

**Algorithme d'exploration.** A chaque prise de décision de l'agent, une variable $Rand$ est générée aléatoirement :

– Si $Rand < \delta_s$ avec $s$ l'état de l'agent, l'agent utilise un protocole *what-order* pour déterminer de nouveaux actes de communication[2]. La valeur du paramètre $\delta_s$ est immédiatement corrigée en fonction de la réponse obtenue. Si l'agent reçoit un message *assert-order* contenant au moins une nouvelle action alors la valeur de $\delta_s$ est augmentée : $\delta_s = \delta_s * 2$ (on notera que $\delta_s$ étant une probabilité nous limitons sa plus haute valeur à 1). Sinon, $\delta_s$ est décrémenté : $\delta_s = \delta_s * 0.8$

[2]Le receveur du message est choisi selon une distribution uniforme

– Si $Rand \geq \delta_s$, l'agent choisit une action à effectuer parmi ses actions (actions propres et actes de communication) en utilisant la température boltzmanienne $T$.

### 3.2 Mécanisme d'apprentissage

Afin que l'agent apprenne à utiliser ses actions c'est à dire ses actions propres ainsi que les actes de communication qu'il a construit, nous utilisons l'algorithme de Q-Learning classique. Cependant cette approche pose trois problèmes.

Premièrement dans un SMA les agents sont asynchrones, les temps de réponse aux actes de communication peuvent être variables. Par conséquent les actions de l'agent apprenant ont des temps d'exécution différents. Or les formalismes des PDMs et des PDMPOs considèrent les actions comme s'exécutant sur un pas de temps fixe et identique pour toutes les actions de l'agent. C'est pourquoi nous utilisons le formalisme des PDSMs qui permet de modéliser un PDM où les actions ont des durées variables d'exécution et se résout via un algorithme de Q-Learning modifié prenant en compte le temps d'exécution de l'action.

Deuxièmement il faut anticiper l'absence de réponse des agents (dû soit à la perte des messages, soit à l'autonomie des agents dans les SMA qui peuvent prendre la décision de ne pas répondre). Pour éviter que l'agent n'attende une réponse indéfiniment, nous utilisons un time-out au delà duquel l'agent considère le message comme perdu.

Troisièmement la réception d'un message d'acte *agree* signifie que l'agent émetteur de la réponse accepte d'effectuer l'action demandée et non qu'elle est finie d'être exécutée. Dans le protocole FIPA-Request, ce problème est résolu par l'utilisation d'un message de performatif *confirm* pour indiquer que l'action est terminée [FIP03]. Cependant cette approche possède plusieurs limites. D'une part cette solution requiert l'envoi de nouveaux messages (lesquels peuvent être coûteux). D'autre part, l'utilisation d'un message *confirm* dans notre algorithme impliquerait que l'agent ne pourrait rien faire jusqu'à ce qu'il ait reçu la confirmation que l'action demandée a été effectuée ce qui est en totale contradiction avec le paradigme SMA. De plus ce message pourrait se perdre.

**Utilisation des PDSMs.** Un processus décisionnel semi markovien (PDSM) est un PDM dans lequel une action peut durer plus d'un pas de temps [Ber95, Put94]. On le définit par un quintuplet $\langle S, A, p, r\rangle$ où :
– $S$ est un ensemble d'états.
– $A$ est un ensemble d'actions.
– $P(\tau, s'|s, a)$ une fonction de transition qui décrit la probabilité de se trouver dans l'état $s'$ au bout de $\tau$ pas de temps si l'agent est dans l'état $s$ et qu'il effectue l'action $a$.
– $r : S*A \Longrightarrow \mathbb{R}$ est la fonction de récompense.
L'algorithme du Q-Learning peut alors s'appliquer en utilisant une formule adaptée. En effet la mise à jour de la fonction Q(s,a) ne se fait pas seulement en fonction du nouvel état $s'$ et de la récompense finale obtenue, mais en tenant compte de chaque récompense obtenue à chaque pas de temps compris entre le début et la fin de l'action [BD95]. Cependant chacune de ces récompenses est modérée en fonction du temps mis pour l'obtenir. Ainsi les récompenses obtenues en début d'action sont plus intéressantes que celles obtenues en fin d'action ceci visant à privilégier entre deux actions obtenant les mêmes récompenses la plus rapide. La mise à jour de la valeur $Q(s, a)$ se fait alors selon la formule suivante :

$Q(s_t, a) = (1 - \alpha_t) * Q(s_t, a) + \alpha_t * [\gamma^k \max_{a' \in A'(s_{t+k})} Q(s_{t+k}, a') + r_{t+1} + \gamma r_{t+2} + .. + \gamma^{k-1} r_{t+k}]$ où :
– $k$ est le nombre de pas de temps nécessaire à l'exécution de l'action
– $A'(s_{t+k})$ est l'ensemble des actions au temps $t + k$
– $r_t$ correspond à la récompense obtenue au temps $t$
– Les paramètres $\alpha_t$ et $\gamma$ sont identiques à ceux utilisés dans le Q-Learning classique présenté dans la section 2.3

**Gestion du time-out.** Afin de gérer le problème de non-réponse aux actes de communication nous utilisons un time-out. Si l'agent ne reçoit pas de réponse au delà de cette limite, il stoppe l'action et il ne modifie pas la Q-Value de ce message, parce qu'il suppose que son message a été perdu.

A chaque prise de décision, l'agent choisit une action $a$ à effectuer et le compteur $k$ qui stocke le nombre de pas de temps depuis lequel l'agent a exécuté son action est initialisé à 0. A chaque pas de temps, k est incrémenté de 1 jusqu'à ce que :
– k>TLIM. Dans ce cas l'agent arrête son action.

– L'action $a$ est terminée (une réponse a été obtenue). L'agent modifie alors la valeur $Q(s, a)$ (où $s$ est l'état de l'agent lorsqu'il a lancé l'action $a$).

**Apprendre à attendre.** Lorsque l'agent reçoit un message de performatif *agree*, cela n'implique pas que l'action demandée est terminée mais seulement que l'agent questionné accepte d'exécuter l'action demandée. Cependant notre agent apprenant ne peut pas utiliser la solution proposée par FIPA [FIP03] basée sur un message de performatif *confirm* pour plusieurs raisons. Premièrement, les messages *confirm* peuvent être coûteux. Deuxièmement, il faudrait utiliser un deuxième time-out pour gérer le problème de non réponse de la part de l'agent. Troisièmement, dans beaucoup de situations, si les effets intéressants de l'action se produisent avant que le message *confirm* soit réceptionné (soit parce que l'action est complexe et nécessite plusieurs sous effets dont seulement certains intéressent l'agent apprenant, soit parce qu'un autre agent change l'état de l'environnement entre temps), notre agent perdra un temps précieux à attendre un message *confirm* inutile. Il devrait au contraire utiliser ses observations et lancer une autre action.

Pour ces raisons, nous proposons d'utiliser une autre solution : une action *wait* = $\langle wait, \emptyset, \emptyset \rangle$ qui est ajoutée à l'ensemble des actions possibles de l'agent et qui lui permet d'attendre les effets des actions *request*. Notre agent devra donc non seulement apprendre à agir et à communiquer mais aussi à attendre, en utilisant le Q-Learning adapté aux PDSMs (voir la section 3.2.1). Dans notre algorithme, nous donnons un coût nul à l'action *wait* et un coût négatif à l'envoi d'un message[3] afin de privilégier l'attente à l'envoi d'un message. Enfin l'utilisation de l'action *wait* contrairement à l'utilisation du message *confirm* permet à l'agent de lancer d'autres actions (au lieu d'attendre la fin de celle qui est lancée) et ainsi permettre que le système exécute plusieurs actions en parallèle.

### 3.3 Gestion de la non markovité

De façon générale dans un système multi-agents la propriété de Markov n'est plus respectée. D'une part parce que les agents ne peuvent toujours observer totalement l'environnement. D'autre part parce que les autres agents agissent aussi sur l'environnement en effectuant leurs propres actions. Comme nous envisageons le problème d'un seul agent apprenant alors l'ensemble des actions effectuées par les autres agents sont assimilables à des états de l'environnement que l'agent ne peut voir. C'est pourquoi nous pouvons modéliser notre problème par un PDMPO.

Pour faciliter la convergence de l'algorithme du Q-Learning dans notre PDMPO, notre idée est d'utiliser les messages de performatif *query* pour obtenir des informations sur l'environnement et les données des autres agents. Cette information est stockée sous forme de croyance dans notre modèle. Cependant, chaque croyance peut devenir fausse durant le processus d'apprentissage, notre agent doit donc apprendre combien de temps il doit garder une croyance donnée.

Pour faire cela, nous avons choisi d'utiliser l'algorithme d'apprentissage que nous avons introduit à la section 2.4. Le principe général est le suivant : chaque croyance reçue après l'envoi d'un message *query* est ajoutée à l'état de l'agent pour la prochaine prise de décision seulement puis elle est effacée immédiatement après. Cependant, puisque l'état de l'agent inclus ses observations et ses croyances elle n'est pas perdue. Par conséquent en utilisant l'algorithme présenté à la section 2.4, l'agent apprend le nombre de pas de temps depuis lequel il a reçu une croyance au lieu de combien de temps il doit s'en souvenir.

Une autre motivation pour choisir d'utiliser la mémoire de l'agent est qu'elle permet à l'agent de se souvenir de ses actions passées. En utilisant cette information, l'agent peut apprendre, par exemple, qu'un acte de communication *query* est plus pertinent après un acte de communication *request*.

De plus l'utilisation de la mémoire permettra d'éviter des situations de blocage dues à l'utilisation de l'action *wait*. En effet si dans un état $s$ il est pertinent d'attendre car l'agent vient de lancer une action $a$, cela l'est beaucoup moins si l'agent vient d'attendre au coup précédent. Sans mémoire l'agent apprendra à attendre dans l'état $s$ et pourra donc rester bloqué dans cet état si celui n'est pas modifié par une aide extérieure (action lancée auparavant ou un autre agent modifiant l'environnement). Au contraire en utili-

[3] Ce coût sera fonction de la réponse obtenue, un message obtenant une réponse "positive" (c'est-à-dire la réception d'un message *agree* pour l'envoi d'un message *request* ou la réception d'un message *inform* pour l'envoi d'un message *query*) sera moins pénalisé qu'un message obtenant une réponse "négative"

sant la mémoire l'agent pourra apprendre par exemple à attendre après un acte *request* et pas après une action *wait*. Comme nous venons de le voir un noeud où la meilleure action est *wait* est un noeud très ambigu par conséquent nous prendrons en compte cette information dans le choix des noeuds les plus ambigus.

**Algorithme d'utilisation de la mémoire.** L'agent apprenant $A_l = \langle Agt, \mathcal{M} \rangle$ est un agent $Agt \in \mathcal{A}$ tel que présenté dans la section 2.1 avec une mémoire $\mathcal{M} = \{a_t, s_t, a_{t-1}, s_{t-1}, ..\}$ où $a_t \in A_{Agt}^{(s_t)}, s_t \in S_{Agt}$.

Comme dans [Mcc96] nous utilisons un arbre représentant les états sur lesquels l'agent va faire son apprentissage. Chaque noeud de l'arbre est représenté par un tupple :

$$Nd = \langle Di, A_{Agt}^{(Nd)}, Q_{Nd}, \Delta Q_{Nd}, UpD_{Nd}, \delta_{Nd} \rangle$$

- $Di$ est une distinction. Elle est égale à la valeur de la branche qui arrive dans le noeud $Nd$. $Di \in \{S_{Agt}, A_{Agt}\}$.
- $A_{Agt}^{(Nd)}$ est l'ensemble des actions que l'agent peut exécuter quand il est dans l'état représenté par le noeud $Nd$. Cet ensemble contient des actes de communication et l'action *wait*.
- $Q_{Nd}$ est l'ensemble des valeurs $Q(Nd, .)$. $Q_{Nd} = \{q_a\}$ avec $a \in A_{Agt}^{(Nd)}$
- $\Delta Q_{Nd} = \{\Delta q_a | \Delta q_a(t+1) = |q_a^{t+1} - q_a^t|$ avec $a \in A_{Agt}^{(Nd)}$ et $q_a^t, q_a^{t+1} \in Q_{Nd}\}$.
- $UpD_{Nd}$ est le nombre de mises à jour du noeud $Nd$, c'est-à-dire le nombre de fois où l'agent a été dans létat représenté par $Nd$ et qu'il l'a mis à jour durant le processus d'apprentissage.
- $\delta_{Nd}$ est le paramètre d'exploration. Il définit la probabilité que l'agent apprenant choisisse d'exécuter un protocole *what-order* dans l'état $Nd$.

L'ensemble des arbres est noté $Arb$. Un arbre $arb$ est représenté par un tupple $\langle Nd, \mathcal{F}_{arb} \rangle$ où $\mathcal{F}_{arb} \subset Arb$ est l'ensemble des arbres fils de $arb$.

A la fin de l'étape d'apprentissage, on détecte les N noeuds les plus ambigus. Un noeud sera plus ambigu qu'un autre si sa meilleure action est *wait*. En cas d'ex-aequo, notre algorithme se basera sur l'heuristique de Dutech et al. [DS03] pour départager les noeuds.

- Le taux d'ambiguïté d'un noeud $s$ est déterminé par la fonction suivante :

$$\begin{aligned} ambigu(s) &= wait(s) \\ &+ \frac{1}{3}(rang_s(UpD_s) + \frac{1}{|\Delta Q_s|} \\ &+ rang_s(\frac{1}{|\Delta Q_s|} \sum_{a \in A_{Agt}^s} \Delta q_a) \\ &+ rangD_s(q^{A_s^1} - q^{A_s^2})) \end{aligned}$$

  Avec $A_s^1, A_s^2$ les deux meilleures actions pour l'état $s$ (c'est-à-dire les actions dont la valeur $q$ est la plus elevée), la fonction $rang_s(x)$ qui renvoie la position du noeud $s$ si on ordonne tous les noeuds dans l'ordre croissant en fonction de $x$, $rangD_s(x)$ renvoie la position du noeud $s$ si on ordonne tous les noeuds dans l'ordre décroissant en fonction de $x$ et $wait(s)$ qui renvoie 0 si l'action *wait* est la meilleure action pour le noeud $s$ et qui renvoie M (M étant très grand) sinon.
- Les $N$ noeuds les plus ambigus sont ceux qui possèdent le plus petit taux d'ambiguïté :

$$\begin{aligned} Nd_ambigu &= \{n | feuille(n, arb), \\ &\quad rang_n(ambigu(n)) < N\} \end{aligned}$$

De même que l'algorithme de Dutech et al., nous arrêtons notre algorithme soit parce que la taille maximale de la mémoire de l'agent est atteinte, soit parce que les performances de notre agent ne peuvent pas être améliorées par un nouveau cran de mémoire. Les performances de notre agent sont déterminées par la récompense moyenne obtenue à l'aide de la politique définie par l'apprentissage par renforcement sur les états courants pour un nombre $N$ d'expériences. L'algorithme s'arrête donc lorsque :

$arret() =$

1 Si $t > Mlim$ Ou
$|\frac{1}{N}(\sum_1^N R(arb_t) - \sum_1^N R(arb_{t-1}))| < \epsilon$

0 Sinon.

Avec $R(arb_t)$, la récompense obtenue par l'agent lorsqu'il utilise l' arbre $arb_t$ et $Mlim$ la taille limite de la mémoire de l'agent.

### 3.4 Implémentation et évaluation en cours

Notre algorithme d'apprentissage a été implémenté sur la plateforme multi-agent VDL[4], laquelle offre une capacité d'introspection adéquate pour notre modèle [Sab02]. Nous avons

[4] http ://www-poleia.lip6.fr :8180/ sabouret/demos/index.html

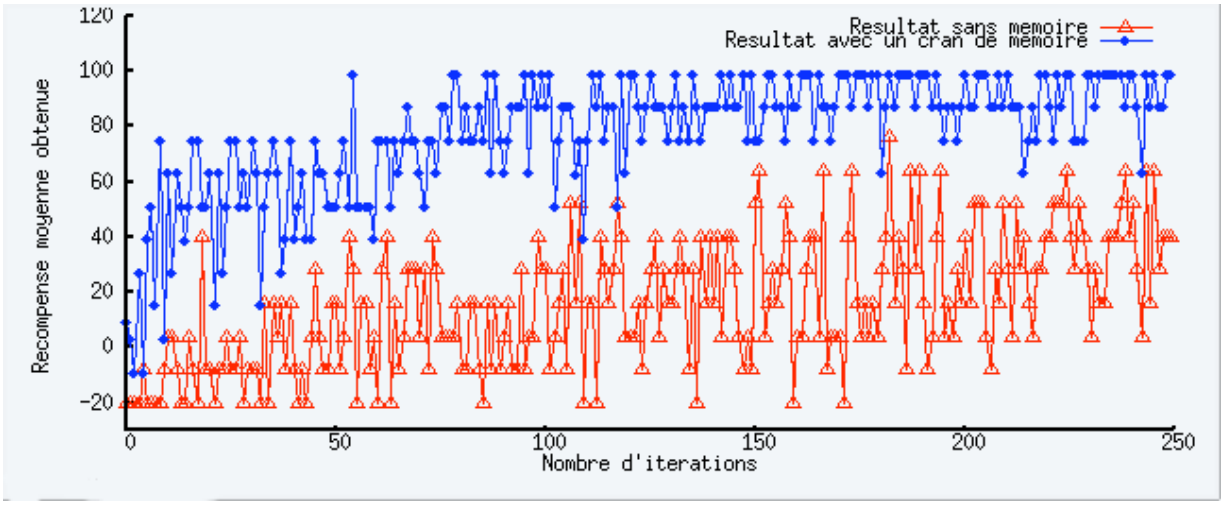



FIG. 4 – Récompense moyenne sur 10 expériences en fonction du nombre d'itérations de l'algorithme pour un agent sans mémoire (triangle) et avec un cran de mémoire (point)

programmé un exemple simple de SMA asynchrone fondé sur le problème de recherche d'un trésor dans un labyrinthe. Un robot se déplace dans un labyrinthe (5*5) afin de trouver un trésor tout en évitant un certain nombre de cases trous (25 % des cases du labyrinthe) pour cela il dispose de 4 actions (haut, droite, gauche, bas). A chaque expérience le labyrinthe n'est pas modifié et le robot est remis sur la même case de départ. L'agent apprenant doit apprendre à diriger le robot afin d'obtenir la récompense. Les deux agents sont asynchrones dans le sens où leurs vitesses d'exécution sont différentes ce qui ne permet pas de garantir que les commandes de l'agent apprenant soient prises en compte immédiatement. Cependant leur vitesse est constante et ne varie pas entre deux expériences. Nous avons choisi cet exemple car il est facile de le complexifier en fonction des modalités de notre algorithme que nous voulons tester (protocole *what-order*, apprentissage de l'action *wait*, utilisation de la mémoire...). Une première expérimentation a permis de valider notre algorithme dans le cas particulier où :

- L'apprenant apprend à utiliser des actes de communication *request* (qu'il a préalablement construit à l'aide du protocole what-order) et l'action *wait*.
- L'apprenant observe totalement son environnement.
- L'agent apprenant s'exécute deux fois plus vite que le robot.
- Les récompenses associées aux états sont les suivantes : le trou apporte une récompense finale de -20 et le trésor rapporte une récompense finale de 100. Les autres états ne rapportent pas de récompense.

Les résultats obtenus sont encourageants comme le montre le graphe ci dessous représentant les récompenses obtenues par l'agent apprenant au cours de son apprentissage en n'utilisant soit aucune mémoire (courbe avec des triangles) ou un cran de mémoire (courbe avec des points). On y voit clairement que l'utilisation d'un cran de mémoire permet à l'algorithme de converger vers une solution où la récompense moyenne finale est de 100, alors que sans mémoire l'agent stagne vers une récompense moyenne finale de 40.

Actuellement nous complétons notre modèle expérimental afin d'introduire la construction des actes *query* et leur utilisation dans un contexte d'observation partielle de l'environnement. Afin de valider nos résultats, nous comparerons notre solution à un algorithme d'apprentissage sans communication tel que celui de [DS03]. En étudiant la convergence de l'algorithme, le coût de la communication et le gain résultant de la politique trouvée, nous voulons démontrer que l'apprentissage d'actes de communication améliore vraiment la qualité du processus d'apprentissage. Cette partie est encore en cours de réalisation.

## 4 Conclusion

Nous nous sommes intéressés dans cette étude à l'apprentissage par renforcement d'acte de communication dans un contexte multi-agents. Dans cette optique nous avons défini trois types de problèmes liés à cette approche : la construction des actes de communication, l'assimilation

des actes de communication à des actions afin de pouvoir leur appliquer des techniques d'apprentissage par renforcement et enfin la résolution des problèmes posés par la non markovité des systèmes multi-agents. Afin de construire les actes de communication nous avons utilisé une solution qui utilise le pouvoir d'introspection des agents et leur capacité à raisonner sur leur actions afin que l'agent apprenant puisse construire les actes de communication dont il a besoin. Ces actes de communication ainsi créés sont alors considérés comme des actions prenant un temps variable afin de leur appliquer un algorithme d'apprentissage par renforcement. Notre algorithme gère le problème de non markovité en employant les informations récupérées par l'agent à l'aide des actes de communication qu'il a créé ainsi qu'en utilisant sa mémoire des actions et des observations passées.

Si les travaux présentés ici permettent d'améliorer l'apprentissage par renforcement dans un contexte multi-agents en proposant un mécanisme d'apprentissage d'acte de communication, ils soulèvent encore de nombreuses perspectives de recherche. Premièrement afin de pallier des problèmes d'explosion combinatoire en mémoires et en calculs, nous avons dû considérer que l'ensemble des actions apprises par notre agent lui étaient utiles comme cela se fait dans la plupart des travaux en apprentissage par renforcement. Dans un cadre plus réaliste, cette hypothèse ne tient plus et par conséquent il serait intéressant de poursuivre les travaux et de définir un mécanisme permettant à l'agent de ne garder en mémoire que les actions utiles à son problème.

Une autre direction de recherche serait de définir une heuristique pour l'utilisation de la mémoire. L'heuristique que nous utilisons est largement inspirée des travaux de [DS03] lesquels ont eu de bons résultats dans un SMA sans communication. En utilisant l'information fournie par la communication (c'est à dire le nombre d'actes de communication ayant échoué pour chaque état de l'agent), nous pensons que l'heuristique pourrait être améliorée. D'autre part il serait intéressant de définir un mécanisme permettant de déterminer de quels types de mémoires (actions ou observations), l'agent a besoin selon l'état où il se trouve.

Enfin nous aimerions par la suite étendre notre recherche à l'ensemble des actes de communication de l'agent afin de bénéficier de la richesse des langages de communication entre agents (ACL).

## Références


[Aus62] J. Austin. How to do things with words. *Oxford University Press, Oxford England*, 1962.

[BD95] Steven J. Bradtke and Michael O. Duff. Reinforcement learning methods for continuous-time markov decision problems. In *Advances in Neural Information Processing Systems 7*, pages 393–400. MIT Press, 1995.

[Ber95] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont , Massachusetts, 1995.

[CKL94] Anthony R. Cassandra, Leslie Pack Kaelbling, and Michael L. Littman. Acting optimally in partially observable stochastic domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, volume 2, pages 1023–1028, Seattle, Washington, USA, 1994. AAAI Press/MIT Press.

[CS07] Yasmine Charif and Nicolas Sabouret. An agent interaction protocol for ambient intelligence. In *2nd International Conference on Intelligent Environnement (IE06)*, 2007.

[DS03] Alain Dutech and Manuel Samuelides. Un algorithme d'apprentissage par renforcement pour les processus décisionnels de Markov partiellement observés : apprendre une extension sélective du passé. 2003.

[Dut00] Alain Dutech. Solving pomdps using selected past events. In *ECAI*, pages 281–285, 2000.

[FIP97] FIPA.org. Fipa communicative act library specification. 1997.

[FIP02] FIPA.org. Fipa acl message structure specification. 2002.

[FIP03] FIPA.org. Fipa request interaction protocol specification. 2003.

[Mat98] Maja J Matari'c. Using communication to reduce locality in distributed multi-agent learning. *Journal of Experimental and Theoretical Artificial Intelligence*, 10 :357–369, 1998.

[Mcc96] Andrew Kachites Mccallum. *Reinforcement learning with selective perception and hidden state*. PhD thesis, 1996. Supervisor-Dana Ballard.

[Put94] M.L. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc, New York, Etats-Unis, 1994.

[Sab02] Nicolas Sabouret. A model of requests about actions for active components in the semantic web. 2002.

[SB98] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. The MIT Press, 1998.

[Sea85] J. R. Searle. Speech acts : An essay in the philosophy of language. *Cambridge University Press : Cambridge, England*, 1985.

[SJJ94] Satinder P. Singh, Tommi Jaakkola, and Michael I. Jordan. Learning without state-estimation in partially observable markovian decision processes. In *International Conference on Machine Learning*, pages 284–292, 1994.

[Tan93] Ming Tan. Multi-agent reinforcement learning : Independent vs. cooperative agents. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 330–337. Morgan Kaufmann, 1993.

[Wat89] Christopher J. C. H. Watkins. *Learning from delayed rewards*. PhD thesis, Cambridge University, Cambridge, United Kingdom, 1989.

# Une axiomatisation des choix prudents

N. Houy*
nicolas.houy@polytechnique.edu

*Département d'Economie
Ecole Polytechnique
91128 Palaiseau Cedex – FRANCE

**Résumé :**
Nous définissons et axiomatisons les choix prudents pour deux critères. Soient deux critères. Les alternatives choisies par la procédure des choix prudents sont celles qui maximisent une composition des deux critères. Cette composition est telle que 1) elle contient le premier critère et un sous-ensemble du second, et 2) la nouvelle relation binaire n'est pas cyclique et ne peut pas être élargie par des préférences du second critère sans devenir cyclique. Nous faisons également le lien entre les choix prudents, les choix rationnels au sens classique, les choix séquentiellement rationnels ([11]) et les choix rationnels par composition lexicographique des relations de base ([15]).

**Mots-clés :** Choix prudents, rationalité, axiomatisation.

**Abstract:**
We define and axiomatize prudent choices for two criteria. Given two criteria, the alternatives chosen by the prudent choice procedure are the ones maximizing some composition of the criteria. This composition is such that 1) it contains the first criterion and a part of the second one, and 2) the new binary relation is not cyclic and cannot be enlarged with preferences of the second criterion without becoming cyclic. We also make the link between prudent choices, classical rational choices, sequentially rational choices ([11]) and lexicographic binary choice rational choices ([15]).

**Keywords:** Prudent choices, rationality, axiomatization.

## 1 Introduction

Des progrès importants ont été faits récemment dans l'analyse de la décision multi-critères. Pour résumer et discuter certains de ces progrès, considérons la situation suivante. Une société doit prendre une décision entre les trois alternatives $a$, $b$ et $c$. Pour le premier critère, disons l'efficacité, $a$ est meilleure que $b$. Pour le second critère, disons l'équité, $b$ est meilleure que $c$, $c$ est meilleure que $a$ et $b$ est meilleure que $a$. Les procédures suivantes ont été proposées par les théoriciens du choix pour choisir entre $a$, $b$ et $c$.

La première procédure a été introduite par [1] et [2], et plus récemment étudiée par [5], [11], [12] et [15]. Cette procédure est mise en œuvre comme suit. Dans un premier temps, la société fait un choix et présélectionne un sous ensemble des alternatives en maximisant le premier critère. Dans notre cas, quand le choix est entre $a$, $b$ et $c$, l'alternative $b$ est éliminée et ainsi, $a$ et $c$ sont préselectionnées. Ensuite, la société fait son choix entre les alternatives préselectionnées en maximisant le second critère. Dans notre cas, entre $a$ et $c$, $c$ est choisie finalement. Cette procédure de composition des deux critères est séduisante parce qu'elle mène à faire des choix non-vides sous des hypothèses très faibles. Cependant, il a été prouvé qu'elle mène très souvent à faire des choix qui ne satisfont pas les axiomes basiques de cohérence.

La seconde procédure proposée pour faire des choix dans le même cadre a été introduite par [15]. D'après cette procédure, la société construit une relation binaire comme suit : pour tout couple d'alternatives $x$ et $y$, $x$ est préférée à $y$ si et seulement si [$x$ est préférée à $y$ pour le premier critère] ou [$y$ n'est pas préférée à $x$ pour le premier critère et $x$ est préférée à $y$ pour le second critère]. Ensuite, la société prend sa décision en maximisant cette nouvelle relation binaire de préférences. Dans l'exemple donné plus haut, la composition des critères d'efficacité et d'équité donne les préférences suivantes : $a$ est préférée a $b$, $b$ est préférée à $c$ et $c$ est préférée à $a$. Dans ce cas, la société ne peut pas choisir entre $a$, $b$ et $c$. La raison pour cette incapacité est la suivante : cette procédure, par définition, empêche la société de faire face à des préférences présentant des cycles de taille deux. Cependant, il est possible, comme il est montré dans notre exemple, qu'il existe des cycles de préférences de taille au moins 3. La procédure que nous proposerons ne présente pas une telle possibilité. Ainsi, même si cette seconde procédure est attirante pour ses propriétés de cohérence, elle ne satisfait que difficilement à l'obligation de pouvoir faire des choix.[1]

Dans cet article, nous proposons une troisème procédure. Elle est mise en œuvre comme suit.

[1] Pour une étude plus approfondie des propriétés des deux premières procèdures, voir [5] et [6].

La société construit une nouvelle relation binaire de préférences en prenant en compte le premier critère d'abord. Ensuite, elle élargit cette nouvelle relation binaire de préférences en ajoutant autant d'éléments du second critère que possible sous la contrainte que les préférences construites soient acyclique dans l'ensemble des alternatives entre lesquelles doit choisir la société. Evidemment, la relation de préférences composée ainsi construite peut ne pas être unique et ainsi, nous définissons le choix final comme l'ensemble des alternatives qui maximise au moins une des relations de préférences construites précédemment. Ainsi, dans notre exemple, la société considère comme acquis que $a$ est préférée à $b$. Le fait que $b$ est préférée à $a$ par le second critère ne peut pas être pris en compte sinon les préférences seraient cycliques. Pour la même raison, les faits que $b$ est préférée à $c$ ET que $c$ est préférée à $a$ par le second critère ne peuvent pas être pris en compte simultanément sinon les préférences seraient aussi cycliques. Cependant le fait que $b$ est préférée à $c$ peut être pris en compte seul. Similairement, le fait que $c$ est préférée à $a$ peut être pris en compte seul. Ainsi, les relations de préférences composées sont : ($a$ préférée à $b$ et $b$ préférée à $c$) et ($c$ préférée à $a$ et $a$ préférée à $b$). Ainsi, le choix entre $a$, $b$ et $c$ est le sous-ensemble des alternatives $a$ et $c$. L'intuition est la suivante. Quand le second critère est considéré, *i.e.* quand il existe un risque que les préférences soient en contradictions avec le premier critère, alors, seulement un sous-ensemble des préférences du second critère est considéré. Cependant, ce sous-ensemble doit être aussi large que possible sous la contrainte que les préférences construites ne soient pas cycliques.

Nous appelons cette procédure "choix prudents" parce qu'elle est liée aux ordres prudents introduits dans [3] et étudiés plus récemment par [7], [9] et [10].[2] Intuitivement, dans un cadre de choix social avec des individus munis d'ordres linéaires, un ordre prudent est un ordre linéaire tel que l'opposition la plus forte contre lui est minimal. Ainsi, un ordre prudent peut être vu comme un ordre linéaire obtenu en composant plusieurs critères comme suit : d'abord, considérons la comparaison binaire avec l'accord le plus grand. Notons $P_1$ cette relation binaire. Ensuite, considérons la comparaison binaire avec le second accord le plus important, $P_2$. Si $P_1 \cup P_2$ est cyclique, alors les ordres prudents sont les extensions linéaires de $P_1$. Sinon, continuons la procédure avec $P_3$, la comparaison binaire avec le troisième accord le plus important. Si $P_1 \cup P_2 \cup P_3$ est cyclique, alors les ordres prudents sont les extensions linéaires de $P_1 \cup P_2$. Sinon, continuons la procédure avec $P_4$ et ainsi de suite. Contrairement à ces études menées dans un cadre de choix social, nous ne considérons que deux critères. Cependant, comme dans les études en théorie du choix rationnel, nous considérons les choix dans les sous-ensembles d'alternatives. Ainsi, la longueur de la procédure décrite ci-dessus dépend de l'ensemble de choix. Dans la mesure de nos connaissances, aucune axiomatisation a jamais été donnée de choix faits à partir de telles relations binaires prudentes ou par une règle de composition de plusieurs critères reliée à une composition prudente.

La suite de cette article est structurée comme suit. Dans la Section 2, nous introduisons les notations. Dans la Section 3, nous donnons une axiomatisation des choix prudents. Dans la Section 4, nous comparons les trois procédures données dans cette introduction. Enfin, dans la Section 5, nous comparons notre procédure à la rationalité des choix au sens classique.

## 2 Notations

Soit $X$ un ensemble fini d'alternatives avec au moins deux éléments. Soit $\mathcal{X}$ l'ensemble de tous les sous-ensembles de $X$ de taille au moins deux. Une correspondance de choix sur $X$ est une fonction $C : \mathcal{X} \to 2^X \setminus \emptyset$ telle que $\forall S \in \mathcal{X}, C(S) \subseteq S$.[3] Soit $\mathcal{C}(X)$ l'ensemble de toutes les correspondances de choix sur $X$.

Soit $C \in \mathcal{C}(X)$. Pour tout sous-ensemble d'alternatives $S \in \mathcal{X}$, nous définissons la relation de domination bipartite ($TD(S, C)$) comme suit : soient $R, T \subseteq S$ tels que $R \cap T = \emptyset$ et $R \cup T = S$. $(R, T) \in TD(S, C)$ si et seulement si $R$ et $T$ sont non-vides et $\forall r \in R, \forall t \in T, r \in C(\{r, t\})$. Ainsi, $R$ domine bipartitement $T$ dans $S$ pour $C$ si et seulement si $\{R,T\}$ est une partition de $S$ (avec $R \neq \emptyset$ et $T \neq \emptyset$) et pour le choix entre tout élément de $R$ et tout élément de $T$, l'élément de $R$ est toujours choisi.

[2] Voir aussi [8] pour un lien avec les "Ranked Pair Rules" qui ont la propriété remarquable d'être indépendantes des clones, [16], [17].

[3] Notons que, par définition, un choix est non-vide. De plus, par simplicité, nous ne considérons pas les choix depuis des singletons.

Soit $P \subseteq X \times X$ une relation binaire sur $X$. Nous disons que $P$ est asymétrique si et seulement si $\forall a, b \in X, (a, b) \in P$ implique $(b, a) \notin P$. Soit $S \in \mathcal{X}$. Nous disons que $P$ est acyclique sur $S$ si et seulement s'il n'existe pas d'entier naturel $n \geq 2$ et pas de $a_1, ..., a_n \in S$ tels que $\forall i \in \{1, ..., n-1\}, (a_i, a_{i+1}) \in P$ et $(a_n, a_1) \in P$.[4] Une relation de préférences est une relation binaire asymétrique. Soit $P \mid_S$ la restriction de $P$ sur $S$, *i.e* $P \mid_S = \{(a, b) \in P / a, b \in S\}$.

Soient $P_1$ et $P_2$ deux relations de préférences sur $X$. Nous définissons $Q(P_1, P_2)$ par $\forall a, b \in X$, $(a, b) \in Q(P_1, P_2)$ si et seulement si $(a, b) \in P_1$ ou [$(b, a) \notin P_1$ et $(a, b) \in P_2$].

Soit $(P_1, P_2)$ un couple ordonné de relations de préférences sur $X$ telles que $P_1$ est acyclique. Soit $S \in \mathcal{X}$. Nous disons que $P \subseteq S \times S$ est une composition prudente de $P_1$ et $P_2$ sur $S$ si et seulement si
- $P = P_1 \mid_S \bigcup Q$ avec $Q \subseteq P_2 \mid_S$,
- $P$ is acyclique et,
- $\forall Q'$ tel que $Q \subset Q' \subseteq P_2 \mid_S$, $P_1 \mid_S \bigcup Q'$ est cyclique sur $S$.

Ainsi, une composition prudente de $P_1$ et $P_2$ sur $S$ est une relation de préférences comprenant $P_1$ et autant d'éléments de $P_2$ que possible sous la contrainte que la composition prudente est acyclique. Par $\widehat{(P_1, P_2)}(S)$, nous notons l'ensemble de toutes les compositions prudentes de $P_1$ et $P_2$ sur $S$. Notons que par définition, $\widehat{(P_1, P_2)}(S)$ est non-vide quand il est bien défini, *i.e.* quand $P_1$ est acyclique sur $S$.[5] Notons également que par définition, $\forall P \in \widehat{(P_1, P_2)}(S)$, $P \subseteq Q(P_1, P_2)$.

Soit $P \subseteq X \times X$ une relation de préférences sur $X$. Par $C_P$ nous définissons

$$\forall S \in \mathcal{X}, C_P(S) = \{a \in S, \forall b \in S, (b, a) \notin P\}.$$

Si $C_P \in \mathcal{C}(X)$, nous disons que $P$ rationalise la correspondance de choix $C_P$. Soient $P_1, P_2 \subseteq X \times X$ deux relations de préférences sur $X$. Nous définissons $C_{(P_1,P_2)}$ comme $\forall S \in \mathcal{X}, C_{(P_1,P_2)}(S) = \{a \in S, \exists P \in \widehat{(P_1, P_2)}(S), \forall b \in S, (b, a) \notin P\}$. Si $C_{(P_1,P_2)} \in \mathcal{C}(X)$, nous disons que $(P_1, P_2)$ rationalise prudemment $C$.

[4] Notons que par définition, l'acyclicité implique l'asymétrie et l'asymétrie implique l'irréflexivité. De plus, quand nous dirons qu'une relation binaire est acyclique, nous signifierons qu'elle est acyclique sur $X$.

[5] Evidemment, si $\widehat{(P_1, P_2)}(S) = \{\emptyset\}$, nous avons bien $\widehat{(P_1, P_2)}(S) \neq \emptyset$.

Pour illustrer les notations et définitions données jusqu'ici, formalisons l'exemple donné dans l'introduction. Soit $X = \{a, b, c\}$. Soient $P_1 = \{(a, b)\}$ et $P_2 = \{(b, a), (b, c), (c, a)\}$. Evidemment, nous avons $\widehat{(P_1, P_2)}(\{a, b\}) = \{\{(a, b)\}\}$, $\widehat{(P_1, P_2)}(\{b, c\}) = \{\{(b, c)\}\}$, $\widehat{(P_1, P_2)}(\{a, c\}) = \{\{(c, a)\}\}$ et $\widehat{(P_1, P_2)}(X) = \{\{(a, b), (b, c)\}, \{(a, b), (c, a)\}\}$. Ainsi, la correspondance de choix $C_{(P_1,P_2)} \in \mathcal{C}(X)$ prudemmment rationalisée par $(P_1, P_2)$ existe et est telle que $C_{(P_1,P_2)}(\{a, b\}) = \{a\}$, $C_{(P_1,P_2)}(\{b, c\}) = \{b\}$, $C_{(P_1,P_2)}(\{a, c\}) = \{c\}$ et $C_{(P_1,P_2)}(X) = \{a, c\}$.

## 3 Axiomatisation

Pour simplifier la compréhension de la suite de cet article, nous caractérisons les choix prudents issus d'un couple ordonné de relations de préférences $(P_1, P_2)$ grâce à une condition simple.

**Lemme 1** *Soient $P_1, P_2$ deux relations de préférences sur $X$ avec $P_1$ acyclique. Soient $S \in \mathcal{X}$ et $a \in S$. $a \in C_{(P_1,P_2)}(S)$ si et seulement si :*
- *$\forall b \in S, (b, a) \notin P_1$ et,*
- *$\forall b \in S$ tel que $(b, a) \in P_2$, $\exists n \in \mathbb{N} \setminus \{1\}, \exists a_1, ..., a_n \in S$ tels que $\forall k \in \{1, ..., n-1\}$, $(a_k, a_{k+1}) \in Q(P_1, P_2)$, $a_1 = a$ et $a_n = b$.*

**Démo.** Seulement Si : Faisons l'hypothèse que $\exists b \in S, (b, a) \in P_1$. Par définition d'une composition prudente, $\forall P \in \widehat{(P_1, P_2)}(S)$, $(b, a) \in P$. Ainsi, $a \notin C_{(P_1,P_2)}(S)$.

Maintenant, faisons l'hypothèse que $a \in C_{(P_1,P_2)}(S)$, $\forall c \in S, (c, a) \notin P_1$ et $\exists b \in S, (b, a) \in P_2$ et $\forall n \in \mathbb{N} \setminus \{1\}, \nexists a_1, ..., a_n \in S$ tels que $\forall k \in \{1, ...n-1\}$, $(a_k, a_{k+1}) \in Q(P_1, P_2)$, $a_1 = a$ et $a_n = b$. Comme, par hypothèse, $a \in C_{(P_1,P_2)}(S)$, par définition de $C_{(P_1,P_2)}$, $\exists P \in \widehat{(P_1, P_2)}(S)$, $(b, a) \notin P$. Par hypothèse, $(b, a) \notin P_1$, $(b, a) \in P_2$. De plus, $(a, b) \notin Q(P_1, P_2)$. Ainsi, $(b, a) \in Q(P_1, P_2)$. Considérons $P \bigcup (b, a)$. Par définition d'une composition prudente, comme $P \in \widehat{(P_1, P_2)}(S)$ et $(b, a) \in P_2$, $P \bigcup (b, a)$ est cyclique sur $S$. De plus, $P \subset P \bigcup (b, a) \subseteq Q(P_1, P_2) \mid_S$. Ainsi, la cyclicité de $P \bigcup (b, a)$ et l'acyclicité de $P$ sur $S$ impliquent $\exists K \in \mathbb{N} \setminus \{1\}, \exists a_1, ..., a_K \in S$ tels que $\forall k \in \{1, ..., K-1\}$, $(a_k, a_{k+1}) \in Q(P_1, P_2)$, $a_1 = a$ et $a_K = b$. D'où une contradiction avec les hypothèses.

Si : Soient $S \in \mathcal{X}$ et $a \in S$ tels que $\nexists b \in S, (b,a) \in P_1$ et, [$\forall b \in S$ tel que $(b,a) \in P_2$, $\exists n \in \mathbb{N} \setminus \{1\}, \exists a_1, ..., a_n \in S$ tels que $\forall k \in \{1, ..., n-1\}$, $(a_k, a_{k+1}) \in Q(P_1, P_2)$, $a_1 = a$ et $a_n = b$]. Montrons qu'il existe $P \in \widehat{(P_1, P_2)}(S)$ tel que $\forall b \in S, (b,a) \notin P$. Soit $B = \{b \in S, (b,a) \in P_2\}$. Définissons $P^0 = Q(P_1, P_2) \mid_S \setminus \bigcup_{b \in B}(b,a)$. Si $P^0$ is acyclique, alors, par définition d'une composition prudente, $P^0 \in \widehat{(P_1, P_2)}(S)$ et la preuve est faite. Sinon, par hypothèse, $P_1 \mid_S$ est acyclique, d'où il existe $P_1 \subseteq P \subseteq P^0$ tel que $P \in \widehat{(P_1, P_2)}(S)$ et la preuve est faite puisque $\forall b \in B, (b,a) \notin P^0$ et ainsi $\forall b \in S, (b,a) \notin P$.□

Comme premier résultat, notons que pour tout couple $P_1$ et $P_2$ de relations de préférences sur $X$ telles que $P_1$ est acyclique, la paire ordonnée $(P_1, P_2)$ rationalise prudemment une correspondance de choix. Ainsi, ce résultat donne les faibles conditions sous lesquelles une paire ordonnée de deux relations de préférences rationalise prudemment une correspondance de choix. La démonstration est triviale suivant la définition de $\widehat{(P_1, P_2)}(S)$ qui implique que l'ensemble $\widehat{(P_1, P_2)}(S)$ est non-vide et contient toujours des relations de préférences acycliques si et seulement s'il est bien défini, c'est-à-dire si $P_1$ est acyclique.

**Proposition 1** *Soient $P_1$ et $P_2$ deux relations de préférences sur $X$. Il existe une correspondance de choix rationalisée prudemment par $(P_1, P_2)$ si et seulement si $P_1$ est acyclique.*

Le but de la suite de cette Section est de donner les conditions que doit satisfaire une correspondance de choix pour être prudemment rationalisée.

Le premier axiome est un axiome usuel en théorie du choix et est donné dans [14] par exemple. Il impose que si une alternative est choisie dans différents ensembles d'alternatives, elle doit être choisie dans l'union de ces ensembles.

**Axiome 1 ($\gamma$)** *Soit $C \in \mathcal{C}(X)$. La correspondance de choix $C$ satisfait $\gamma$ si et seulement si $\forall n \in \mathbb{N}$ et $\forall S_1, ..., S_n \in \mathcal{X}$,*

*$a \in \bigcap_{i \in \{1,...,n\}} C(S_i)$ implique $a \in C(\bigcup_{i \in \{1,...,n\}} S_i)$.*

Le second axiome est une version faible de la condition de cohérence $\alpha$ dite de Chernoff. Donnons une interprétation. Faisons l'hypothèse que l'ensemble d'alternatives $S$ peut être divisé en deux sous-ensembles distincts, $T$ et $R$ tels que $R$ domine bipartitement $T$, *i.e.* tout élémént de $R$ est choisi quand il n'est considéré dans un ensemble qu'avec un élément de $T$. Maintenant, considérons qu'il existe un élément $a$ de $T$ qui n'est pas choisi quand il n'est considéré qu'avec l'élément $b$ de $R$. Alors, $a$ est dominé par $b$, non seulement parce qu'il n'est pas choisi dans l'ensemble de choix $\{a,b\}$, mais aussi parce que $a$ est un élément de $T$, dominé bipartitement par $R$, dont $b$ est un élément. Dans ce cas de domination double, l'axiome $\alpha$ Faible impose que que $a$ ne soit pas choisi dans $S$.

**Axiome 2 ($\alpha$ Faible)** *Soit $C \in \mathcal{C}(X)$. La correspondance de choix $C$ satisfait $\alpha$ Faible si et seulement si $\forall S \in \mathcal{X}$ et $\forall a \in S$,*

*si $\exists (R,T) \in TD(S,C)$ tel que $a \in T$ et $\exists b \in R, a \notin C(\{a,b\})$, alors $a \notin C(S)$.*

Notons que l'axiome $\alpha$ Faible est plus faible que l'axiome $\alpha$ usuel. En fait, il est aussi plus faible que l'axiome $\alpha 2$ qui lui-même est une version faible de l'axiome $\alpha$, voir [13].[6] En effet, si l'axiome $\alpha 2$ est posé, $a \notin C(S)$ est une conséquence de $\exists b \in S, a \notin C(\{a,b\})$ seulement et la condition de domination bipartite n'est pas nécessaire.

Montrons grâce à un exemple que l'axiome $\alpha$ Faible est strictement plus faible que les axiomes $\alpha$ et $\alpha 2$. Soient $X = \{a,b,c\}$ et $C \in \mathcal{C}(X)$ définie par $C(\{a,b\}) = \{a\}$, $C(\{a,c\}) = \{c\}$, $C(\{b,c\}) = \{b\}$ et $C(X) = \{c\}$. On peut voir que $C$ ne satisfait pas $\alpha 2$ puisque $C(\{b,c\}) = \{b\}$ devrait impliquer $c \notin C(X)$. Comme l'axiome $\alpha$ est plus fort que l'axiome $\alpha 2$, il est évident que l'axiome $\alpha$ n'est pas satisfait par $C$. Cependant, $C$ satisfait $\alpha$ Faible puisque même si $C(\{b,c\}) = \{b\}$, nous avons $TD(X,C) = \emptyset$ et donc l'axiome $\alpha$ Faible est trivialement satisfait. Notons que $C$ est rationalisée prudemment par $(P_1, P_2)$ avec $P_1 = \{(a,b),(c,a)\}$ et $P_2 = \{(b,c)\}$.

[6] Avec nos notations, l'axiome $\alpha$ peut être écrit comme suit : La correspondance de choix $C$ satisfait $\alpha$ si et seulement si $\forall S, T \in \mathcal{X}$ tels que $T \subseteq S$ et $\forall a \in T$, si $a \notin C(T)$, alors $a \notin C(S)$. L'axiome $\alpha 2$ peut être écrit comme suit : La correspondance de choix $C$ satisfait $\alpha 2$ si et seulement si $\forall S \in \mathcal{X}$ et $\forall a \in S$, si $\exists b \in S \setminus \{a\}, a \notin C(\{a,b\})$, alors $a \notin C(S)$.

Le troisième axiome peut être interprété comme suit. Faisons l'hypothèse qu'il n'existe pas de domination bipartite dans $S$ pour $C$. Alors, par définition, il n'y a pas de sous-ensemble d'alternatives "légitime" qui pourrait être désigné dans un sens très faible. Maintenant, faisons l'hypothèse que, pour l'alternative $b$, nous avons la situation suivante : pour toute $a$ dans $S$, si seulement $a$ est choisie quand le choix porte entre $a$ et $b$, alors on peut trouver un ensemble de choix comprenant $a$ et $b$ et tel que $b$ est parmi les alternatives choisies. Ainsi, dans un sens, la domination des éléments de $S$ sur $b$ est soit non-existante soit dépendante de l'ensemble de choix. Dans ce cas, d'après l'axiome de Non Domination Bipartite, l'alternative $b$ devrait être choisie parmi les éléments de $S$ puisqu'elle est au plus dominée dans un sens dépendant de l'ensemble de choix.

**Axiome 3 (Non Domination Bipartite, NDB)** *Soit $C \in \mathcal{C}(X)$. La correspondance de choix $C$ satisfait NDB si et seulement si $\forall S \in \mathcal{X}$ et $\forall b \in S$,*

*si (i) $TD(S,C) = \emptyset$ et (ii) $\forall a \in S \setminus \{b\}, [C(\{a,b\}) = \{a\} \Rightarrow \exists S' \in \mathcal{X}, a \in S'$ et $b \in C(S')]$,*

*alors $b \in C(S)$.*

L'intuition soutenant l'axiome NDB est la suivante. Dans la théorie du choix rationnel traditionnelle, $TD(S,C)$ vide implique, puisqu'elle implique que chacune des alternatives de $S$ est dominée et domine une autre alternative de $S$ dans la relation binaire de base, un choix vide dans $S$. NDB impose que dans une telle situation, une domination qui dépend de l'ensemble de choix devrait être considérée comme plus faible qu'une domination qui ne dépendrait pas de l'ensemble de choix et seulement les alternatives qui sont dominées au sens d'une domination qui dépend de l'ensemble de choix devraient être choisies.

La proposition 2 montre qu'une correspondance de choix peut être rationalisée prudemment si et seulement si elle satisfait $\gamma$, $\alpha$ Faible et NDB.

**Proposition 2** *Soit $C$ in $\mathcal{C}(X)$. La correspondance de choix $C$ satisfait $\gamma$, $\alpha$ Faible et NDB si et seulement si elle est prudemment rationalisée par une paire ordonnée de relations de préférences $(P_1, P_2)$ avec $P_1$ acyclique.*

**Démo.** Seulement Si : Soit $C \in \mathcal{C}(X)$ satisfaisant $\gamma$, $\alpha$ Faible et NDB. Définissons $P_1$ comme suit : $\forall a, b \in X$ tels que $a \neq b$, $(a,b) \in P_1$ si et seulement $C(\{a,b\}) = \{a\}$ et $\forall S \in \mathcal{X}$, $a \in S \Rightarrow b \notin C(S)$. Définissons $P_2$ comme suit : $\forall a, b \in X$ tels que $a \neq b$, $(a,b) \in P_2$ si et seulement si $C(\{a,b\}) = \{a\}$ et $\exists S \in \mathcal{X}, a \in S$ et $b \in C(S)$. Par définition, $P_1$ et $P_2$ sont asymmériques. De plus, $P_1$ is acyclique. En effet, si $P_1$ avait un cycle, nous aurions $\exists n \in \mathbb{N} \setminus \{1\}$, $\exists a_1, ..., a_n \in X$ tels que $\forall k \in \{1, ..., n-1\}$, $(a_k, a_{k+1}) \in P_1$ et $(a_n, a_1) \in P_1$. Alors, par définition, nous aurions $C(\{a_1, ..., a_n\}) = \emptyset$ ce qui serait en constradiction avec le fait que $C$ est une correspondance de choix.

Montrons que $C$ est prudemment rationalisée par $(P_1, P_2)$. Soient $S \in \mathcal{X}$ et $a \in S$. En utilisant le Lemme 1, nous allons montrer que : 1) si $\exists b \in S, (b,a) \in P_1$, alors $a \notin C(S)$, 2) si $\forall b \in S, (b,a) \notin P_1$ et, $\forall b \in S$ tel que $(b,a) \in P_2$, $\exists n \in \mathbb{N} \setminus \{1\}, \exists a_1, ..., a_n \in S$ tels que $\forall k \in \{1, ..., n-1\}, (a_k, a_{k+1}) \in Q(P_1, P_2)$, $a_1 = a$ et $a_n = b$, alors $a \in C(S)$, 3) if $\forall d \in S, (d,a) \notin P_1$ et, $\exists b \in S$ tel que $(b,a) \in P_2$ et $\forall n \in \mathbb{N} \setminus \{1\}$, $\nexists a_1, ..., a_n \in S$ tels que $\forall k \in \{1, ..., n-1\}$, $(a_k, a_{k+1}) \in Q(P_1, P_2)$, $a_1 = a$ et $a_n = b$, alors, $a \notin C(S)$.

1) Faisons l'hypothèse que $\exists b \in S, (b,a) \in P_1$. Par définition de $P_1$, $a \notin C(S)$.

2) Faisons l'hypothèse que $\forall b \in S, (b,a) \notin P_1$ et, $\forall b \in S$ tel que $(b,a) \in P_2$, $\exists n \in \mathbb{N} \setminus \{1\}, \exists a_1, ..., a_n \in S$ tels que $\forall k \in \{1, ..., n-1\}$, $(a_k, a_{k+1}) \in Q(P_1, P_2)$, $a_1 = a$ et $a_n = b$. Définissons $B = \{b \in S, (b,a) \in P_2\}$. Si $B = \emptyset$, alors, par définition de $P_1$ et $P_2$, $\forall b \in S$, $a \in C(\{a,b\})$ et par $\gamma$, $a \in C(S)$. Soit $B \neq \emptyset$. Soit $b \in B$. Par hypothèse, $\exists n \in \mathbb{N} \setminus \{1\}, \exists a_1, ..., a_n \in S$ tels que $\forall k \in \{1, ..., n-1\}$, $(a_k, a_{k+1}) \in Q(P_1, P_2)$, $a_1 = a$ et $a_n = b$. Soit $B_b = \{a_1, ..., a_n\}$. Comme $\forall k \in \{1, ..., n-1\}$, $(a_k, a_{k+1}) \in Q(P_1, P_2)$ et $(b,a) \in Q(P_1, P_2)$, *i.e.* $\{a_k\} = C(\{a_k, a_{k+1}\})$ et $\{b\} = C(\{a,b\})$, $TD(B_b, C) = \emptyset$. De plus, $\forall c \in B_b \setminus \{a\}$, $(c,a) \notin P_1$. Alors, $\exists S' \in \mathcal{X}, b \in S', a \in C(S')$. Alors, par NDB, $a \in C(B_b)$. Ainsi, $a \in \bigcap_{b \in B} C(B_b)$. D'où, par $\gamma$, $a \in C(\bigcup_{b \in B} B_b)$. Si $\bigcup_{b \in B} B_b = S$, $a \in C(S)$. Sinon, par hypothèse, $\forall c \in S \setminus \bigcup_{b \in B} B_b$, $(c,a) \notin P_1 \bigcup P_2$. D'où, par définition, $a \in C(\{a,c\})$. D'où, $a \in C(\bigcup_{b \in B} B_b) \bigcap_{c \in S \setminus \bigcup_{b \in B} B_b} C(\{a,c\})$. D'où, par $\gamma$, $a \in C(S)$.

3) Faisons l'hypothèse que $\forall d \in S, (d,a) \notin P_1$ et, $\exists b \in S$ tel que $(b,a) \in P_2$ et $\forall n \in \mathbb{N} \setminus \{1\}$, $\nexists a_1, ..., a_n \in S$ tels que $\forall k \in \{1, ..., n-1\}$, $(a_k, a_{k+1}) \in Q(P_1, P_2)$, $a_1 = a$ et $a_n = b$. Définissons $ID_a = \{c \in S, \exists K \in \mathbb{N} \setminus \{1\}, \exists a_1, ...a_K \in S, \forall k \in \{1, ..., K-1\}(a_k, a_{k+1}) \in Q(P_1, P_2), a_1 = a, a_K = c\}$. Alors, $ID_a$ est l'ensemble des éléments de $S$ directement ou indirectement dominés par $a$. Par hypothèse, $b \notin ID_a$. Soit $B = S \setminus (ID_a \bigcup \{a\})$. Evidemment $b \in B$. De plus, $\forall c \in ID_a \bigcup \{a\}$, $\forall d \in B, (c,d) \notin Q(P_1, P_2)$ (sinon, évidemment, $d \in ID_a$ ce qui mène à une contradiction). Alors, $\forall c \in ID_a \bigcup \{a\}$, $\forall d \in B, d \in C(\{c,d\})$. Alors, $(B, ID_a \bigcup \{a\}) \in TD(S,C)$. De plus, par hypothèse, $a \notin C(\{a,b\})$ avec $a \in ID_a \bigcup \{a\}$ et $b \in B$. D'où, par $\alpha$ Faible, $a \notin C(S)$.

Si : Soient $P_1$ et $P_2$ deux relations de préférences sur $X$ avec $P_1$ acyclique. Soit $C \in \mathcal{C}(X)$ une correspondance de choix prudemment rationalisée par la paire ordonnée $(P_1, P_2)$.

1) Montrons que $C$ satisfait $\gamma$. Soient $S_1, ..., S_n \in \mathcal{X}$. Soit $a \in \bigcap_{i \in \{1,...,n\}} C(S_i)$. Par le Lemme 1, $\forall i \in \{1, ..., n\}, \forall b \in S_i, (b,a) \notin P_1$. D'où, $\forall b \in \bigcup_{i \in \{1,...,n\}} S_i, (b,a) \notin P_1$. Par le Lemme 1, $\forall i \in \{1, ..., n\}$, $\forall b \in S_i$ tel que $(b,a) \in P_2$, $\exists m_i \in \mathbb{N} \setminus \{1\}, \exists a_1, ..., a_{m_i} \in S_i$ tels que $\forall k \in \{1, ..., m_i - 1\}$, $(a_k, a_{k+1}) \in Q(P_1, P_2)$, $a_1 = a$ et $a_{m_i} = b$. D'où, $\forall b \in \bigcup_{i \in \{1,...,n\}} S_i$ tel que $(b,a) \in P_2$, $\exists m \in \mathbb{N} \setminus \{1\}, \exists a_1, ..., a_m \in \bigcup_{i \in \{1,...,n\}} S_i$ tels que $\forall k \in \{1, ..., m-1\}$, $(a_k, a_{k+1}) \in Q(P_1, P_2)$, $a_1 = a$ et $a_m = b$. D'où, par le Lemme 1, $a \in \bigcup_{i \in \{1,...,n\}} C(S_i)$.

2) Montrons que $C$ satisfait $\alpha$ Faible. Soit $S \in \mathcal{X}$ et $a \in S$. Soit $(R,T) \in TD(S,C)$ tel que $a \in T$ et $[\exists d \in R, a \notin C(\{a,d\})]$. Par définition de $TD(S,C)$, $\forall b \in R, \forall c \in T$, $b \in C(\{b,c\})$, ce qui implique que $(c,b) \notin Q(P_1, P_2)$. Par définition de $Q(P_1, P_2)$ et de la rationalité prudente, $a \notin C(\{a,d\})$ implique $(d,a) \in Q(P_1, P_2)$, ou de manière équivalente, 1) $(d,a) \in P_1$, ou 2) $(a,d), (d,a) \notin P_1$ et $(d,a) \in P_2$. Dans le cas 1), par le Lemme 1, $a \notin C(S)$. Dans le cas 2), comme $\forall b \in R, \forall c \in T$, $(c,b) \notin Q(P_1, P_2)$, nous avons $\forall n \in \mathbb{N} \setminus \{1\}$, $\nexists a_1, ..., a_n \in S$ tels que $\forall k \in \{1, ..., n-1\}$, $(a_k, a_{k+1}) \in Q(P_1, P_2)$, $a_1 = a$ et $a_n = d$. D'où, par le Lemme 1, $a \notin C(S)$.

3) Montrons que $C$ satisfait NDB. Soient $S \in \mathcal{X}$ et $b \in S$. Faisons l'hypothèse que $TD(S,C) = \emptyset$ et $\forall a \in S, C(\{a,b\}) = \{a\} \Rightarrow \exists S' \in \mathcal{X}$ tel que $a, b \in S'$ et $b \in C(S')$. Soit $a \in S$. Par le Lemme 1, si $C(\{a,b\}) = \{a\}$ et $\exists S' \in \mathcal{X}$ tel que $a, b \in S'$ et $b \in C(S')$, alors $(a,b), (b,a) \notin P_1$ et $(a,b) \in P_2$. De plus, par définition de la rationalité prudente, si $b \in C(\{a,b\})$, alors, $(a,b) \notin P_1 \bigcup P_2$. Alors, $\forall a \in S, (a,b) \notin P_1$. Maintenons, considérons $a \in S$ tel que $(a,b) \in P_2$. Alors, par définition de la rationalité prudente, $C(\{a,b\}) = \{a\}$. Définissons $ID_b = \{c \in S, \exists K \in \mathbb{N} \setminus \{1\}, \exists d_1, ...d_K \in S$ tels que $\forall k \in \{1, ..., K-1\}, (d_k, d_{k+1}) \in Q(P_1, P_2), d_1 = b$ et $d_K = c\} \cup \{b\}$ et $\overline{ID_b} = S \setminus ID_b$. Faisons l'hypothèse que $\overline{ID_b} \neq \emptyset$. Par définition, $ID_b \bigcap \overline{ID_b} = \emptyset$ et $ID_b \bigcup \overline{ID_b} = S$. Alors, pour avoir $TD(S,C) = \emptyset$, nous avons nécessairement $\exists c \in ID_b, \exists d \in \overline{ID_b}$ tels que $\{c\} = C(\{c,d\})$. Cela implique que $(c,d) \in Q(P_1, P_2)$ et alors, $d \in ID_b$ ce qui contredit le fait que $d \in \overline{ID_b}$. Alors, $\overline{ID_b} = \emptyset$ et en particulier, $a \in ID_b$. Alors, par définition, $\exists K \in \mathbb{N} \setminus \{1\}, \exists d_1, ...d_K \in S$ tels que $\forall k \in \{1, ..., K-1\}, (d_k, d_{k+1}) \in Q(P_1, P_2), d_1 = b$ et $d_K = a$. En appliquant le même raisonnement pour tout $a \in S$ tel que $(a,b) \in P_2$, nous avons $b \in C(S)$ par le Lemme 1.□

Evidemment, quand l'ensemble des alternatives a trois éléments ou moins, l'axiome NDB est trivialement satisfait. Pour au moins quatre alternatives, la proposition suivante montre que les axiomes donnés dans la Proposition 2 sont indépendants.

**Proposition 3** *Les axiomes $\gamma$, $\alpha$ Faible et NDB sont indépendants.*

**Démo.** $\gamma$ et $\alpha$ Faible $\not\Rightarrow$ NDB : Soit $X = \{a,b,c,d\}$ et soit la correpondance de choix $C \in \mathcal{C}(X)$ définie par : $C(\{a,b\}) = \{a\}$, $C(\{a,c\}) = \{c\}$, $C(\{a,d\}) = \{d\}$, $C(\{b,c\}) = \{b\}$, $C(\{b,d\}) = \{b\}$, $C(\{c,d\}) = \{c\}$, $C(\{a,b,c\}) = \{c\}$, $C(\{a,b,d\}) = \{a\}$, $C(\{a,c,d\}) = \{c\}$, $C(\{b,c,d\}) = \{b\}$, $C(X) = X$. Il est immédiat de vérifier que $C$ satisfait $\gamma$ et $\alpha$ Faible. Montrons que $C$ ne satisfait pas NDB. Nous avons $TD(\{a,b,c\}, C) = \emptyset$ et $\forall e \in \{a,b,c\}$ tel que $C(\{a,e\}) = \{e\}$, $\exists S \in \mathcal{X}$ tel que $a, e \in S$ et $a \in C(S)$ (Pour voir le dernier point, considérer $S = X$). D'où, par NDB, nous devrions avoir $a \in C(\{a,b,c\})$ ce qui n'est pas le cas.

$\alpha$ Faible et NDB $\not\Rightarrow \gamma$ : Soit $X = \{a, b, c\}$ et soit la correpondance de choix $C \in \mathcal{C}(X)$ définie par : $C(\{a, b\}) = \{a, b\}$, $C(\{a, c\}) = \{a, c\}$, $C(\{b, c\}) = \{b, c\}$, $C(X) = \{b\}$. Il est immédiat de vérifier que $C$ satisfait $\alpha$ Faible et NDB. Cependant, $C$ ne satisfait pas $\gamma$ puisque $C(\{a, b\}) = \{a, b\}$ et $C(\{a, c\}) = \{a, c\}$ devraient impliquer $a \in C(\{X\})$ ce qui n'est pas le cas.

$\gamma$ et NDB $\not\Rightarrow \alpha$ Faible : Soit $X = \{a, b, c\}$ et soit la correpondance de choix $C \in \mathcal{C}(X)$ définie par : $C(\{a, b\}) = \{a\}$, $C(\{a, c\}) = \{a\}$, $C(\{b, c\}) = \{b\}$, $C(X) = \{a, b\}$. Il est immédiat de vérifier que $C$ satisfait $\gamma$ et NDB. Cependant, $C$ ne satisfait pas $\alpha$ Faible. En effet, nous avons $(\{a\}, \{b, c\}) \in TD(X, C)$ et $C(\{a, b\}) = \{a\}$. D'où, par $\alpha$ Faible, nous devrions avoir $b \notin C(X)$, ce qui n'est pas le cas. □

## 4 Relation avec les autres règles de composition

Nous continuons cette étude des choix prudents en faisant le lien entre ceux-ci et deux des principales procédures pour composer deux critères existant dans la littérature. La première procédure a été étudiée par [11] (rationalité séquentielle), la seconde l'a été par [15] (rationalité par composition lexicographique des relations de base). Formellement, soit $C \in \mathcal{C}(X)$ une correspondance de choix sur $X$. Nous disons que $C$ est séquentiellement rationnelle si et seulement s'il existe deux relations de préférences $P_1$ et $P_2$ sur $X$ telles que $\forall S \in X, C(S) = C_{P_2}(C_{P_1}(S))$. Nous disons que $C$ est rationnelle par composition lexicographique des relations de base (CLB-rationnelle) si et seulement s'il existe deux relations de préférences sur $X$, $P_1$ et $P_2$, telles que $\forall S \in X, C(S) = C_{Q(P_1,P_2)}(S)$. Enfin, pour la procédure que nous avons introduite dans cet article, nous disons que $C$ est prudemment rationnelle si et seulement s'il existe deux relations de préférences $P_1$ et $P_2$ telles que $C$ est prudemment rationalisée par la paire ordonnée $(P_1, P_2)$.

**Proposition 4** *Soit $C \in \mathcal{C}(X)$.*

1. *Si $C$ est CLB-rationnelle, alors elle est séquentiellement rationnelle et prudemment rationnelle.*
2. *$C$ peut être séquentiellement rationnelle et ne pas être prudemment rationnelle.*
3. *$C$ peut être prudemment rationnelle et ne pas être séquentiellement rationnelle.*
4. *$C$ peut être séquentiellement rationnelle ou prudemment rationnelle et ne pas être CLB-rationnelle.*

**Démo.** 1 : Soit $C \in \mathcal{C}(X)$ CLB-rationalisée par $P_1$ et $P_2$. Si nous posons $P'_1 = Q(P_1, P_2)$ et $P'_2 = \emptyset$, $C$ est séquentiellement rationalisée par $P'_1$ et $P'_2$ et $C$ est prudemment rationalisée par $(P'_1, P'_2)$.

2 : Soit $X = \{a, b, c, d\}$ et soit la correpondance de choix $C \in \mathcal{C}(X)$ définie par : $C(\{a, b\}) = \{a\}$, $C(\{a, c\}) = \{a, c\}$, $C(\{a, d\}) = \{a, d\}$, $C(\{b, c\}) = \{b\}$, $C(\{b, d\}) = \{d\}$, $C(\{c, d\}) = \{c\}$, $C(\{a, b, c\}) = \{a, c\}$, $C(\{a, b, d\}) = \{a, d\}$, $C(\{a, c, d\}) = \{a, c\}$, $C(\{b, c, d\}) = \{c\}$ et $C(X) = \{a, c\}$. Cette correspondance de choix est séquentiellement rationalisée par $P_1 = \{(a, b), (d, b), (c, d)\}$ et $P_2 = \{(b, c)\}$. Cependant, il est immédiat de vérifier que $C$ n'est pas prudemment rationnelle. En effet, elle ne satisfait pas l'axiome $\alpha$ Faible. Pour voir cela, notons que $(\{a, b\}, \{c\}) \in TD(\{a, b, c\}, C)$, $\{b\} = C(\{b, c\})$ et, $c \in C(\{a, b, c\})$, ce qui est interdit par l'axiome $\alpha$ Faible.

3 : Soit $X = \{a, b, c\}$ et soit la correpondance de choix $C \in \mathcal{C}(X)$ définie par : $C(\{a, b\}) = \{a\}$, $C(\{b, c\}) = \{b\}$, $C(\{a, c\}) = \{c\}$ et $C(X) = \{a, c\}$. Cette correspondance de choix est prudemment rationalisée par $(P_1, P_2)$ avec $P_1 = \{(a, b)\}$ et $P_2 = \{(b, c), (c, a)\}$. Cependant, il est immédiat de vérifier que $C$ n'est pas séquentiellement rationnelle.

4 : Par 1, 2 et 3 de la Proposition 4.□

## 5 Relation avec la théorie classique

La théorie classique de la rationalité des choix caractérise les correspondances de choix qui peuvent être obtenues par maximisation d'une seule relation de préférences. Il est bien connu que ces correspondances de choix sont celles qui satisfont les axiomes $\alpha$ et $\gamma$.

**Proposition 5 ([4])** *Soit $C \in \mathcal{C}(X)$. La correspondance de choix $C$ satisfait $\alpha$ et $\gamma$ si et seulement s'il existe une relation de préférences acyclique $P$ telle que $P$ rationalise $C$.*

Pour faire le lien entre notre caractérisation des choix prudents et la Proposition 5, nous allons poser l'axiome suivant qui impose à une correspondance de choix $C$ que $TD(S,C)$ soit non-vide pour tout $S \in \mathcal{X}$.

**Axiome 4 (Toujours Domination Bipartite, TDB)** *Soit $C \in \mathcal{C}(X)$. La correspondance de choix $C$ satisfait TDB si et seulement si $\forall S \in \mathcal{X}$, $TD(S,C) \neq \emptyset$.*

Bien sûr, si une correspondance de choix satisfait TDB, alors elle satisfait également trivialement NDB. D'où la proposition suivante qui montre que la rationalité au sens classique correspond à la rationalité prudente à laquelle on a imposé TDB

**Proposition 6** *Soit $C \in \mathcal{C}(X)$. La correspondance de choix $C$ satisfait TDB, $\alpha$ Faible et $\gamma$ si et seulement s'il existe une relation de préférences acyclique $P$ telle que $P$ rationalise $C$.*

**Démo.** Seulement Si : Soit $C \in \mathcal{C}(X)$ satisfaisant $\gamma$, $\alpha$ Faible et TDB. Définissons $P$ comme suit : $\forall a,b \in X$ tels que $a \neq b$, $(a,b) \in P$ si et seulement si $C(\{a,b\}) = \{a\}$. Par définition, $P$ est asymétrique. De plus, $P$ est acyclique. En effet, si $P$ avait un cycle, nous aurions $\exists n \in \mathbb{N} \setminus \{1\}$, $\exists a_1, ..., a_n \in X$ tels que $\forall k \in \{1, ..., n-1\}$, $(a_k, a_{k+1}) \in P$ et $(a_n, a_1) \in P$. Alors, par définition de $P$, nous aurions $TD(\{a_1, ..., a_n\}, C) = \emptyset$ ce qui contredirait TDB.

Montrons que $C$ est rationalisée par $P$. Soit $S \in \mathcal{X}$ et $a \in S$.

1) Faisons l'hypothèse que $\exists b \in S$ tel que $(b,a) \in P$. Définissons $ID_a = \{c \in S, \exists K \in \mathbb{N} \setminus \{1\}, \exists a_1, ...a_K \in S, \forall k \in \{1, ..., K-1\}(a_k, a_{k+1}) \in P, a_1 = a, a_K = c\}$. Ainsi, $ID_a$ est l'ensemble des éléments de $S$ directement ou indirectement dominés par $a$. Comme $P$ est acylique, $b \notin ID_a$. De plus, $\forall c \in ID_a \bigcup \{a\}, \forall d \in S \setminus (ID_a \bigcup \{a\})$, par définition de $P$, $(c,d) \notin P$ (ou sinon, nous aurions $d \in ID_a \bigcup \{a\}$ ce qui contredirait $d \in S \setminus (ID_a \bigcup \{a\})$). D'où $\forall c \in ID_a \bigcup \{a\}, \forall d \in S \setminus (ID_a \bigcup \{a\})$, $d \in C(\{c,d\})$. D'où, $\{b\} = C(\{a,b\})$ et $(S \setminus (ID_a \bigcup \{a\}), ID_a \bigcup \{a\}) \in TD(S,C)$. Alors, par $\alpha$ Faible, $a \notin C(S)$.

2) Faisons l'hypothèse que $\forall b \in S$, $(b,a) \notin P$. Alors, par définition de $P$, $\forall b \in S, a \in C(\{a,b\})$. D'où, par $\gamma$, $a \in C(S)$.

Si : Soit $P$ une relation de préférences acyclique sur $X$. Soit $C \in \mathcal{C}(X)$ une correspondance de choix rationalisée par $P$ (elle existe par Proposition 5).

1) $C$ satisfait $\gamma$ par Proposition 5.

2) $C$ satisfait $\alpha$ Faible par Proposition 5 et le fait que $\alpha$ Faible est impliqué par $\alpha$.

3) Montrons que $C$ satisfait TDB. Soit $S \in \mathcal{X}$. Comme $P$ est acyclique et $S$ a un nombre fini d'éléments, il existe $a \in S$ tel que $\forall b \in S, (b,a) \notin P$. Par définition de la domination bipartite, $(\{a\}, S \setminus \{a\}) \in TD(S,C)$. □

De plus, quand l'ensemble des alternatives a trois éléments au plus, les axiomes TDB, $\alpha$ Faible et $\gamma$ sont independants.

**Proposition 7** *Soit $X$ un ensemble d'alternatives avec au moins trois éléments. TDB, $\alpha$ Faible et $\gamma$ sont independants.*

**Démo.** $\gamma$ et $\alpha$ Faible $\not\Rightarrow$ TDB : Soit $X = \{a,b,c\}$ et soit la correpondance de choix $C \in \mathcal{C}(X)$ définie par : $C(\{a,b\}) = \{a\}$, $C(\{a,c\}) = \{c\}$, $C(\{b,c\}) = \{b\}$, $C(X) = \{a\}$. Il est immédiat de vérifier que $C$ satisfait $\gamma$ et $\alpha$ Faible. Cependant, $TD(\{a,b,c\}, C) = \emptyset$ et donc $C$ ne satisfait pas TDB.

$\alpha$ Faible et TDB $\not\Rightarrow \gamma$ : Soit $X = \{a,b,c\}$ et soit la correpondance de choix $C \in \mathcal{C}(X)$ définie par : $C(\{a,b\}) = \{a,b\}$, $C(\{a,c\}) = \{a,c\}$, $C(\{b,c\}) = \{b,c\}$, $C(X) = \{b\}$. Il est immédiat de vérifier que $C$ satisfait $\alpha$ Faible et TDB. Cependant, $C$ ne satisfait pas $\gamma$ puisque $C(\{a,b\}) = \{a,b\}$ et $C(\{a,c\}) = \{a,c\}$ devraient impliquer $a \in C(\{X\})$.

$\gamma$ et TDB $\not\Rightarrow \alpha$ Faible : Soit $X = \{a,b,c\}$ et soit la correpondance de choix $C \in \mathcal{C}(X)$ définie par : $C(\{a,b\}) = \{a\}$, $C(\{a,c\}) = \{a\}$, $C(\{b,c\}) = \{b\}$, $C(X) = \{a,b\}$. Il est immédiat de vérifier que $C$ satisfait $\gamma$ et TDB. Cependant, $C$ ne satisfait pas $\alpha$ Faible. En effet, nous avons $(\{a\}, \{b,c\}) \in TD(X,C)$ et $C(\{a,b\}) = \{a\}$. D'où, par $\alpha$ Faible, nous devrions avoir $b \notin C(X)$. □

## Références

[1] Aizerman M. A. (1985) "New Problems in the General Choice Theory" *Social Choice and Welfare* 2 : 235-282.

[2] Aizerman, M. A. et F. T. Aleskerov (1995), *Theory of Choice*. Amsterdam : North Holland.

[3] Arrow K. et H. Raynaud (1986) Social Choice and Multicriterion Decision Making. MIT Press.

[4] Blair D. H., G. Bordes, J. S. Kelly et K. Suzumura (1976) "Impossibility Theorems without Collective Rationality" *Journal of Economic Theory* 13 : 361-379.

[5] Houy N. (2007) "Rationality and order-dependent sequential rationality" *Theory and Decision* 62(2) : 119-134.

[6] Houy N. et K. Tadenuma (2007) "Lexicographic compositions of two criteria for decision making" *Journal of Economic Theory, forthcoming*.

[7] Lamboray C. (2007) "A Characterization of the Prudent Order Preference Function" *Mathematical Social Sciences* 57 : 389-405.

[8] Lamboray C. (2009) "A Characterization of the Ranked Pairs Rule" *Social Choice and Welfare,* 32 : 129-155.

[9] Lansdowne Z. (1996) "Ordinal Ranking Methods for Multicriterion Decision Making" *Naval Research Logistics* 43 : 613-627.

[10] Lansdowne Z. (1997) "Outranking Methods for Multicriterion Decision-Making : Arrow's and Raynaud's Conjecture" *Social Choice and Welfare* 14 : 125-128.

[11] Manzini P. et M. Mariotti (2007) "Sequentially Rationalizable Choice" *American Economic Review* 97(5) : 1824-1839.

[12] Manzini P. et M. Mariotti (2007) "Shorlisting" Working paper EconWPA ewp-pe/0503006.

[13] Sen A. (1977) "Social Choice Theory : a Re-Examination" *Econometrica* 45(1) : 53-89.

[14] Sen A. (1993) "Internal Consistency of Choice" *Econometrica* 61(3) : 495-521.

[15] Tadenuma K. (2002) "Efficiency First or Equity First ? Two Principles and Rationality of Social Choice" *Journal of Economic Theory* 104(2) : 462-472.

[16] Tideman T.N. (1987) "Independent of Clones as a Criterion for Voting Rules" *Social Choice and Welfare* 4(3) : 185-206.

[17] Zavist T.M. et T.N. Tideman (1989) "Complete Independence of Clones in the Ranked Pairs Rule" *Social Choice and Welfare* 6(2) : 167-173.

# Choix résolu et utilité espérée dépendant du rang dans les diagrammes d'influence

G. Jeantet* O. Spanjaard*
gildas.jeantet@lip6.fr olivier.spanjaard@lip6.fr

*Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie
104, avenue du Président Kennedy
75016 Paris – FRANCE

**Résumé :**
Nous nous intéressons ici à l'optimisation de l'utilité espérée dépendant du rang (RDU) dans un diagramme d'influence. Les stratégies potentielles considérées habituellement dans un diagramme d'influence forment un sous-ensemble $X$ de l'ensemble $Y$ des stratégies potentielles considérées dans l'arbre de décision-hasard obtenu en développant le diagramme. A la différence du modèle EU, il n'existe pas nécessairement une stratégie optimale selon RDU dans $X$ (autrement dit, l'ensemble des stratégies optimales selon RDU peut être inclus dans $Y \setminus X$). Dans cet article, nous proposons d'une part une méthode en deux phases pour déterminer la stratégie optimale dans $Y$ (développement du diagramme d'influence en arbre décision hasard, puis détermination de la stratégie optimale selon RDU dans cet arbre), et nous montrons d'autre part comment enrichir l'espace $X$ des stratégies lorsque l'on souhaite opérer directement sur le diagramme d'influence. Nous présentons en particulier un algorithme d'énumération implicite afin de déterminer une stratégie optimale dans cet espace. Les expérimentations numériques menées s'avèrent encourageantes.

**Mots-clés :** Théorie de la décision algorithmique, planification dans l'incertain, utilité dépendant du rang, diagrammes d'influence

**Abstract:**
We investigate here how to optimize rank expected utility in influence diagrams. We propose two methods : a two-phase method that computes an optimal strategy whose description is exponential in the size of the diagram, and an implicit enumeration algorithm that computes an optimal strategy in a subset of concisely expressed strategies. Numerical results are provided that show the interest of our approach.

**Keywords:** Algorithmic decision theory, planning under uncertainty, rank dependent utility, influence diagrams

## 1 Introduction

On appelle *décision dans l'incertain* les situations de choix où les conséquences des actions d'un agent ne sont pas connues avec certitude. Lorsque cette incertitude est probabilisée, autrement dit lorsque le résultat obtenu ne dépend que de la réalisation d'événements dont on connaît les probabilités, on parle alors de *décision dans le risque* [5]. Dans ce cadre, on est amené à modéliser des préférences sur des *loteries*, c'est-à-dire des distributions de probabilité sur des conséquences potentielles. De nombreux modèles de décision dans le risque existent dans la littérature. Le plus connu et le plus utilisé est le modèle de l'*espérance d'utilité* (EU), de par sa simplicité et son attrait sur le plan normatif [21]. Dans ce modèle, on assigne une valeur numérique à chaque conséquence via une fonction d'utilité $u$, puis on privilégie les décisions conduisant à maximiser l'espérance d'utilité. Néanmoins, malgré son aspect intuitif séduisant, le modèle EU ne permet pas de rendre compte de tous les comportements décisionnels rationnels. A titre illustratif, nous présentons ci-dessous une variante du paradoxe d'Allais [1] dûe à Kahneman et Tversky [12].

**Exemple 1 (paradoxe d'Allais)** *On propose à un agent, dans un premier problème, de choisir entre deux loteries $L_1$ et $L'_1$, et entre les loteries $L_2$ et $L'_2$ dans un second problème (voir table 1). Dans le premier problème, l'agent préfère la loterie $L_1$ à $L'_1$ (il préfère être sûr de gagner $3000$ euros plutôt que risquer de ne rien gagner pour un surplus de gain qui lui semble insuffisant), alors que dans le second problème l'agent préfère la loterie $L'_2$ à la loterie $L_2$ (car la probabilité de gagner $4000$ euros dans la loterie $L'_2$ est pratiquement la même que la probabilité de gagner $3000$ euros dans $L_2$). Le modèle EU n'est pas capable de rendre compte de ce comportement parfaitement rationnel qui décrit les choix qu'auraient fait de nombreux décideurs. En effet, la préférence pour $L_1$ sur $L'_1$ implique $u(3000) > 0.1u(0) + 0.9u(4000)$. Ce qui est équivalent à $0.1u(3000) > 0.01u(0) + 0.09u(4000)$ qui est lui-même équivalent, par*

| Loterie | 0 € | 3000 € | 4000 € |
|---|---|---|---|
| $L_1$ | 0.00 | 1.00 | 0.00 |
| $L'_1$ | 0.10 | 0.00 | 0.90 |
| $L_2$ | 0.90 | 0.10 | 0.00 |
| $L'_2$ | 0.91 | 0.00 | 0.09 |

TAB. 1 – Loteries dans le paradoxe d'Allais.

*ajout de* $0.9u(0)$ *de chaque côté de l'inéquation, à* $0.9u(0) + 0.1u(3000) > 0.91u(0) + 0.09u(4000)$. *Cette dernière inégalité traduit la préférence de* $L_2$ *sur* $L'_2$. *Ainsi, quelle que soit la fonction d'utilité utilisée, la préférence pour* $L_1$ *sur* $L'_1$ *est équivalente à la préférence pour* $L_2$ *sur* $L'_2$ *dans le cadre du modèle EU.*

Allais qualifie ce comportement, loin d'être paradoxal, comme un comportement raisonnable de "préférence pour la sécurité dans le voisinage de la certitude" [2]. En d'autres termes, "un tiens vaut mieux que deux tu l'auras". Ce phénomène est aujourd'hui bien connu sous le nom d'*effet de certitude*. Dans l'exemple, la probabilité de gagner 3000 € dans $L_1$ (resp. 4000 € dans $L'_1$) est simplement multipliée par 0.1 dans $L_2$ (resp. $L'_2$). Le renversement de préférence peut être expliqué comme suit : quand la probabilité de gagner devient basse, la sensibilité aux probabilités décroît. Pour tenir compte de l'effet de certitude dans l'élaboration d'un critère de choix, la traitement des probabilités ne doit donc pas être linéaire. Ceci a conduit à l'émergence de modèles non-linéaires plus sophistiqués qui généralisent EU. A ce titre, nous nous intéressons ici plus particulièrement au modèle de l'*utilité espérée dépendant du rang* (RDU pour *rank dependent utility*) [18] qui généralise le modèle EU. Dans ce modèle, une fonction non-linéaire $\varphi$ de déformation des probabilités est introduite, ce qui permet de rendre compte de comportement décisionnel tel que celui observé dans le paradoxe d'Allais. De plus, la fonction $\varphi$ de déformation des probabilités permet aussi de modéliser plus finement l'attitude du décideur vis-à-vis du risque. En effet, contrairement au modèle EU, le modèle RDU permet de distinguer entre l'aversion au risque (i.e., une conséquence certaine est toujours préférées à une loterie dont l'espérance d'utilité est égale à l'utilité de la conséquence certaine) et l'aversion à un accroissement de risque (i.e., si deux loteries ont la même espérance d'utilité, l'agent préfére toujours celle dont l'étalement des conséquences possibles est le plus faible). Ce soucis de modéliser finement l'attitude de l'agent vis-à-vis du risque a été mis en avant dans plusieurs travaux récents en IA [16, 13].

La difficulté de mise en œuvre opérationnelle du modèle RDU a freiné son utilisation jusqu'à aujourd'hui. En effet, dans la pratique, il est fréquent de rencontrer des problèmes de décision *séquentielle* dans le risque, c'est-à-dire des problèmes où l'on ne suit pas une unique décision, mais une *stratégie* (i.e. une séquence de décisions conditionnellement à des événements incertains) conduisant à une conséquence finale incertaine. C'est le cas en particulier dans les problèmes de planification dans l'incertain étudiés en IA, tels que la navigation d'un robot mobile, le diagnostic médical, l'organisation des secours ou encore la planification des actions d'un joueur artificiel. Dans de tels problèmes, on cherche à établir un plan optimal, ce qui nous amène à comparer des stratégies. Pour ce faire, on associe à chaque stratégie la loterie induite sur les conséquences potentielles, puis on compare ces loteries à l'aide d'un critère de décision (EU usuellement). Ces problèmes peuvent se représenter à l'aide de plusieurs outils formels, parmi lesquels on peut mentionner entre autres les arbres décision-hasard [19], les diagrammes d'influence [7] et les processus décisionnels markoviens [4]. Il est important de noter que dans toutes ces représentations l'ensemble des stratégies potentielles est combinatoire. La recherche d'une stratégie optimale pour une représentation et un critère de décision donnés est alors un problème algorithmique en soi. La non-linéarité du critère RDU induit de nouvelles difficultés algorithmiques, qui ne se posaient pas avec le critère EU. Ceci fait précisément l'objet de notre travail.

Un premier travail a déjà été mené sur l'optimisation de RDU dans les arbres décision hasard [10] (le problème a été prouvé NP-difficile, et un algorithme d'énumération implicite a été proposé). Toutefois, cette approche se heurte à l'inconvénient principal des arbres décision hasard : leur taille peut rapidement devenir prohibitive lorsque le nombre d'étapes de décision croît. Pour cette raison, nous nous intéressons ici à l'optimisation de RDU dans le cadre des *diagrammes d'influence* [7]. En effet, les diagrammes d'influence sont des modèles graphiques qui permettent une représentation particulièrement compacte des problèmes de décision séquentielle dans l'incertain, en exploitant les dépendances probabilistes du problème pour factoriser certaines données. La détermination d'une stratégie EU-optimale dans un diagramme d'influence est déjà un problème NP-difficile. Néanmoins des algorithmes performants utilisant des techniques importées des réseaux bayésiens existent pour le résoudre [11]. Ces algorithmes ne s'étendent pas pour l'optimisation du critère RDU. De plus, une difficulté supplémentaire survient lorsque l'on utilise ce dernier critère : contrairement à une stratégie EU-optimale, une stratégie RDU-optimale n'est pas nécessairement *conséquentialiste*. En

toute rigueur, un *comportement* est dit conséquentialiste lorsque les décisions de l'agent ne dépendent ni du passé ni de scénarios contrefactuels (scénarios qui auraient pu se produire, mais cela n'a pas été le cas) [8]. Par abus de langage, nous qualifions ici de conséquentialiste une *stratégie* où chaque décision prise ne dépend que des variables influençant les paramètres futurs (probabilités et conséquences). Dans les diagrammes d'influence, cette propriété est cruciale car la description de telles stratégies reste alors linéaire dans la taille du diagramme, ce qui n'est plus le cas pour les stratégies non-conséquentialistes. La description d'une stratégie RDU-optimale peut ainsi nécessiter un espace mémoire exponentiel dans la taille de l'instance, ce qui rend la tâche d'optimisation singulièrement ardue.

Le papier est organisé comme suit : dans la section 2, nous rappellons les bases du modèle RDU ; dans la section 3 nous rappelons le formalisme des diagrammes d'influence et nous mettons en avant les difficultés soulevées par l'utilisation de RDU ; dans la section 5 nous présentons deux approches visant à optimiser RDU dans un diagramme d'influence ; enfin dans la section 6 nous présentons les résultats numériques que nous avons obtenus.

## 2 Utilité espérée dépendant du rang

Dans le modèle RDU, le poids attribué à une conséquence ne dépend pas uniquement de la probabilité d'obtenir cette conséquence, mais également de l'attractivité du rang de cette conséquence par rapport aux autres conséquences possibles. Ce modèle repose sur deux paramètres : une fonction d'utilité qui est déjà présente dans le modèle EU, et une fonction $\varphi$ de déformation des probabilités. Il s'agit d'une fonction strictement croissante sur $[0,1]$ telle que $\varphi(0) = 0$ et $\varphi(1) = 1$. Cette déformation des probabilités porte, non sur des probabilités simples, mais sur des cumuls de probabilités. Pour rappel, étant donnée une loterie $L = (p_1, u_1; \ldots; p_k, u_k)$ (qui rapporte une utilité $u_i$ avec une probabilité $p_i$), on appelle *fonction décumulative* de $L$ la fonction $G_L : S \mapsto [0,1]$ qui associe à chaque utilité $u_i$ la probabilité d'avoir au moins cette utilité. Plus formellement, $G_L(x) = \sum_{i:u_i \geq x} p_i$. La valeur selon RDU d'une loterie $L$ est alors définie de la manière suivante :

$RDU(L) = u_{(1)} + \sum_{i=2}^{k} [u_{(i)} - u_{(i-1)}]\varphi(G_L(u_{(i)}))$

où (.) correspond à une permutation de $\{1, \ldots, k\}$ telle que $u_{(1)} \leq \ldots \leq u_{(k)}$. Ce critère peut être interprété comme suit : on est sûr d'obtenir au moins une utilité de $u_{(1)}$, puis on est susceptible d'obtenir un supplément d'utilité de $u_{(2)} - u_{(1)}$ avec une masse de probabilité $\varphi(G_L(u_{(2)}))$, puis un supplément d'utilité de $u_{(3)} - u_{(2)}$ avec une masse de probabilité $\varphi(G_L(u_{(3)}))$, et ainsi de suite... Ce modèle est en fait une généralisation du modèle EU. En effet, si la fonction de déformation des probabilités est linéaire, alors on se retrouve dans le cadre du modèle EU. Le modèle RDU permet de résoudre le paradoxe d'Allais (voir exemple 2) et bien d'autres situations où le modèle EU était impuissant.

**Exemple 2** *Revenons sur le paradoxe d'Allais. On définit pour l'agent une utilité linéaire $u(x) = x$ et on utilise la fonction $\varphi$ suivante : $\varphi(0.09) = \varphi(0.1) = 0.2, \varphi(0.9) = 0.7$. Les préférences retournées par le modèle RDU sont alors compatibles avec le paradoxe d'Allais :*

$RDU(L_1) = u(3000)$
$= 3000$
$RDU(L'_1) = u(0) + \varphi(0.9)(u(4000) - u(0))$
$= 2800$

*Ainsi $L_1$ est préférée à $L'_1$. De même on a :*

$RDU(L_2) = u(0) + \varphi(0.1)(u(3000) - u(0))$
$= 600$
$RDU(L'_2) = u(0) + \varphi(0.09)(u(4000) - u(0))$
$= 800$

*Et donc $L'_2$ est préférée à $L_2$.*

L'intérêt de déformer des cumuls de probabilités, et non directement les probabilités elles-mêmes (comme c'est par exemple le cas dans le modèle de Handa [6]), est d'obtenir un critère de choix compatible avec la dominance stochastique. On dit qu'une loterie $L = (p_1, u_1; \ldots; p_k, u_k)$ domine stochastiquement une loterie $L' = (p'_1, u'_1; \ldots; p'_k, u'_k)$ si $\forall x \in \mathbb{R}, G_L(x) \geq G_{L'}(x)$, autrement dit, pour tout $x \in \mathbb{R}$, la probabilité d'obtenir une utilité d'au moins $x$ avec la loterie $L$ est au moins aussi grande qu'avec la loterie $L'$. La compatibilité avec la dominance stochastique signifie que $RDU(L) \geq RDU(L')$ dès lors que $L$ domine stochastiquement $L'$ [17]. Cette propriété est bien entendu souhaitable pour décrire un comportement rationnel, et elle est bien vérifiée par le modèle RDU (contrairement au modèle de Handa).

## 3 Diagramme d'influence

Le formalisme des *arbres décision hasard* fournit une représentation simple et explicite d'un problème de décisions séquentielles dans le risque. Il s'agit d'une arborescence comportant trois types de nœuds : les *nœuds de décision* (représentés par des carrés), les *nœuds de hasard* (représentés par des cercles), et les *nœuds terminaux* (les feuilles de l'arborescence). Les branches issues d'un nœud de décision correspondent à différentes décisions possibles, tandis que celles issues d'un nœud de hasard correspondent aux différents événements possibles, dont les probabilités sont indiquées sur les branches. Enfin, les valeurs figurant au niveau des feuilles de l'arborescence correspondent aux utilités des différentes conséquences. Un exemple d'arbre décision hasard est fourni sur la figure 2.

Un *diagramme d'influence* [7] est une représentation graphique alternative d'un problème de décisions séquentielles. A la différence d'un arbre décision hasard, l'accent est mis sur la décomposabilité de la structure de probabilité sous-jacente au problème. En effet, en tirant parti des indépendances entre les variables aléatoires et les variables d'utilité impliquées, on obtient une représentation beaucoup plus compacte que celle obtenue en explicitant tous les cas de figure possibles, comme on le ferait dans un arbre décision hasard. Un diagramme d'influence incluant des variables aléatoires $A_1, \ldots, A_n$ et des variables de décision $D_1, \ldots, D_k$ est un graphe orienté sans circuit $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ tel que :

- l'ensemble $\mathcal{N}$ se partitionne en trois sous-ensembles : un ensemble $\mathcal{N}_D = D_1, \ldots, D_n$ de variables de décision (représentées par des carrés), un ensemble $\mathcal{N}_A = A_1, \ldots, A_p$ de variables aléatoires (représentées par des cercles), et un ensemble $\mathcal{N}_U = U_1, \ldots, U_m$ de nœuds d'utilité (représentés par des losanges) ;
- l'ensemble $\mathcal{E}$ des arcs se partionne en deux sous-ensembles : un ensemble d'arcs fonctionnels allant d'une variable de décision ou d'une variable aléatoire vers une variable aléatoire ou un nœud d'utilité (arcs représentant des dépendances), un ensemble d'arcs informationnels allant d'une variable de décision ou d'une variable aléatoire vers une variable de décision (arcs représentant les variables observées avant de prendre une décision) ;
- les nœuds représentant les variables aléatoires sont munis d'une table de probabilité conditionnelle, qui indique la probabilité de chaque événement conditionnellement aux nœuds parents ;
- les nœuds d'utilité sont munis d'une table indiquant l'utilité conditionnellement aux nœuds parents.

Le graphe doit de plus satisfaire la condition structurelle suivante : il existe un chemin (comportant des arcs fonctionnels et informationnels) connectant tous les nœuds représentant les variables de décision.

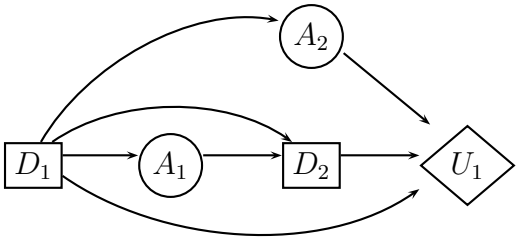



FIG. 1 – Diagramme d'influence.

Un diagramme d'influence est représenté sur la figure 1, où l'on a omis volontairement les tables de probabilité conditionnelle et d'utilité par soucis de brièveté. L'ordre des décisions et des observations est $D_1 - A_1 - D_2 - A_2$. Nous adoptons ici la convention que les variables de décision sont disposées sur le graphe de la gauche vers la droite dans l'ordre chronologique. En développant le diagramme, on retrouve l'arbre décision hasard de la figure 2 (remarquons que les probabilités et les utilités indiquées sur l'arbre respecte les conditionnements imposés par le diagramme). Du fait que $A_2$ et $U_1$ sont indépendants de $A_1$ étant donné $D_1$, les deux sous-arbres en pointillés sont identiques, de même que les deux sous-arbres en gras. Remarquons que la présence de sous-arbres identiques conduit à effectuer plusieurs fois les mêmes calculs lors de la détermination d'une stratégie maximisant EU dans l'arbre décision hasard. Les diagrammes d'influence permettent d'éviter cet écueil, ce que nous détaillons dans le paragraphe qui suit.

L'objet de notre travail est de "résoudre" un diagramme d'influence lorsque les préférences du décideur ne suivent pas le modèle EU mais le modèle RDU. Résoudre le diagramme consiste à déterminer la meilleure stratégie au sens du critère de décision utilisé (EU, RDU ou autre). Pour définir une stratégie, il est nécessaire de connaître les variables aléatoires déjà observées au moment de prendre chaque décision, ainsi que l'ordre chronologique dans lequel les dé-

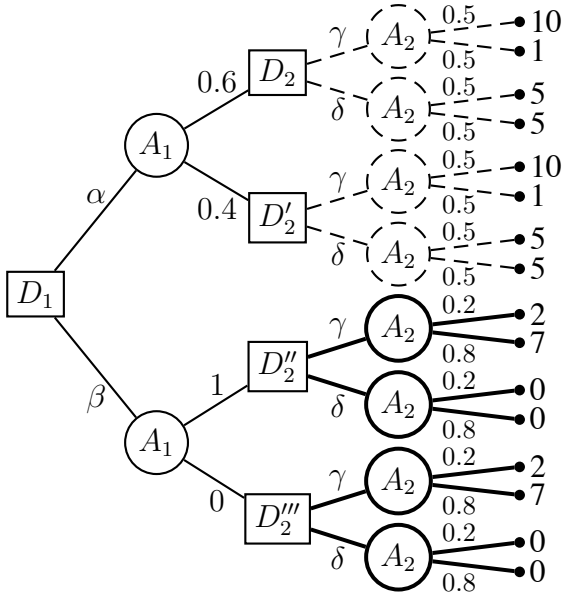



FIG. 2 – Arbre décision hasard.

cisions sont prises. Une stratégie consiste alors à fixer une valeur à chaque variable de décision conditionnellement à son passé. Dans un arbre décision hasard, le passé d'une variable de décision se définit simplement comme l'ensemble des variables aléatoires et des variables de décision situées sur le chemin allant de la racine à cette variable. L'ensemble des stratégies pour l'arbre de la figure 2 comporte 8 stratégies, dont en particulier la stratégie $\{D_1 = \alpha, D_2 = \gamma, D'_2 = \delta\}$ (notons que les nœuds $D''_2$ et $D'''_2$ ne peuvent être atteints lorsqu'on fixe $D_1 = \alpha$). Dans un diagramme d'influence, l'ordre chronologique est moins apparent. C'est pourquoi l'ensemble $\mathcal{N}_A$ est partitionné en des ensembles disjoints $I_0, I_1, \ldots, I_n$. L'ensemble $I_0$ comporte les variables aléatoires observées avant la première décision $D_1$ (correspondant aux parents de $D_1$), $I_k$ les variables aléatoires observées entre $D_k$ et $D_{k+1}$ (correspondant aux parents de $D_{k+1}$), et enfin $I_n$ les variables aléatoires restantes, c.à.d. celles qui ne sont jamais observées ou celles qui sont observées après la dernière décision $D_n$ (i.e., les variables qui ne sont pas parents d'une variable de décision). Cela induit un ordre partiel $\prec$ sur $\mathcal{N}_D \cup \mathcal{N}_A$ : $I_0 \prec D_1 \prec I_1 \prec \ldots \prec D_n \prec I_n$. Par exemple, pour le diagramme de la figure 1, l'ordre partiel est $D_1 \prec \{A_1\} \prec D_2 \prec \{A_2\}$. Le passé d'une variable de décision $D_k$ se définit alors comme l'ensemble des variables $X$ telles que $X \prec D$. Formellement, une stratégie dans un diagramme d'influence est un ensemble de *règles de décision* pour les variables $D_k$, où une règle de décision pour $D_k$ associe une valeur de son domaine pour chaque affectation des variables de son passé. Néanmoins, en pratique dans les diagrammes d'influence, seules les stratégies conséquentialistes (i.e., où chaque décision prise ne dépend que des variables influençant les paramètres futurs à la décision) sont envisagées. De ce fait, il n'est pas nécessaire de connaître les valeurs de toutes les variables du passé pour fixer la décision à prendre. Par exemple, pour le diagramme de la figure 1, du fait que $A_2$ et $U_1$ sont indépendants de $A_1$ étant donné $D_1$, seule la variable $D_1$ influence les valeurs des variables $A_2$ et $U_1$, qui constituent les paramètres décrivant le futur de $D_2$. Comme indiqué en section 1, dans une stratégie conséquentialiste, la décision prise en $D_2$ ne dépend que des variables influençant les paramètres futurs ($A_2$ et $U_1$ ici), autrement dit la décision prise en $D_2$ ne dépend que de $D_1$. Une règle de décision en $D_2$ s'écrira par exemple $D_2 = \gamma | D_1 = \alpha$ et $D_2 = \delta | D_1 = \beta$, ce qui signifie qu'on prend la décision $\gamma$ (resp. $\delta$) en $D_2$ si on a pris la décision $\alpha$ (resp. $\beta$) en $D_1$. Par suite, l'ensemble des stratégies sur le diagramme de la figure 1 comporte les 4 stratégies suivantes : $\{D_1 = \alpha, D_2 = \gamma | D_1 = \alpha\}$ (notons que le cas $D_1 = \beta$ est omis dans la règle de décision en $D_2$ car on a fixé $D_1 = \alpha$), $\{D_1 = \alpha, D_2 = \delta | D_1 = \alpha\}$, $\{D_1 = \beta, D_2 = \gamma | D_1 = \beta\}$ et $\{D_1 = \beta, D_2 = \delta | D_1 = \beta\}$. Il est important de remarquer que l'ensemble des stratégies envisagées dans le diagramme d'influence n'est qu'un sous-ensemble des stratégies envisagées dans l'arbre décision hasard correspondant (par exemple la stratégie mentionnée plus haut pour l'arbre décision hasard ne figure pas dans ce sous-ensemble). Lorsqu'on optimise EU, cela ne porte pas à conséquence car il est bien connu qu'une stratégie EU-optimale se trouve dans ce sous-ensemble (puisque c'est une stratégie conséquentialiste). A contrario, une stratégie optimisant RDU ne se trouve pas nécessairement dans ce sous-ensemble (puisque ce n'est pas nécessairement une stratégie conséquentialiste), comme illustré par l'exemple suivant.

**Exemple 3** *Considérons l'arbre décision hasard de la figure 2. Remarquons tout d'abord que toute stratégie dans laquelle $D_1 = \beta$ et $D''_2 = \delta$ ne peut être optimale au sens de RDU car toutes les conséquences possibles seraient alors nulles. De plus, on est indifférent à la valeur que peut prendre $D'''_2$ puisque cette décision est conditionnée à l'occurence d'un événement de probabilité nulle. Ainsi, seules 5 stratégies*

*sont susceptibles de maximiser le critère RDU :*
$s_1 = \{D_1 = \alpha, D_2 = \gamma, D_2' = \gamma\}$,
$s_2 = \{D_1 = \alpha, D_2 = \gamma, D_2' = \delta\}$,
$s_3 = \{D_1 = \alpha, D_2 = \delta, D_2' = \gamma\}$,
$s_4 = \{D_1 = \alpha, D_2 = \delta, D_2' = \delta\}$,
$s_5 = \{D_1 = \beta, D_2 = \gamma, D_2' = \gamma\}$.
*En supposant que les préférences du décideur suivent le modèle RDU avec la fonction $\varphi$ définie par $\varphi(0) = 0$, $\varphi(p) = 0.45$ si $0 < p \leq 0.7$, $\varphi(p) = 1$ si $p > 0.7$, les valeurs selon RDU des différentes stratégies sont : $RDU(s_1) = RDU((0.5, 10; 0.5, 1)) = 1 + (10 - 1) \times \varphi(0.5) = 5.05$, $RDU(s_2) = 5.05$, $RDU(s_3) = 7.25$, $RDU(s_4) = 5$ et $RDU(s_5) = 7$. L'unique stratégie optimale selon RDU est $s_3$, qui n'est pas conséquentialiste, puisqu'on prend deux décisions différentes en $D_2$ et $D_2'$.*

## 4 Conséquentialisme et cohérence dynamique

L'absence de conséquentialisme pourrait être vu comme un défaut du modèle RDU dans un contexte de décision séquentielle. L'exemple 4 ci-dessous est d'ailleurs souvent utilisé par les partisans de EU pour remettre en cause l'utilisation de RDU dans un contexte de décision séquentielle, car il semble montrer à première vue que le non-conséquentialisme conduirait à adopter des préférences qui ne sont pas *dynamiquement cohérentes*. Les préférences sont dites dynamiquement cohérentes lorsque les décisions (conditionnelles) futures jugées préférables aujourd'hui, à un nœud donné, continuent à l'être demain au nœud où elles devront être effectivement prises. Nous montrons par la suite que la critique de RDU vient d'une confusion entre cette dernière définition de la cohérence dynamique et le principe d'optimalité de la programmation dynamique ("toute sous-stratégie d'une stratégie optimale est optimale").

**Exemple 4** *Considérons un candidat à un jeu télévisé qui a une chance sur dix de remporter le jeu au terme de l'émission. Si il l'emporte il peut repartir immédiatement avec 3000 € ou décider de retenter sa chance lors de l'émission du lendemain, et de repartir avec 4000 € s'il gagne à nouveau. Les statistiques indiquent qu'un candidat ayant gagné une première fois l'emporte neuf fois sur dix une seconde fois. Par soucis de commodité, nous représentons ce problème de décision séquentielle (élémentaire) à l'aide de l'arbre décision hasard de la figure 3,*

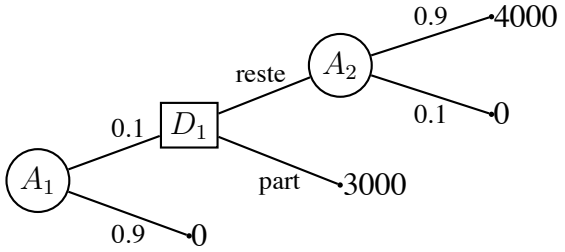



FIG. 3 – RDU et cohérence dynamique.

*mais les arguments qui suivent resteraient valables dans le formalisme des diagrammes d'influence. Au nœud $A_1$, la loterie obtenue pour la stratégie consistant à partir (resp. rester) après une première victoire est la loterie $L_2$ (resp. $L_2'$) de l'exemple 1. Au nœud $D_1$, la loterie obtenue pour la stratégie consistant à partir (resp. rester) après une première victoire est la loterie $L_1$ (resp. $L_1'$) de l'exemple 1. Un candidat présentant le renversement de préférence fréquemment observé face à l'exemple de Allais aurait donc tendance à changer d'avis après avoir gagné une première fois (c'est-à-dire à partir finalement avec 3000 €, alors qu'il prévoyait initialement de viser les 4000 €).*

A la suite de Machina [14], nous expliquons ici pourquoi cette volte-face n'aura pas lieu dès lors que le décideur est résolument non-conséquentialiste. Une différence importante entre un agent $a$ dont les préférences suivent le modèle RDU (ou plus généralement un modèle autre que EU) et un agent $b$ dont les préférences suivent le modèle EU vient en effet de ce que l'agent $a$ accorde encore de l'importance à un risque *déjà encouru* (non-conséquentialisme), contrairement à $b$ qui ne tient plus compte d'un risque une fois qu'il a été encouru. Autrement dit, un scénario contrefactuel joue un rôle dans RDU, contrairement à EU (un agent dont les préférences suivent le modèle EU a en effet un comportement conséquentialiste). Plus formellement, pour faire son choix en $D_1$, l'agent $a$ se conforme toujours aux préférences qu'il avait en $A_1$, c'est-à-dire qu'en $D_1$ il compare toujours les loteries $(0.9, 0; 0.1, 3000)$ et $(0.91, 0; 0.09, 4000)$ ($L_2$ et $L_2'$), et non les loteries $(1, 3000)$ et $(0.9, 0; 0.1, 4000)$ ($L_1$ et $L_1'$). En explicitant les préférences ainsi, on se rend donc compte que l'agent reste cohérent dynamiquement car la sous-stratégie consistant à choisir la décision de rester une fois atteint le nœud $D_1$ reste préférée à celle de partir. Ce type de comportement a été formalisé en particulier par McClennen [15] sous le nom de *choix résolu*.

Le choix résolu désigne le comportement d'un agent qui se fixe un plan initialement et n'en dévie plus ensuite. Dans l'exemple 4, intuitivement, cela signifie que l'agent $a$, une fois établie son intention de participer pour viser les 4000 € ne changera pas d'avis avec la première victoire, afin de rester cohérent avec lui-même. Autrement dit, le modèle RDU vérifie bien la cohérence dynamique sous condition de choix résolu, même s'il ne vérifie pas le principe d'optimalité. C'est pourquoi nous nous intéressons dans la suite de cet article à déterminer un plan optimal au sens de RDU au vu de la situation initiale, et à s'y conformer ensuite. Remarquons que d'autres approches de choix résolu ont déjà été envisagée dans la littérature. Par exemple, Jaffray et Nielsen [9] considèrent chaque nœud de décision comme un *ego* de l'agent, et visent à déterminer une stratégie qui réalise un compromis entre les différents egos. Nous insistons donc sur le fait que ce n'est pas l'approche que nous adoptons ici, même si elle se justifie également.

## 5 Algorithmes

Du fait que la stratégie RDU-optimale n'est pas nécessairement conséquentialiste, une première méthode qui semble naturelle consiste à procéder en deux phases : 1) développer le diagramme d'influence en un arbre décision hasard, puis 2) déterminer la stratégie optimale au sens de RDU directement dans l'arbre décison hasard. Pour la phase 2, on peut s'appuyer sur un algorithme que nous avons proposé récemment [10] dans les arbres décision hasard (rappelons que la violation du principe d'optimalité empêche d'employer directement la programmation dynamique pour déterminer une stratégie optimale au sens de RDU). Toutefois, si elle présente l'avantage de permettre de déterminer la réelle stratégie optimale au sens de RDU, cette méthode est bien évidemment coûteuse en espace mémoire, et devient même impraticable lorsque la taille de l'arbre décision hasard est prohibitive.

C'est pourquoi nous proposons une autre méthode qui tire parti de la structure compacte des diagrammes d'influence (sans procéder à son développement en arbre décision hasard), au prix d'une réduction de l'ensemble des stratégies envisagées. Pour ce faire, une première idée serait d'explorer uniquement l'espace des stratégies conséquentialistes, mais nous perdrions alors une part importante des avantages descriptifs du modèle RDU sur le modèle EU. C'est pourquoi l'approche que nous proposons ici ne renonce pas au non-conséquentialisme : nous introduisons une forme affaiblie de conséquentialisme afin de réaliser un compromis entre potentiel descriptif et compacité de la représentation. Il s'agit d'insérer des arcs fonctionnels supplémentaires dans le diagramme afin d'élargir l'espace (des stratégies) exploré. Remarquons que créer des dépendances supplémentaires factices entre variables indépendantes ne change pas le problème lui-même mais seulement sa représentation. Nous illustrons au travers de l'exemple 5 ci-dessous les changements induit par l'introduction d'un nouvel arc fonctionnel.

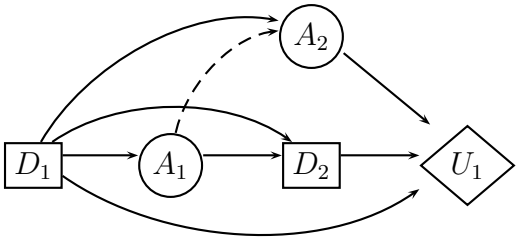



FIG. 4 – Diagramme d'influence modifié.

**Exemple 5** *Après l'ajout de l'arc $(A_1, A_2)$ sur le diagramme d'influence de la figure 1, on obtient le diagramme indiqué sur la figure 4. Du point de la perte de compacité de la représentation, il faut signaler que, si la variable $A_1$ est de modalité 2, cela aura pour effet de doubler le nombre de lignes dans la table assignée à la probabilité d'occurence d'un événement de $A_2$. En effet, la table initiale (avant ajout de l'arc) stockait la probabilité conditionnelle $P(A_2|D_1)$, tandis que la nouvelle table (après ajout de l'arc) représente $P(A_2|D_1, A_1)$. Du point de l'élargissement de l'espace des stratégies considérées, puisque $A_1$ influence les paramètres décrivant le futur de $D_2$ à la suite de l'ajout de l'arc, la décision prise en $D_2$ est maintenant conditionnée par la valeur de $A_1$. Si la variable aléatoire $A_1$ peut prendre deux valeurs $\theta_1$ et $\theta_2$, une règle de décision en $D_2$ s'écrira par exemple :*

$$D_2 = \delta | D_1 = \alpha, A_1 = \theta_1,$$
$$D_2 = \gamma | D_1 = \alpha, A_1 = \theta_2,$$
$$D_2 = \gamma | D_1 = \beta, A_1 = \theta_1,$$
$$D_2 = \delta | D_1 = \beta, A_1 = \theta_2.$$

*Autrement dit, dans ce cas précis, on constate que l'ajout de l'arc $(A_1, A_2)$ permet de tenir compte de toutes les stratégies non-conséquentialistes.*

Par la suite, nous appelons $\Delta$-affaiblissement du conséquentialisme le fait d'ajouter $\Delta$ dépendances supplémentaires pour chaque variable de décision. Cela signifie qu'au pire on rajoute $|\mathcal{N}_D| \times \Delta$ arcs supplémentaires. L'ajout d'un arc dans le diagramme a cependant bien sûr une contrepartie : augmenter la taille d'une table du diagramme, et augmenter la masse de calculs réalisés par l'algorithme de résolution du diagramme. Pour limiter les effets négatifs liés à l'ajout d'un arc, on peut jouer sur deux paramètres :

– en tête de l'arc, la taille (i.e., le nombre de lignes) de la table assignée à la variable ;
– en queue de l'arc, la taille (i.e., le cardinal) du domaine de définition de la variable.

Par exemple, si l'on avait ajouté l'arc $(A_1, U_1)$ sur le diagramme de la figure 1 plutôt que l'arc $(A_1, A_2)$, la décision en $D_2$ serait également devenue dépendante de $A_1$. Cependant, le nombre de lignes de la table d'utilités en $U_1$ aurait été multiplié par 2, passant de 8 à 16 lignes, contre un passage de 2 à 4 lignes en $A_2$ pour l'arc $(A_1, A_2)$. Par ailleurs, il est important de remarquer que l'ajout d'un unique arc $(V, V')$, avec $V \in \mathcal{N}_A \cup \mathcal{N}_D$ et $V' \in \mathcal{N}_A \cup \mathcal{N}_U$, est susceptible de créer plusieurs dépendances $D_i|V$, où $V \prec D_i \prec V'$. Ainsi, il y a un arbitrage à réaliser entre le nombre d'arcs ajoutés et la taille des tables impliquées. Beaucoup d'heuristiques peuvent être envisagées pour accomplir cette tâche. Pour les expérimentations, nous avons développé une heuristique gloutonne adhoc pour les instances particulières générées, que nous explicitons dans la section portant sur les expérimentations numériques.

Nous allons maintenant détailler la procédure permettant de déterminer la stratégie maximisant le critère RDU à la suite d'un $\Delta$-affaiblissement. Nous proposons ici une procédure par séparation et évaluation pour résoudre le diagramme d'influence.

**Initialisation.** L'initialisation consiste à déterminer une stratégie optimisant EU. Plusieurs algorithmes de résolution ont été proposés pour déterminer une stratégie maximisant EU dans un diagramme d'influence [20, 11]. L'algorithme de Shachter [20] consiste à éliminer incrémentalement les noeuds du diagramme d'influence tout en respectant l'ordre partiel sur ceux-ci. Pour ce faire, l'algorithme doit effectuer des inversions d'arcs de dépendance probabiliste, parfois très coûteuses en terme de calculs. L'algorithme de Jensen et al. [11] (inspiré des algorithmes d'inférence dans les *réseaux bayésiens*) consiste à transformer le diagramme d'influence en un *arbre de jonction* et à faire de la programmation dynamique non sérielle [3] dans l'arbre de jonction obtenu. Ses performances étant généralement réputées supérieures à celle de l'algorithme de Shachter, nous avons adopté cette dernière approche dans notre implémentation.

**Principe de branchement.** Le principe de séparation que nous avons utilisé consiste simplement à fixer une valeur à une variable de décision donnée conditionnellement aux variables qui l'influencent. Par exemple, pour le diagramme de la figure 4, on sépare l'ensemble de stratégies caractérisé par :

$(D_1 = \boldsymbol{\alpha}) \wedge (D_2 = \boldsymbol{\gamma} | D_1 = \alpha, A_1 = \theta_1)$

en deux sous-ensembles caractérisés respectivement par :

- $(D_1 = \boldsymbol{\alpha}) \wedge (D_2 = \boldsymbol{\gamma} | D_1 = \alpha, A_1 = \theta_1)$ $\wedge (D_2 = \gamma | D_1 = \alpha, A_1 = \theta_2)$ ;
- $(D_1 = \boldsymbol{\alpha}) \wedge (D_2 = \boldsymbol{\gamma} | D_1 = \alpha, A_1 = \theta_1)$ $\wedge (D_2 = \delta | D_1 = \alpha, A_1 = \theta_2)$.

Nous avons représenté l'arbre d'énumération sur la figure 6. Les nœuds instanciés y sont indiqués en gras. L'ordre d'instanciation des variables dans l'arbre d'énumération est compatible avec l'ordre total sur les nœuds de décision (rappelons qu'il existe toujours une chemin reliant tous les nœuds de décision du diagramme).

**Calcul de la borne inférieure.** En chaque nœud de l'arbre d'énumération, on calcule une stratégie optimisant EU dans le sous-ensemble des stratégies considérées. Lorsque sa valeur selon RDU est plus grande que la valeur de la meilleure stratégie trouvée jusqu'alors, elle prend la place de cette dernière.

**Calcul de la borne supérieure.** Comme nous l'avons précisé, à chaque stratégie on peut associer une loterie (variable aléatoire sur les conséquences potentielles) et c'est l'évaluation de ces loteries par un critère de décision qui permet de comparer les stratégies. Nous basons notre évaluation sur la notion de dominance stochastique entre les loteries (rappelons qu'une loterie $L$ domine stochastiquement une loterie $L'$ si $\forall x \in \mathbb{R}, G_L(x) \geq G_{L'}(x)$). Il s'agit de calculer une loterie qui domine stochastiquement toutes les stratégies potentielles compatibles avec les décisions déjà fixées, son évaluation selon RDU fournissant une borne supérieure (par compatibilité de RDU avec la dominance stochastique, i.e. $RDU(L) \geq RDU(L')$ dès lors que $L$ domine $L'$). Le calcul peut se faire par programmation dynamique, en déterminant récursivement une telle loterie pour chaque variable de décision et chaque instanciation possible des va-

riables de son passé (compatible avec les décisions déjà fixées). Dans la suite, pour simplifier, on suppose qu'il y a une seule variable d'utilité $U$ à valeurs dans $\{u_1, \ldots, u_m\}$ avec $u_1 \leq \ldots \leq u_m$. On procède par induction arrière, l'initialisation en $U$ étant réalisée par la formule suivante :

$$\forall i \geq 1, \quad P(U = u_i | I_0 \cdot\cdot I_n, D_1 \cdot\cdot D_n) = \begin{cases} 1 \text{ si } U(I_0 \cdot\cdot I_n, D_1 \cdot\cdot D_n) = u_i \\ 0 \text{ sinon} \end{cases}$$

où $U(I_0 \cdot\cdot I_n, D_1 \cdot\cdot D_n)$ indique la valeur que prend la variable $U$ en fonction des valeurs des variables $I_0, \ldots, I_n, D_1, \ldots, D_n$. Considérons maintenant une variable de décision $D_k$. Pour une réalisation donnée des variables aléatoires $I_0, \ldots, I_{k-1}$ et un choix donné de décisions en $D_1, \ldots, D_{k-1}$, la loterie dominante en $D_k$ se calcule par l'équation de récurrence suivante :

$$\begin{aligned} &\forall i \geq 1, \quad P(U \geq u_i | I_0 \cdot\cdot I_{k-1}, D_1 \cdot\cdot D_{k-1}) \\ &= \max_{D_k} \sum_{j=i}^{m} \sum_{I_k} P(I_k | I_0 \cdot\cdot I_{k-1}, D_1 \cdot\cdot D_k) \\ &* P(U = u_j | I_0 \cdot\cdot I_{k-1}, I_k, D_1 \cdot\cdot D_k) \end{aligned}$$

La loterie dominante proprement dite se déduit facilement via la formule :

$$\begin{aligned} & P(U = u_i | I_0 \cdot\cdot I_{k-1}, D_1 \cdot\cdot D_{k-1}) \\ = \; & P(U \geq u_i | I_0 \cdot\cdot I_{k-1}, D_1 \cdot\cdot D_{k-1}) \\ & - P(U \geq u_{i+1} | I_0 \cdot\cdot I_{k-1}, D_1 \cdot\cdot D_{k-1}) \end{aligned}$$

La valeur retournée au final est calculée en appliquant RDU à la loterie obtenue en $D_1$. Notons ici qu'en pratique les conditionnements se font sur des ensembles beaucoup plus restreints de variables grâce aux multiples indépendances mises en avant dans le diagramme d'influence. Enfin, précisons que, pour optimiser les calculs, nous avons utilisé, à la manière de l'algorithme de Jensen [11], un arbre de jonction (construit à partir du diagramme d'influence).

**Exemple 6** *Considérons à nouveau l'arbre d'énumération représenté sur la figure 6. Il y a 2 stratégies possibles compatibles avec les règles de décisions déjà fixées, dont les loteries associées sont* $L_1 = (1, 0.5; 10, 0.5)$ *et* $L_2 = (1, 0.3; 5, 0.4; 10, 0.3)$. *Nous indiquons sur le repère de gauche de la figure 5 leurs fonctions décumulatives. La loterie dominante calculée par programmation dynamique correspondra alors à la loterie dont la fonction décumulative est le max de* $G_{L_1}$ *et* $G_{L_2}$. *Nous indiquons cette fonction décumulative en gras sur le repère de droite de la figure 5. Cette loterie dominante est ensuite évaluée selon le critère RDU et nous fournit une borne supérieure des valeurs RDU des stratégies compatibles avec les règles de décisions fixées dans l'arbre d'énumération.*

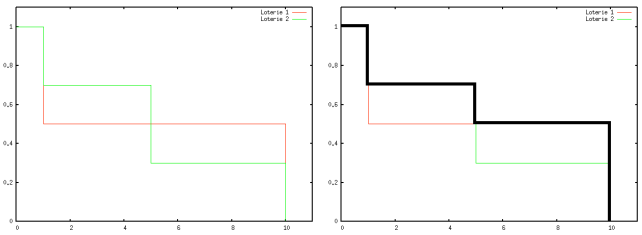


FIG. 5 – Exemple du calcul d'une loterie dominante

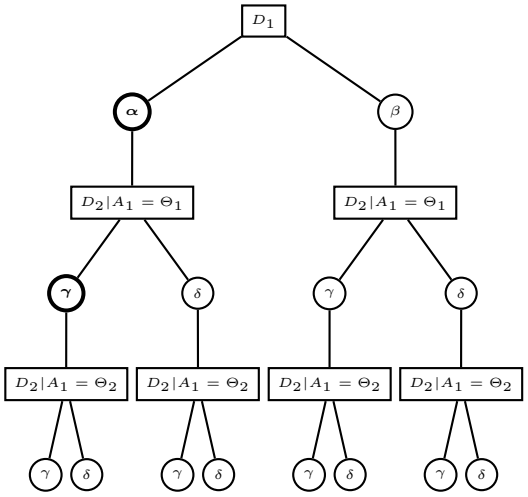



FIG. 6 – Arbre d'énumération.

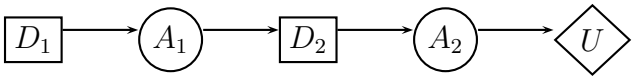
## 6 Expérimentations numériques

Les deux algorithmes proposés (la méthode en deux phases et celle opérant directement sur le diagramme d'influence enrichi de dépendances fictives) ont été implémentés en C++ et les tests numériques ont été menés sur un ordinateur muni d'un processeur Pentium IV à 2.13Ghz et 3.5Go de mémoire vive. Nous présentons ci-dessous les résultats de tests effectués sur des diagrammes d'influence générés aléatoirement. Nous décrivons tout d'abord le fonctionnement du générateur aléatoire, avant de détailler les résultats obtenus, tant au niveau des performances en temps que de la qualité des stratégies retournées. Pour toutes les expérimentations, nous avons utilisé la fonction $\varphi$ de déformation suivante : $\varphi(p) = p^\gamma / (p^\gamma + (1-p)^\gamma)$. Cette fonction de déformation est celle proposée usuellement

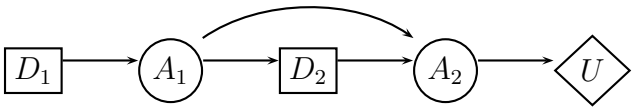
dans la littérature [18]. Le paramètre $\gamma$ est à valeur dans $[0, 1]$. Remarquons que, pour $\gamma = 1$, on a $\varphi(p) = p$ et on retrouve alors le cas particulier EU. Nous avons fixé ici $\gamma = 0.5$.

### 6.1 Génération d'instances

Afin de réellement contrôler la taille des diagrammes d'influence générés, il est nécessaire que tous les nœuds du diagramme entrent en compte lors du calcul de la stratégie optimale. Pour ce faire, étant donné un nombre $n$ fixé de variables de décision, on crée en premier lieu un chemin de longueur $2n+1$ en alternant variables de décisions et variables aléatoires, pour terminer sur une variable d'utilité. Par exemple, pour 2 variables de décision on obtient le chemin suivant :




Ensuite, pour éviter que certaines variables du diagramme d'influence ne jouent aucun rôle réel (i.e., n'aient pas d'impact sur le choix de la stratégie optimale), on impose que chaque variable aléatoire influence une autre variable aléatoire (dès lors qu'il en existe au moins une dans son futur). En reprenant l'exemple précédent on peut alors obtenir :




Enfin, on insère des arcs supplémentaires de manière aléatoire entre certains noeuds.

En ce qui concerne le $\Delta$-affaiblissement par ajout de $\Delta$ dépendances supplémentaires pour chaque variable de décision, nous avons procédé comme suit. Pour chaque variable de décision $D_k$, on rajoute un arc dont le sommet de départ est le nœud précédant $D_k$ et dont le sommet d'arrivée est le nœud (représentant une variable aléatoire ou une variable de décision) de plus petit degré entrant dans le futur de $D_k$. Ensuite, on réitère jusqu'à ajouter $\Delta$ arcs en prenant à chaque fois le sommet $N$ de départ un rang plus tôt dans le passé (dans l'ordre total induit par notre mode de génération des instances), sous condition que $D_k$ ne dépende pas déjà de $N$.

### 6.2 Résultats obtenus

Afin d'évaluer le gain computationnel réalisé lorsque l'on travaille directement sur le diagramme d'influence, nous avons comparé les temps de résolution pour la méthode en deux phases avec les temps de résolution pour différentes valeurs de $\Delta$. Le tableau 2 indique ces temps d'exécutions (temps moyens en seconde sur l'ensemble des instances testées). Dans chaque colonne on fait varier le nombre de variables de décision (i.e. on fait varier la valeur de $n$) et dans chaque ligne on fait varier la valeur de $\Delta$. Dans la dernière ligne du tableau figure le temps mis par la méthode en deux phases. Pour chaque valeur de $n$, 40 instances ont été générées aléatoirement pour lesquelles on a appliqué successivement l'approche par $\Delta$-affaiblissement pour différentes valeurs de $\Delta$ et l'approche en deux phases. On constate sans surprise que plus la taille du diagramme et la valeur de $\Delta$ augmentent, plus le temps de résolution augmente. La méthode en deux phases ne permet pas d'aller au delà de 7 nœuds de décision. La méthode par $\Delta$-affaiblissement permet de traiter des instances de taille plus importantes, au prix de l'optimalité. Nous nous intéressons précisément dans le paragraphe suivant à la qualité des solutions retournées.

| $\Delta \backslash n$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0.21 | 0.85 | 2.78 | 9.11 |
| 1 | 0 | 0 | 0 | 0.23 | 0.62 | 3.94 | 12.15 |
| 2 | 0 | 0 | 0 | 0.38 | 1.10 | 4.79 | 23.36 |
| 3 | 0 | 0 | 0.10 | 0.72 | 2.33 | 9.25 | 29.97 |
| 4 | 0 | 0 | 0.56 | 1.07 | 4.46 | 18.61 | 93.52 |
| 5 | 0 | 0.08 | 1.91 | 3.43 | 19.84 | 87.40 | — |
| 2 ph. | 0.17 | 0.93 | 5.02 | 17.38 | — | — | — |

TAB. 2 – Temps d'exécution (sec.).

Remarquons que la stratégie optimale au sens de EU peut être vue comme une approximation de la stratégie optimale au sens de RDU. Cette approximation peut être calculée rapidement en utilisant les algorithmes classiques dans les diagrammes d'influence. Etant donné un diagramme d'influence $DI$, notons $RDU_{EU}(DI)$ (resp. $RDU_{\Delta}(DI)$) la valeur RDU de la stratégie optimisant EU (resp. optimisant RDU dans $DI$ au terme d'un $\Delta$-affaiblissement du conséquentialisme). Afin d'évaluer l'influence du paramètre $\Delta$ sur la qualité des stratégies obtenues au sens de RDU, nous avons calculé le ratio $(RDU_{\Delta}(DI) - RDU_{EU}(DI))/RDU_{EU}(DI)$ pour différentes valeurs de $\Delta$ et du nombre $n$ de variables de décision. Pour chaque couple

$(\Delta, n)$, 200 instances $DI$ ont été aléatoirement générées. Les résultats obtenus sont indiqués dans le tableau 3. Dans chaque case du tableau on indique le ratio moyen (exprimé en pourcentage) sur les 200 instances, ainsi que le plus grand ratio obtenu. Dans la dernière ligne du tableau, on indique les résultats toutes valeurs de $n$ confondues. On remarque que le ratio moyen augmente significativement lorsque $\Delta$ croît. Par ailleurs, le gain obtenu est monté jusqu'à $48\%$ sur certaines instances.

| $n$ | $\Delta=2$ | | $\Delta=3$ | | $\Delta=4$ | | $\Delta=5$ | |
|---|---|---|---|---|---|---|---|---|
| | moy | max | moy | max | moy | max | moy | max |
| 2 | 11% | 26% | — | | — | | — | |
| 3 | 6% | 9% | 6% | 11% | 13% | 32% | — | |
| 4 | 3% | 6% | 7% | 23% | 18% | 48% | 14% | 36% |
| 5 | 3% | 8% | 4% | 13% | 10% | 23% | 14% | 40% |
| 6 | 1% | 11% | 9% | 27% | 7% | 28% | 11% | 30% |
| | 5% | 26% | 6% | 27% | 12% | 48% | 13% | 40% |

TAB. 3 – Influence du paramètre $\Delta$ sur la qualité des stratégies obtenues.

## 7 Conclusion

Dans cet article, nous avons proposé une approche algorithmique pour optimiser RDU dans un diagramme d'influence, en visant à concilier des préoccupations d'optimalité (obtenir une stratégie dont la valeur RDU soit la plus proche possible de l'optimum) et de compacité de représentation (obtenir une stratégie dont le stockage en machine ne soit pas trop gourmand en mémoire). Nous avons montré que l'espace des stratégies habituellement envisagées dans les diagrammes d'influence (espace des stratégies dites *conséquentialistes*) est bien plus restreint que l'espace des stratégies potentielles dans l'arbre décision hasard obtenu par développement du diagramme (espace qui inclut bien sûr le sous-espace précédent mais aussi l'ensemble des stratégies non-conséquentialistes). C'est pourquoi, dans notre approche, le diagramme d'influence est muni d'arcs supplémentaires représentant des dépendances fictives entre décision afin d'enrichir l'espace exploré par un sous-ensemble de stratégies non-conséquentialistes ($\Delta$-affaiblissement du conséquentialisme, qui correspond à l'ajout de $\Delta$ arcs fictifs par nœud de décision). Les expérimentations numériques menées montrent l'intérêt de procéder à ce $\Delta$-affaiblissement car la valeur RDU de la stratégie calculée améliore parfois considérablement la valeur obtenue pour une stratégie conséquentialiste.

Afin d'améliorer l'approche proposée ici, la piste de recherche la plus prometteuse nous semble être un travail sur l'identification des arcs dont l'ajout dans le diagramme d'influence joue le plus fortement sur la valeur RDU de la stratégie retournée. En effet, cela permettrait à la fois de limiter la croissance de la taille du diagramme (liée à l'ajout d'arcs) et d'améliorer la qualité de la stratégie retournée. Dans une toute autre direction, on pourrait étudier l'utilisation de RDU dans les processus décisionnels markoviens. La difficulté dans ce nouveau cadre est que la stratégie optimisant RDU à horizon infini n'est pas nécessairement markovienne. Un affaiblissement de la propriété markovienne similaire à l'affaiblissement du conséquentialisme présenté ici pourrait être envisagé.

## Références


[1] M. Allais, Le comportement de l'homme rationnel devant le risque : critique des postulats de l'école américaine, *Econometrica*, Vol. 21, pp. 503-546, 1953.

[2] M. Allais, An outline of my main contributions to economic science, *The American Economic Review*, Vol. 87, Num. 6, pp. 3-12, 1997.

[3] U. Bertelé et F. Brioschi, *Nonserial Dynamic Programming*, Academic Press, 1972.

[4] T. Dean, Planning with deadlines in stochastic domains, *Proc. of the 11th AAAI*, pp. 574-579, 1993.

[5] J.-P. Gayant, *Risque et décision,* Vuibert, 2001.

[6] J. Handa, Risk, probabilities and a new theory of cardinal utility, *Journal of Political Economics*, Vol. 85, pp. 97-122, 1977.

[7] R. Howard et J. Matheson, Influence Diagrams, *Readings on the Principles and Applications of Decision Analysis,* pp. 721-762, 1984.

[8] J.-Y. Jaffray, Décision séquentielle et rationalité, *Actes de PDMIA*, 2004.

[9] J.-Y. Jaffray et T. Nielsen, An operational approach to rational decision making based on rank dependent utility, *European Journal of Operational Research*, Vol. 169, Num. 1, pp. 226-246, 2006.

[10] G. Jeantet et O. Spanjaard, Rank-Dependent Probability Weighting in Sequential Decision Problems under Uncertainty, *Proc. of ICAPS*, pp. 148-155, 2008.

[11] F. Jensen, F.V. Jensen et S.L. Dittmer, From Influence Diagrams to Junction Trees, *UAI 1994*, pp. 367-373, 1994.

[12] D. Kahneman et A. Tversky, Prospect theory : An analysis of decision under risk, *Econometrica*, Vol 47, pp. 263-291, 1979.

[13] Y. Liu et S. Koenig, An exact algorithm for solving MDPs under risk-sensitve planning objectives with one-switch utility functions, *Proc. of the Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 453-460, 2008.

[14] M.J. Machina, Dynamic Consistency and Non-Expected Utility Models of Choice Under Uncertainty, *Journal of Economic Literature*, pp. 1622-1668, 1989.

[15] E. McClennen, *Rationality and Dynamic choice : Foundational Explorations*, Cambridge University Press, 1990.

[16] P. Perny, O. Spanjaard et L.-X. Storme, State space search for risk-averse agents, *20th International Joint Conference on Artificial Intelligence*, pp. 2353-2358, 2007.

[17] J. Quiggin, A theory of anticipated utility, *Journal of Economic Behavior and Organisation*, Vol. 3, Num. 4, pp. 323-343, 1982.

[18] J. Quiggin, *Generalized Expected Utility Theory : The Rank-Dependent Model,* Kluwer, 1993.

[19] H. Raiffa, *Decision Analysis : Introductory Lectures on Choices under Uncertainty,* Addison-Wesley, 1968.

[20] R. Shachter, Evaluating influence diagrams, *Operations Research*, Vol. 34, pp. 871-882, 1986.

[21] J. von Neuman et O. Morgenstern, *Theory of games and economic behaviour,* Princeton University Press, 1947.

# Opérateurs de Confluence

Sébastien Konieczny* Ramón Pino Pérez**
konieczny@cril.fr pino@ula.ve

*CRIL - CNRS
Faculté des Sciences, Université d'Artois, Lens, France

**Departamento de Matemáticas
Facultad de Ciencias, Universidad de Los Andes, Mérida, Venezuela

**Résumé :**
Dans le cadre des approches logiques pour la représentation des connaissances et le raisonnement, beaucoup d'opérateurs ont été définis pour capturer différents types de changement : révision, mise-à-jour, fusion, et beaucoup d'autres. Il y a de forts liens entre révision, mise à jour et fusion. Les opérateurs de fusion peuvent être considérés comme une extension des opérateurs de révision à plusieurs bases. Et les opérateurs de mise à jour peuvent être vus comme des révisions ponctuelles, s'intéressant à chaque modèle de la base, plutôt que de considérer la base globalement. A partir de là, une question naturelle est : Existe-t-il des opérateurs qui sont des fusions ponctuelles, comme la mise à jour est une révision ponctuelle ? Le but de cet artice est de répondre positivement à cette question. A cette fin nous introduisons une nouvelle classe d'opérateurs de changement : les opérateurs de confluence. Ces nouveaux opérateurs peuvent être utiles pour la modélisation de processus de négociation.

**Mots-clés :** Conciliation, Fusion, Mise à jour, Négociation

**Abstract:**
In the logic based framework of knowledge representation and reasoning many operators have been defined in order to capture different kinds of change : revision, update, merging and many others. There are close links between revision, update, and merging. Merging operators can be considered as extensions of revision operators to multiple belief bases. And update operators can be considered as pointwise revision, looking at each model of the base, instead of taking the base as a whole. Thus, a natural question is the following one : Are there natural operators that are pointwise merging, just as update are pointwise revision ? The goal of this work is to give a positive answer to this question. In order to do that, we introduce a new class of operators : the confluence operators. These new operators can be useful in modelling negotiation processes.

**Keywords:** Conciliation, Belief Merging, Update, Negotiation

## 1 Introduction

La théorie de la dynamique des croyances a produit beaucoup d'opérateurs différents, qui permettent de modéliser les changement des croyances d'un (ou de plusieurs) agent(s). Parmi ceux-ci ont peut citer par exemple la révision [1, 5, 10, 6], la mise à jour [9, 8], la fusion [19, 14], l'abduction [16], l'extrapolation [4], etc.

Dans ce papier on se focalisera sur la révision, la mise à jour et la fusion. Commençons par décrire ces opérateurs de manière informelle :

**Révision** La révision de croyances est le processus qui consiste à incorporer dans les croyances de l'agent une nouvelle information plus fiable. En révision de croyances le monde est statique, ce ne sont que les croyances de l'agent à propos de ce monde qui évoluent.

**Mise à jour** Dans le cadre de la mise à jour, la nouvelle information concerne un changement qui est intervenu dans le monde. Le monde est dynamique, et ces changements, lorsqu'ils sont observés, modifient les croyances de l'agent.

**Fusion** La fusion de croyances consiste à définir les croyances d'un groupe d'agents à partir des croyances (individuelles) de chaque agent.

Ces opérateurs ont donc des utilisations bien différentes. Néanmoins il existe tout de même de forts liens entre ceux-ci. Ceci est particulièrement clair lorsque l'on regarde les définitions techniques.

Les opérateurs de révision [1, 5, 10] et les opérateurs de mise à jour [9] ont des définitions assez proches. La principale différence est que les premiers considèrent les croyances d'un agent globallement, alors que les seconds les considèrent localement [1].

Il y a également un lien entre la révision et la fusion. En fait, les opérateurs de révision peuvent être vus comme un cas particulier d'opérateur de fusion.

A partir de ces deux derniers faits, une question naturelle surgit : Quelle est la famille d'opérateurs qui sont une généralisation des opérateurs de mise à jour, de la même manière que les opérateurs de fusion généralisent les opérateurs de révision ?

[1] Voir [8, 4, 15] pour plus de détails sur la mise à jour et ses liens avec la révision.

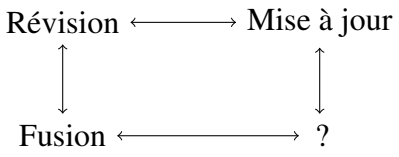



FIG. 1 – Confluence

Ou, dit autrement, quels sont les opérateurs qui peuvent être considérés comme une fusion ponctuelle, de la même manière que les opérateurs de mise à jour peuvent être considérés comme une révision ponctuelle.

Nous pouvons résumer cela par la figure ci-dessus. Le but de cet article est d'introduire et d'étudier les opérateurs correspondant au point d'interrogation. Nous appelons ces nouveaux opérateurs, opérateurs de confluence.

Ces nouveaux opérateurs sont plus prudents que les opérateurs de fusion. Cela suggère qu'ils peuvent être utilisés pour définir des opérateurs de négociation (voir [2, 20, 18, 17, 12]), ou comme une première étape d'un processus de négociation, afin de trouver tous les résultats possibles de la négociation, puis de sélectionner les plus adéquats.

Pour illustrer l'intérêt de ces opérateurs et la différence de comportement avec les opérateurs de fusion, nous proposons le petit exemple suivant.

**Exemple 1** *Sonia et Bruno souhaitent acheter une voiture. Sonia n'aime pas les voitures allemandes, ni les voitures chères. Et elle aime les petites voitures. Bruno hésite entre une petite voiture allemande et chère, et une grande voiture qui ne serait pas chère et pas allemande.*

*En considérant les trois variables propositionnelles* `Allemande`*,* `Chere`*, et* `Petite` *dans cet ordre, les souhaits de Sonia se représentent donc par* $mod(S) = \{001\}$ *et ceux de Bruno par* $mod(B) = \{111, 000\}$.

*Avec la plupart des opérateurs de fusion*[2] *le résultat serait* $\{001, 000\}$. *C'est-à-dire le même résultat que l'on obtiendrait si les souhaits de Bruno étaient simplement une grande voiture qui ne serait pas chère et pas allemande (*$mod(B') = \{000\}$*).*

*Les opérateurs de confluence prennent en compte la nature disjonctive des souhaits de Bruno, en ajoutant également au résultat des interprétations qui sont un compromis entre* $001$ *et* $111$*. Par exemple les interprétations* $011$ *et* $101$ *viendront s'ajouter à* $001$ *et* $000$ *si on utilise l'opérateur de confluence* $\diamondsuit^{d_H, Gmax}$ *(défini section 7).*

Ce genre d'opérateurs est particulièrement intéressant lorsque les bases représentent une situation qui n'est pas parfaitement connue (on ne connaît pas l'ensemble des souhaits des agents), ou qui peut évoluer dans le futur (c'est-à-dire que les agents peuvent ensuite continuer à contraindre leur ensemble de souhaits).

Par exemple les souhaits de Bruno peuvent être soit imparfaitement connus (il désire une des deux situations, mais nous ne savons pas laquelle), ou ses souhaits peuvent évoluer dans le futur (il choisira plus tard entre les deux situations). Dans ces situations les solutions proposées par les opérateurs de confluence sont plus adéquates que celles issues des opérateurs de fusion.

La prochaine section contient les définitions et notations nécessaires à l'article. Dans la section 3 nous rappelons les définitions axiomatiques et les théorèmes de représentation de la révision, de la mise à jour et de la fusion, et les liens entre ces opérateurs. Dans la section 4 nous définissons les opérateurs de confluence. Nous proposons un théorème de représentation pour ces opérateurs dans la section 5. Dans la section 6 nous étudions les liens entre les opérateurs de confluence et la mise à jour et la fusion. Enfin dans la section 7 nous donnons des exemples d'opérateurs de confluence, avant de conclure dans la section 8.

## 2 Preliminaries

On considère un langage propositionnel $\mathcal{L}$, défini à partir d'un ensemble fini de variables propositionnelles $\mathcal{P}$ et des connecteurs standards.

Une interprétation $\omega$ est une fonction de $\mathcal{P}$ dans $\{0, 1\}$. L'ensemble de toutes les interprétations est noté $\mathcal{W}$. Une interprétation $\omega$ est un modèle d'une formule $\phi \in \mathcal{L}$ si elle la rend vrai. $mod(\varphi)$ dénote l'ensemble des modèles de la formule $\varphi$, i.e., $mod(\varphi) = \{\omega \in \mathcal{W} \mid \omega \models \varphi\}$. Lorsque $M$ est un ensemble de modèles, on note par $\varphi_M$ une formule telle que $mod(\varphi_M) = M$.

Une *base* $K$ est un ensemble fini de formules

[2] Comme par exemple $\triangle^{d_H, \Sigma}$ et $\triangle^{d_H, Gmax}$ [14].

propositionnelles. Afin de simplifier les notations, on identifiera la base $K$ avec la formule $\varphi$ qui est la conjonction des formule de $K$[3].

Un *profil* $\Psi$ est un multi-ensemble non vide de bases $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ (donc deux agents peuvent avoir des bases identiques), et représente un groupe de $n$ agents. L'union sur les multi-ensembles est noté $\sqcup$.

On note $\bigwedge \Psi$ la conjonction des bases de $\Psi = \{\varphi_1, \ldots, \varphi_n\}$, i.e., $\bigwedge \Psi = \varphi_1 \wedge \ldots \wedge \varphi_n$.

Un profil $\Psi$ est dit consistant si et seulement si $\bigwedge \Psi$ est consistant.

Une formule $\varphi$ est complete si elle a un unique modèle. Un profil $\Psi$ est complet si toutes les bases de $\Psi$ sont des formules complètes.

Si $\leq$ est un pré-ordre sur $\mathcal{W}$ (i.e. une relation transitive et réflexive), alors $<$ dénote l'ordre strict associé, défini par $\omega < \omega'$ si et seulement si $\omega \leq \omega'$ et $\omega' \not\leq \omega$, et $\simeq$ dénote la relation d'équivalence associée, définie par $\omega \simeq \omega'$ si et seulement si $\omega \leq \omega'$ et $\omega' \leq \omega$. Un pré-ordre est *total* si $\forall \omega, \omega' \in \mathcal{W}$, $\omega \leq \omega'$ ou $\omega' \leq \omega$. Un pré-ordre qui n'est pas total est *partiel*.

Soit un pré-ordre $\leq$ sur $A$, et $B \subseteq A$, alors $\min(B, \leq) = \{b \in B \mid \nexists a \in B \; a < b\}$.

# 3 Revision, mise à jour et fusion

Dans cette section nous allons rappeler la définition des opérateurs de révision, mise à jour et fusion. Nous rappellerons également leur théorème de représentation respectifs en terme de pré-ordres sur les interprétations. Cela nous permettra de souligner les liens entre ces opérateurs.

## 3.1 Révision

**Définition 1 (Katsuno-Mendelzon [10])** *Un opérateur $\circ$ est un opérateur de révision AGM si il satisfait les propriétés suivantes :*

**(R1)** $\varphi \circ \mu \vdash \mu$

**(R2)** *Si* $\varphi \wedge \mu \nvdash \bot$ *alors* $\varphi \circ \mu \equiv \varphi \wedge \mu$

**(R3)** *Si* $\mu \nvdash \bot$ *alors* $\varphi \circ \mu \nvdash \bot$

**(R4)** *Si* $\varphi_1 \equiv \varphi_2$ *et* $\mu_1 \equiv \mu_2$ *alors* $\varphi_1 \circ \mu_1 \equiv \varphi_2 \circ \mu_2$

**(R5)** $(\varphi \circ \mu) \wedge \phi \vdash \varphi \circ (\mu \wedge \phi)$

**(R6)** *Si* $(\varphi \circ \mu) \wedge \phi \nvdash \bot$ *alors* $\varphi \circ (\mu \wedge \phi) \vdash (\varphi \circ \mu) \wedge \phi$

Lorsque l'on travaille avec un langage propositionnel fini ces postulats, proposés par Katsuno et Mendelzon, sont équivalents aux postulats AGM [1, 5]. Dans [10] Katsuno et Mendelzon donnent également un théorème de représentation pour les opérateurs de révision, montrant que chaque opérateur de révision correspond à un assignement fidèle, qui est une fonction qui associe à chaque base un pré-ordre sur les interprétations. Ce pré-ordre représentant la plausibilité des différentes interprétations. Cette idée peut être rapprochée des systèmes de sphères de Grove [7].

**Définition 2** *Un assignement fidèle est une fonction qui associe à chaque base $\varphi$ un pré-ordre $\leq_\varphi$ sur les interprétations tel que :*

1. *Si* $\omega \models \varphi$ *et* $\omega' \models \varphi$, *alors* $\omega \simeq_\varphi \omega'$
2. *Si* $\omega \models \varphi$ *et* $\omega' \not\models \varphi$, *alors* $\omega <_\varphi \omega'$
3. *Si* $\varphi \equiv \varphi'$, *alors* $\leq_\varphi = \leq_{\varphi'}$

**Théorème 1 (Katsuno-Mendelzon [10])** *Un opérateur $\circ$ est un opérateur de révision (i.e. il satisfait (R1)-(R6)) si et seulement si il existe un assignement fidèle qui associe à chaque base $\varphi$ un pré-ordre total $\leq_\varphi$ tel que*

$$mod(\varphi \circ \mu) = \min(mod(\mu), \leq_\varphi).$$

Ce théorème de représentation est important puisqu'il fournit un moyen de définir facilement des opérateurs de révision en définissant les assignements fidèles. Mais également parce qu'il existe des théorèmes similaires pour la mise à jour et pour la fusion, et que cette représentation en termes d'assignements permet de trouver facilement les liens entre ces opérateurs.

## 3.2 Mise à jour

**Définition 3 (Katsuno-Mendelzon [9, 11])** *Un opérateur $\diamond$ est un opérateur de mise à jour (partielle) si il satisfait les propriétés* **(U1)-(U8)**. *C'est un opérateur de mise à jour totale si il satisfait les propriétés* **(U1)-(U5), (U8), (U9)**.

[3] Certaines approches sont sensibles à la représentation syntaxique. Dans ce cas il est important de faire une distinction entre $K$ et la conjonction de ses formules (voir e.g. [13]). Mais les opérateurs que nous définissons dans cet articles sont tous indépendants de la syntaxe.

**(U1)** $\varphi \diamond \mu \vdash \mu$
**(U2)** *Si* $\varphi \vdash \mu$, *alors* $\varphi \diamond \mu \equiv \varphi$
**(U3)** *Si* $\varphi \nvdash \bot$ *et* $\mu \nvdash \bot$ *alors* $\varphi \diamond \mu \nvdash \bot$
**(U4)** *Si* $\varphi_1 \equiv \varphi_2$ *et* $\mu_1 \equiv \mu_2$ *alors* $\varphi_1 \diamond \mu_1 \equiv \varphi_2 \diamond \mu_2$
**(U5)** $(\varphi \diamond \mu) \wedge \phi \vdash \varphi \diamond (\mu \wedge \phi)$
**(U6)** *Si* $\varphi \diamond \mu_1 \vdash \mu_2$ *et* $\varphi \diamond \mu_2 \vdash \mu_1$, *alors* $\varphi \diamond \mu_1 \equiv \varphi \diamond \mu_2$
**(U7)** *Si* $\varphi$ *est une formule complète, alors* $(\varphi \diamond \mu_1) \wedge (\varphi \diamond \mu_2) \vdash \varphi \diamond (\mu_1 \vee \mu_2)$
**(U8)** $(\varphi_1 \vee \varphi_2) \diamond \mu \equiv (\varphi_1 \diamond \mu) \vee (\varphi_2 \diamond \mu)$
**(U9)** *Si* $\varphi$ *est une formule complète et* $(\varphi \diamond \mu) \wedge \phi \nvdash \bot$, *alors* $\varphi \diamond (\mu \wedge \phi) \vdash (\varphi \diamond \mu) \wedge \phi$

Comme pour la révision, il y a un théorème de représentation en termes d'assignements fidèles.

**Définition 4** *Un assignement fidèle est une fonction qui associe à chaque interprétation* $\omega$ *un pré-ordre sur les interprétations* $\leq_\omega$ *tel que si* $\omega \neq \omega'$, *alors* $\omega <_\omega \omega'$.

On peut facilement vérifier que ces assignements fidèles sur les interprétations sont justes un cas particulier des assignements fidèles sur les bases définis dans la section précédente lorsque l'on considère les bases complètes correspondant aux interprétations.

Katsuno et Mendelzon donnent deux théorèmes de représentation pour les opérateurs de mise à jour. Le premier théorème correspond à la mise à jour partielle.

**Théorème 2 (Katsuno-Mendelzon [9, 11])** *Un opérateur de mise à jour* $\diamond$ *satisfait (U1)-(U8) si et seulement si il existe un assignement fidèle qui associe à chaque interprétation* $\omega$ *un pré-ordre partiel* $\leq_\omega$ *tel que*

$$mod(\varphi \diamond \mu) = \bigcup_{\omega \models \varphi} \min(mod(\mu), \leq_{\varphi_{\{\omega\}}})$$

Le second théorème correspond aux pré-ordres totaux.

**Théorème 3 (Katsuno-Mendelzon [9, 11])** *Un opérateur de mise à jour* $\diamond$ *satisfait (U1)-(U5), (U8) et (U9) si et seulement si il existe un assignement fidèle qui associe à chaque interprétation* $\omega$ *un pré-ordre total* $\leq_\omega$ *tel que*

$$mod(\varphi \diamond \mu) = \bigcup_{\omega \models \varphi} \min(mod(\mu), \leq_{\varphi_{\{\omega\}}})$$

### 3.3 Fusion

**Définition 5 (Konieczny-Pino Pérez [14])** *Un opérateur* $\triangle$ *est un opérateur de fusion contrainte (IC merging) si il satisfait les propriétés suivantes :*

**(IC0)** $\triangle_\mu(\Psi) \vdash \mu$
**(IC1)** *Si* $\mu \nvdash \bot$, *alors* $\triangle_\mu(\Psi) \nvdash \bot$
**(IC2)** *Si* $\bigwedge \Psi \wedge \mu \nvdash \bot$, *alors* $\triangle_\mu(\Psi) \equiv \bigwedge \Psi \wedge \mu$
**(IC3)** *Si* $\Psi_1 \equiv \Psi_2$ *et* $\mu_1 \equiv \mu_2$, *alors* $\triangle_{\mu_1}(\Psi_1) \equiv \triangle_{\mu_2}(\Psi_2)$
**(IC4)** *Si* $\varphi_1 \vdash \mu$ *et* $\varphi_2 \vdash \mu$, *alors* $\triangle_\mu(\{\varphi_1, \varphi_2\}) \wedge \varphi_1 \nvdash \bot$ *si et seulement si* $\triangle_\mu(\{\varphi_1, \varphi_2\}) \wedge \varphi_2 \nvdash \bot$
**(IC5)** $\triangle_\mu(\Psi_1) \wedge \triangle_\mu(\Psi_2) \vdash \triangle_\mu(\Psi_1 \sqcup \Psi_2)$
**(IC6)** *Si* $\triangle_\mu(\Psi_1) \wedge \triangle_\mu(\Psi_2) \nvdash \bot$, *alors* $\triangle_\mu(\Psi_1 \sqcup \Psi_2) \vdash \triangle_\mu(\Psi_1) \wedge \triangle_\mu(\Psi_2)$
**(IC7)** $\triangle_{\mu_1}(\Psi) \wedge \mu_2 \vdash \triangle_{\mu_1 \wedge \mu_2}(\Psi)$
**(IC8)** *Si* $\triangle_{\mu_1}(\Psi) \wedge \mu_2 \nvdash \bot$, *alors* $\triangle_{\mu_1 \wedge \mu_2}(\Psi) \vdash \triangle_{\mu_1}(\Psi)$

Il y a également un théorème de représentation en termes de pré-ordres sur les interprétations pour les opérateurs de fusion [14].

**Définition 6** *Un* assignement syncrétique *est une fonction qui associe à chaque profil* $\Psi$ *un pré-ordre total sur les interprétations* $\leq_\Psi$ *tel que :*

1. *Si* $\omega \models \Psi$ *et* $\omega' \models \Psi$, *alors* $\omega \simeq_\Psi \omega'$
2. *Si* $\omega \models \Psi$ *et* $\omega' \not\models \Psi$, *alors* $\omega <_\Psi \omega'$
3. *Si* $\Psi_1 \equiv \Psi_2$, *alors* $\leq_{\Psi_1} = \leq_{\Psi_2}$
4. $\forall \omega \models \varphi \; \exists \omega' \models \varphi' \; \omega' \leq_{\{\varphi\} \sqcup \{\varphi\}'} \omega$
5. *Si* $\omega \leq_{\Psi_1} \omega'$ *et* $\omega \leq_{\Psi_2} \omega'$, *alors* $\omega \leq_{\Psi_1 \sqcup \Psi_2} \omega'$
6. *Si* $\omega <_{\Psi_1} \omega'$ *et* $\omega \leq_{\Psi_2} \omega'$, *alors* $\omega <_{\Psi_1 \sqcup \Psi_2} \omega'$

**Théorème 4 (Konieczny-Pino Pérez [14])** *Un opérateur* $\triangle$ *est un opérateur de fusion contrainte si et seulement si il existe un assignement syncrétique qui associe à chaque profil* $\Psi$ *un pré-ordre total* $\leq_\Psi$ *tel que :*

$$mod(\triangle_\mu(\Psi)) = \min(mod(\mu), \leq_\Psi)$$

### 3.4 Révision et mise à jour

Intuitivement les opérateurs de révision transforment la base minimalement en sélectionnant les modèles les plus plausibles parmi les modèles de la nouvelle information. Alors que les opérateurs de mise à jour apportent un changement minimal à chaque monde possible (modèle) de la base, afin de prendre en compte le changement décrit par la nouvelle information pour chacun de ces mondes possibles.

Donc, si l'on examine les théorèmes de représentation (théorèmes 1, 2 et 3), on obtient facilement le résultat suivant :

**Théorème 5** *Si $\circ$ est un opérateur de révision (i.e. il satisfait (R1)-(R6)), alors l'opérateur $\diamond$ défini par :*

$$\varphi \diamond \mu = \bigvee_{\omega \models \varphi} \varphi_{\{\omega\}} \circ \mu$$

*est un opérateur de mise à jour qui satisfait (U1)-(U9).*

*De plus, pour chaque opérateur de mise à jour $\diamond$, il existe un opérateur de révision $\circ$ qui lui correspond au sens de cette équation.*

Cette proposition explique donc formellement en quoi la mise à jour peut être considérée comme une révision ponctuelle.

### 3.5 Révision et fusion

Les opérateurs de révision sélectionnent dans une formule (la nouvelle information) l'information la plus proche d'une information de base (la base actuelle). Similairement, les opérateurs de fusion contraintes sélectionnent dans une formule (les contraintes d'intégrité) l'information la plus proche d'une information de base (le profil). A partir de ce parallèle, on peut donc facilement envisager le lien entre fusion contrainte et révision [14] :

**Théorème 6 (Konieczny-Pino Pérez [14])** *Si $\triangle$ est un opérateur de fusion contrainte (il satisfait (IC0-IC8)), alors l'opérateur $\circ$, défini par $\varphi \circ \mu = \triangle_\mu(\varphi)$, est un opérateur de révision AGM (il satisfait (R1-R6)).*

Voir [14] pour plus de détails sur les liens entre la révision et la fusion contrainte.

## 4 Opérateurs de confluence

A présent que l'on a étudié précisemment les connections décrites dans la figure 1 entre la révision, la mise à jour et la fusion, nous pouvons définir les opérateurs de confluence, que l'on désire définir comme des fusions ponctuelles, tout comme la mise à jour peut être considérée comme une révision ponctuelle, comme décrit section 3.4.

Définissons tout d'abord la notion de p-consistance pour les profils.

**Définition 7** *Un profil $\Psi = \{\varphi_1, \ldots \varphi_n\}$ est* p-consistant *si toutes ses bases sont consistantes, i.e $\forall \varphi_i \in \Psi$, $\varphi_i$ est consistant.*

Notons que la notion de p-consistance est bien plus faible que celle de consistance pour un profil. La première notion demande juste que toutes les bases du profil soient consistantes, la seconde exige que leur conjonction soit consistante.

**Définition 8** *Un operator $\diamond$ est un opérateur de confluence si il satisfait les propriétés suivantes :*

**(UC0)** $\diamond_\mu(\Psi) \vdash \mu$

**(UC1)** *Si* $\mu \not\vdash \bot$ *et* $\Psi$ *is p-consistent, alors* $\diamond_\mu(\Psi) \not\vdash \bot$

**(UC2)** *Si* $\Psi$ *est complet,* $\Psi \not\vdash \bot$ *et* $\bigwedge \Psi \vdash \mu$, *alors* $\diamond_\mu(\Psi) \equiv \bigwedge \Psi$

**(UC3)** *Si* $\Psi_1 \equiv \Psi_2$ *et* $\mu_1 \equiv \mu_2$, *alors* $\diamond_{\mu_1}(\Psi_1) \equiv \diamond_{\mu_2}(\Psi_2)$

**(UC4)** *Si* $\varphi_1$ *et* $\varphi_2$ *sont des formules completes et* $\varphi_1 \vdash \mu$, $\varphi_2 \vdash \mu$, *alors* $\diamond_\mu(\{\varphi_1, \varphi_2\}) \wedge \varphi_1 \not\vdash \bot$ *si et seulement si* $\diamond_\mu(\{\varphi_1, \varphi_2\}) \wedge \varphi_2 \not\vdash \bot$

**(UC5)** $\diamond_\mu(\Psi_1) \wedge \diamond_\mu(\Psi_2) \vdash \diamond_\mu(\Psi_1 \sqcup \Psi_2)$

**(UC6)** *Si* $\Psi_1$ *et* $\Psi_2$ *sont des profils complets et* $\diamond_\mu(\Psi_1) \wedge \diamond_\mu(\Psi_2) \not\vdash \bot$, *alors* $\diamond_\mu(\Psi_1 \sqcup \Psi_2) \vdash \diamond_\mu(\Psi_1) \wedge \diamond_\mu(\Psi_2)$

**(UC7)** $\diamond_{\mu_1}(\Psi) \wedge \mu_2 \vdash \diamond_{\mu_1 \wedge \mu_2}(\Psi)$

**(UC8)** *Si* $\Psi$ *est un profil complet et si* $\diamond_{\mu_1}(\Psi) \wedge \mu_2 \not\vdash \bot$, *alors* $\diamond_{\mu_1 \wedge \mu_2}(\Psi) \vdash \diamond_{\mu_1}(\Psi) \wedge \mu_2$

**(UC9)** $\diamond_\mu(\Psi \sqcup \{\varphi \vee \varphi'\}) \equiv \diamond_\mu(\Psi \sqcup \{\varphi\}) \vee \diamond_\mu(\Psi \sqcup \{\varphi'\})$

Certains postulats (UC) sont exactement les mêmes que le postulats (IC), tout comme certains postulats (U) pour la mise à jour, sont exactement les mêmes que les postulats (R) pour la révision.

Plus exactement (UC0), (UC3), (UC5) et (UC7) sont exactement les mêmes que les postulats (IC) correspondants. Donc la particularité des opérateurs de confluence doit être recherchée dans les postulats (UC1), (UC2), (UC6), (UC8) et (UC9). (UC4), (UC6) et (UC8) sont proches des postulats (IC) correspondants, mais ne sont vraies que pour des profils complets. (UC2) est très proche du postulat (U2) de la mise à jour. Mais notons que dans le cas de profil complet, l'hypothèse de (UC2) équivaut à demander la consistance avec les contraintes, i.e. l'hypothèse de (IC2). Les postulats (UC8) et (UC9) expriment les plus grandes différences avec les postulats pour la fusion contrainte, et correspondent également aux différences principales entre la révision et la mise à jour. (UC9) est le postulat le plus important, qui défini les opérateurs de confluences comme des opérateurs de fusion ponctuelle, tout comme (U8) défini les opérateurs de mise à jour comme des opérateurs de révision ponctuelle. Cela sera exprimé plus formellement dans la prochaine section (Lemme 1).

## 5 Théorème de représentation pour les opérateurs de confluence

Afin de définir le théorème de représentation pour les opérateurs de confluence, nous devons d'abord trouver un moyen de "ponctualiser" le problème. Pour la mise à jour cela est fait en examinant chaque modèle de la base séparément, au lieu de considérer la base (l'ensemble des modèles) comme un tout (comme le fait la révision). Donc, pour le problème de l'agrégation, nous devons trouver la notion de vue ponctuelle d'un profil. C'est ce que nous appelons un état.

**Définition 9** *Un multi-ensemble d'interpretations est appelé un* état. *On notera les états* $e$. *Si* $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ *est un profil et* $e = \{\omega_1, \ldots, \omega_n\}$ *est un état tel que* $\omega_i \models \varphi_i$ *pour tout* $i$, *on dit que* $e$ *est un état du profil* $\Psi$, *ou que l'état* $e$ *satisfait le profil* $\Psi$, *ce qui sera noté* $e \models \Psi$.

*Si* $e = \{\omega_1, \ldots, \omega_n\}$ *est un état, on définit le profil* $\Psi_e$ *par* $\Psi_e = \{\varphi_{\{\omega_1\}}, \ldots, \varphi_{\{\omega_n\}}\}$.

La notion d'état est une notion intéressante. Si on considère chaque base comme le point de vue actuel d'un agent (ou ses buts), qui peuvent éventuellement être affinés dans le futur, alors les états sont tous les points de départ possibles pour des négociations.

Les états sont les points d'intérêt pour les opérateurs de confluence (comme les interprétations pour la mise à jour), comme souligné dans le lemme suivant :

**Lemme 1** *Si* $\diamond$ *satisfait (UC3) et (UC9) alors* $\diamond$ *satisfait la propriété suivante*

$$\diamond_\mu(\Psi) \equiv \bigvee_{e \models \Psi} \diamond_\mu(\Psi_e)$$

Si on définit la satisfaction de profil par $\Psi \vdash \Psi'$ si tout état de $\Psi$ est un état de $\Psi'$, le lemme précédent a comme corollaire :

**Corollaire 1** *Si* $\diamond$ *est un opérateur de confluence, alors il est monotone pour les profils, i.e. si* $\Psi \vdash \Psi'$ *alors* $\diamond_\mu(\Psi) \vdash \diamond_\mu(\Psi')$

Cette propriété de monotonie, qui n'est pas vérifiée par les opérateurs de fusion, montre une importante différence entre les opérateurs de confluence et de fusion. Remarquons qu'il y a une propriété similaire de monotonie pour la mise à jour.

Tout comme les assignements fidèles doivent être "localisés" aux interprétations pour la mise à jour, les assignements syncrétiques doivent être "localisés" aux états pour la confluence.

**Définition 10** *Un* assignment distribué *est une fonction qui associe à chaque état* $e$ *un préordre total* $\leq_e$ *sur les interprétations tel que :*

1. $\omega <_{\{\omega,\ldots,\omega\}} \omega'$ *si* $\omega' \neq \omega$
2. $\omega \simeq_{\{\omega,\omega'\}} \omega'$
3. *Si* $\omega \leq_{e_1} \omega'$ *et* $\omega \leq_{e_2} \omega'$, *alors* $\omega \leq_{e_1 \sqcup e_2} \omega'$
4. *Si* $\omega <_{e_1} \omega'$ *et* $\omega \leq_{e_2} \omega'$, *alors* $\omega <_{e_1 \sqcup e_2} \omega'$

Nous pouvons à présent donner le résultat principal de cet article, le théorème de représentation pour les opérateurs de confluence.

**Théorème 7** *Un opérateur $\diamond$ est un opérateur de confluence si et seulement si il existe un assignement distribué qui associe à chaque état $e$ un pré-ordre total $\leq_e$ tel que*

$$mod(\diamond_\mu(\Psi)) = \bigcup_{e \models \Psi} \min(mod(\mu), \leq_e)$$

On peut noter que ce théorème est également vrai si on retire respectivement le postulat (UC4) et la condition 2 des assignements distribués.

## 6 Confluence, Mise à jour et Fusion

Montrons à présent que la mise à jour est un cas particulier de confluence, tout comme la révision est un cas particulier de fusion.

**Théorème 8** *Si $\diamond$ est un opérateur de confluence (i.e. il satisfait (UC0-UC9)), alors l'opérateur $\diamond$, défini par $\varphi \diamond \mu = \diamond_\mu(\varphi)$, est un opérateur de mise à jour (i.e. il satisfait (U1-U9)).*

En ce qui concerne les opérateurs de fusion, on peut voir facilement que la restriction d'un assignement syncrétique à des profils complets donne un assignement distribué. On peut donc obtenir le résultat suivant (qui correspond au Théorème 5) :

**Théorème 9** *Si $\triangle$ est un opérateur de fusion contrainte (i.e. il satisfait (IC0-IC8)) alors l'opérateur $\diamond$ défini par*

$$\diamond_\mu(\Psi) = \bigvee_{e \models \Psi} \triangle_\mu(\Psi_e)$$

*est un opérateur de confluence (i.e. il satisfait (UC0-UC9)).*

*De plus, pour tout opérateur de confluence $\diamond$, il existe un opérateur de fusion $\triangle$ qui lui correspond via l'équation précédente.*

Il est intéressant de noter que ce théorème montre que chaque opérateur de fusion peut être utilisé pour définir un opérateur de confluence, et montre également en quoi on peut considérer les opérateurs de confluence comme des opérateurs de fusion ponctuelle.

A l'inverse du théorème 5, où la seconde partie du théorème est une conséquence directe des théorèmes de représentation, on a besoin ici de construire un assignement syncrétique qui étend l'assignement distribué représentant l'opérateur de confluence. Pour ce faire on peut utiliser la construction suivante : chaque pré-ordre $\leq_e$ définit naturellement une fonction de rang $r_e$ sur les entiers. On définit alors $\omega \leq_\Psi \omega'$ si et seulement si $\sum_{e \models \Psi} r_e(\omega) \leq \sum_{e \models \Psi} r_e(\omega')$

Comme corollaire du théorème de représentation on obtient :

**Corollaire 2** *Si $\diamond$ est un opérateur de confluence alors la propriété suivante est vérifiée :*

*Si $\bigwedge \Psi \vdash \mu$ et $\Psi$ est consistant alors $\bigwedge \Psi \wedge \mu \vdash \diamond_\mu(\Psi)$*

*Mais, à l'inverse des opérateurs de fusion, on n'a généralement pas $\diamond_\mu(\Psi) \vdash \bigwedge \Psi \wedge \mu$.*

Notons que cette "moitié de (IC2)" est similaire à la "moitié de (R2)" satisfaite par les opérateurs de mise à jour.

Ce corollaire est intéressant puisqu'il souligne une importante différence entre les opérateurs de fusion et les opérateurs de confluence. Si toutes les bases sont d'accord (i.e. leur conjonction est consistante), alors un opérateur de fusion donne comme résultat exactement cette conjonction. Alors qu'avec un opérateur de confluence, cette conjonction ne sera qu'une partie du résultat.

Cela est utile lorsque les bases ne représentent pas une information connue et figée, mais une information incertaine/imparfaite à propos des bus courants ou futurs de l'agent.

## 7 Exemple

Dans cette section nous donnerons des exemples d'opérateurs de confluence et nous illustrerons le comportement de ces sur un exemple.

Il est possible de définir des opérateurs de confluence, comme les opérateurs de fusion, en utilisant une distance et une fonction d'agrégation.

| $\mathcal{W}$ | 1111 | 1110 | 0000 | 0110 | $e_1$ | | $e_2$ | | $e_3$ | | $e_4$ | | $e_5$ | | $e_6$ | | $\diamond_\mu^{d,\Sigma}$ | $\diamond_\mu^{d,\text{Gmax}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | $\Sigma$ | Gmax | $\Sigma$ | Gmax | $\Sigma$ | Gmax | $\Sigma$ | Gmax | $\Sigma$ | Gmax | $\Sigma$ | Gmax | | |
| 0000 | 4 | 3 | 0 | 2 | 11 | 4430 | 10 | 4420 | 10 | 4330 | 9 | 4320 | 9 | 3330 | 8 | 3320 | | |
| 0001 | 3 | 4 | 1 | 3 | 11 | 4331 | 10 | 3331 | 12 | 4431 | 11 | 4331 | 13 | 4441 | 12 | 4431 | | |
| 0010 | 3 | 2 | 1 | 1 | 9 | 3321 | 8 | 3311 | 8 | 3221 | 7 | **3211** | **7** | **2221** | **6** | **2211** | × | × |
| 0011 | 2 | 3 | 2 | 2 | 9 | 3222 | 8 | **2222** | 10 | 3322 | 9 | 3222 | 11 | 3332 | 10 | 3322 | | × |
| 0100 | 3 | 2 | 1 | 1 | 9 | 3321 | 8 | 3311 | 8 | **3221** | **7** | **3211** | **7** | **2221** | **6** | **2211** | × | × |
| 0101 | 2 | 3 | 2 | 2 | 9 | 3222 | 8 | **2222** | 10 | 3322 | 9 | 3222 | 11 | 3332 | 10 | 3322 | | × |
| 0110 | 2 | 1 | 2 | 0 | 7 | 2221 | 6 | 2220 | 6 | 2211 | 5 | 2210 | 5 | 2111 | 4 | 2110 | | |
| 0111 | 1 | 2 | 3 | 1 | 7 | **3211** | **6** | 3111 | 8 | **3221** | **7** | **3211** | 9 | 3222 | 8 | 3221 | × | × |
| 1000 | 3 | 2 | 1 | 3 | 9 | 3321 | 10 | 3331 | 8 | **3221** | 9 | 3321 | **7** | **2221** | 8 | 3221 | × | × |
| 1001 | 2 | 3 | 2 | 4 | 9 | 3222 | 10 | 4222 | 10 | 3322 | 11 | 4322 | 11 | 3332 | 12 | 4332 | | |
| 1010 | 2 | 1 | 2 | 2 | 7 | 2221 | 8 | 2222 | 6 | 2211 | 7 | 2221 | 5 | 2111 | 6 | 2211 | | |
| 1011 | 1 | 2 | 3 | 3 | 7 | **3211** | 8 | 3311 | 8 | **3221** | 9 | 3321 | 9 | 3222 | 10 | 3322 | | × |
| 1100 | 2 | 1 | 2 | 2 | 7 | 2221 | 8 | 2222 | 6 | 2211 | 7 | 2221 | 5 | 2111 | 6 | 2211 | | |
| 1101 | 1 | 2 | 3 | 3 | 7 | **3211** | 8 | 3311 | 8 | **3221** | 9 | 3321 | 9 | 3222 | 10 | 3322 | | × |
| 1110 | 1 | 0 | 3 | 1 | 5 | 3110 | 6 | 3111 | 4 | 3100 | 5 | 3110 | 3 | 3000 | 4 | 3100 | | |
| 1111 | 0 | 1 | 4 | 2 | **5** | 4100 | **6** | 4200 | **6** | 4110 | **7** | 4210 | **7** | 4111 | 8 | 4211 | × | |

TAB. 1 – Calculs de l'exemple 2

**Définition 11** *Une* pseudo-distance *entre interprétations est une fonction* $d : \mathcal{W} \times \mathcal{W} \mapsto \mathbb{R}^+$ *telle que pour tout* $\omega$, $\omega' \in \mathcal{W}$ : $d(\omega, \omega') = d(\omega', \omega)$, *et* $d(\omega, \omega') = 0$ *si et seulement si* $\omega = \omega'$.

Une distance largement utilisée est la distance de Dalal [3], notée $d_H$, qui est la distance de Hamming entre les interprétations (le nombre de variables propositionnelles sur lesquelles les deux interprétations diffèrent).

**Définition 12** *Une* fonction d'agrégation $f$ *est une fonction qui associe un réel positif à tout tuple fini de réels positifs tel que pour tout* $x_1, \ldots, x_n, x, y \in \mathbb{R}^+$ :
- *si* $x \leq y$, *alors* $f(x_1, \ldots, x, \ldots, x_n) \leq f(x_1, \ldots, y, \ldots, x_n)$ *(monotonie)*
- $f(x_1, \ldots, x_n) = 0$ *si et seulement si* $x_1 = \ldots = x_n = 0$ *(minimalité)*
- $f(x) = x$ *(identité)*

Des fonctions d'agrégation raisonnables sont par exemple le $\max$, la somme, ou le leximax ($Gmax$).

**Définition 13** *Soit une pseudo-distance entre interprétations* $d$ *et une fonction d'agrégation* $f$. *L'opérateur* $\diamond_\mu^{d,f}(\Psi)$ *est défini par :*

$$mod(\diamond_\mu^{d,f}(\Psi)) = \bigcup_{e \models \Psi} \min(mod(\mu), \leq_e)$$

*où le pré-ordre* $\leq_e$ *sur* $\mathcal{W}$ *induit par* $e$ *est défini par :*
- $\omega \leq_e \omega'$ *si et seulement si* $d(\omega, e) \leq d(\omega', e)$, *où*
- $d(\omega, e) = f(d(\omega, \omega_1) \ldots, d(\omega, \omega_n))$ *avec* $e = \{\omega_1, \ldots, \omega_n\}$.

Il est facile de vérifier qu'avec les fonctions d'agrégation usuelles, on obtient bien des opérateurs de confluence.

**Proposition 1** *Soit une pseudo-distace* $d$, $\diamond_\mu^{d,\Sigma}(\Psi)$ *et* $\diamond_\mu^{d,Gmax}(\Psi)$ *sont des opérateurs de confluence (i.e. ils satisfont (UC0)-(UC9)).*

**Exemple 2** *Considérons le profil* $\Psi = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ *et les contraintes d'intégrité* $\mu$ *definis sur un langage propositionnel basé sur quatre variables :* $mod(\mu) = \mathcal{W} \setminus \{0110, 1010, 1100, 1110\}$, $mod(\varphi_1) = mod(\varphi_2) = \{1111, 1110\}$, $mod(\varphi_3) = \{0000\}$, $mod(\varphi_4) = \{1110, 0110\}$.

*Les calculs sont résumés dans le tableau 1. Les lignes grisées correspondent aux interprétations rejetées par les contraintes d'intégrité. Le résultat doit donc être trouvé parmi les interprétations non grisées. Les états qui satisfont le profil sont les suivants :*

$e_1 = \{1111, 1111, 0000, 1110\}$,
$e_2 = \{1111, 1111, 0000, 0110\}$,
$e_3 = \{1111, 1110, 0000, 1110\}$,
$e_4 = \{1110, 1111, 0000, 0110\}$,
$e_5 = \{1110, 1110, 0000, 1110\}$,
$e_6 = \{1110, 1110, 0000, 0110\}$.

*Pour chaque état le tableau donne la distance entre l'interprétation et cet état pour la fonction d'agrégation $\Sigma$, et pour la fonction d'agrégation $Gmax$. On peut alors chercher quelles sont les meilleures interprétations pour chaque état.*

*Par exemple pour* $\diamond_\mu^{d,\Sigma}(\Psi)$, $e_1$ *sélectionne l'interprétation* $1111$, $e_2$ *sélectionne* $0111$ *et* $1111$, *etc. Donc, en prenant l'union des interprétations sélectionnées par chaque état, le résultat est* $mod(\diamond_\mu^{d,\Sigma}(\Psi)) = \{0010, 0100, 0111, 1000, 1111\}$.

*De la même manière on obtient* $mod(\diamond_\mu^{d,Gmax}(\Psi)) = \{0100, 0011, 0010, 0101, 0111, 1000, 1011, 1101\}$.

## 8 Conclusion

Nous avons proposé dans cet article une nouvelle famille d'opérateurs de changement : les opérateurs de confluence, qui sont des opérateurs de fusion ponctuelle, tout comme la mise à jour peut être considérée comme une révision ponctuelle.

Nous avons proposé une définition axiomatique de cette famille, un théorème de représentation en termes de pré-ordres sur les interprétations, et donné des exemples d'opérateurs.

Dans cet article nous avons défini les opérateurs de confluence comme généralisation à plusieurs bases des opérateurs de mise à jour totale (dont la contrepartie sémantique correspond à des pré-ordres totaux). Une perspective de ce travail est de tenter d'étendre ces résultats aux opérateurs de mise à jour partielle (correspondant à des pré-ordres partiels).

Comme le suggère l'exemple 1, ces opérateurs peuvent être utiles pour agréger les buts d'un groupe d'agents. Ils semblent moins adéquats pour agréger des croyances, où la minimisation globale réalisée par les opérateurs de fusion est plus appropriée pour trouver les interprétations les plus plausibles. Cette distinction entre agrégation de croyances et de buts est une intéressante perspective.

### Remerciements

L'idée développée dans ce travail provient de discussions lors des séminaires de Dagstuhl de 2005 et 2007 ($\#5321$ and $\#7351$) *"Belief Change in Rational Agents"*. Les auteurs voudraient remercier l'institution "Schoss Dagstuhl", et les participants aux séminaires. En particulier Andreas Herzig pour la question initiale *"Si la fusion peut être vue comme une généralisation de la révision, qu'est-ce que la généralisation de la mise à jour ?"*.

Le premier auteur a bénéficié du support du projet ANR PHAC (ANR-05-BLAN-0384).

Le second auteur a été partiellement financé par une bourse de recherche de la Mairie de Paris et par le projet CDCHT-ULA N° C-1451-07-05-A. Une partie de ce travail a été réalisé alors que le second auteur était professeur invité au CRIL (CNRS UMR 8188), puis chercheur invité au département TSI de Telecom ParisTech (CNRS UMR 5141 LTCI). Il remercie ces institutions pour les excellentes conditions de travail.

## Références

[1] C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change : Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50 :510–530, 1985.

[2] R. Booth. Social contraction and belief negotiation. In *Proceedings of the Eighth Conference on Principles of Knowledge Representation and Reasoning (KR'02)*, pages 374–384, 2002.

[3] M. Dalal. Investigations into a theory of knowledge base revision : preliminary report. In *Proceedings of the American National Conference on Artificial Intelligence (AAAI'88)*, pages 475–479, 1988.

[4] F. Dupin de Saint-Cyr and J. Lang. Belief extrapolation (or how to reason about observations and unpredicted change). *Proceedings of the Eighth Conference on Principles of Knowledge Representation and Reasoning (KR'02)*, pages 497–508, 2002.

[5] P. Gärdenfors. *Knowledge in flux*. MIT Press, 1988.

[6] P. Gärdenfors, editor. *Belief Revision*. Cambridge University Press, 1992.

[7] A. Grove. Two modellings for theory change. *Journal of Philosophical Logic*, 17(157-180), 1988.

[8] A. Herzig and O. Rifi. Update operations : a review. In *Proceedings of the Thirteenth*

*European Conference on Artificial Intelligence (ECAI'98)*, pages 13–17, 1998.

[9] H. Katsuno and A. O. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 387–394, 1991.

[10] H. Katsuno and A. O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52 :263–294, 1991.

[11] H. Katsuno and A. O. Mendelzon. On the difference between updating a knowledge base and revising it. In P. Gärdenfors, editor, *Belief Revision*. Cambridge University Press, 1992.

[12] S. Konieczny. Belief base merging as a game. *Journal of Applied Non-Classical Logics*, 14(3) :275–294, 2004.

[13] S. Konieczny, J. Lang, and P. Marquis. $DA^2$ merging operators. *Artificial Intelligence*, 157(1-2) :49–79, 2004.

[14] S. Konieczny and R. Pino Pérez. Merging information under constraints : a logical framework. *Journal of Logic and Computation*, 12(5) :773–808, 2002.

[15] J. Lang. Belief update revisited. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 2517–2522, 2007.

[16] J. Lobo and C. Uzcátegui. Abductive change operators. *Fundamenta Informaticae*, 27(4) :385–411, 1996.

[17] T. Meyer, N. Foo, D. Zhang, and R. Kwok. Logical foundations of negotiation : Outcome, concession and adaptation. In *Proceedings of the American National Conference on Artificial Intelligence (AAAI'04)*, pages 293–298, 2004.

[18] T. Meyer, N. Foo, D. Zhang, and R. Kwok. Logical foundations of negotiation : Strategies and preferences. In *Proceedings of the Ninth Conference on Principles of Knowledge Representation and Reasoning (KR'04)*, pages 311–318, 2004.

[19] P. Z. Revesz. On the semantics of arbitration. *International Journal of Algebra and Computation*, 7(2) :133–160, 1997.

[20] D. Zhang, N. Foo, T. Meyer, and R. Kwok. Negotiation as mutual belief revision. In *Proceedings of the American National Conference on Artificial Intelligence (AAAI'04)*, pages 317–322, 2004.

# Sur la complexité de l'apprentissage de préférences séparables de type *ceteris paribus*

Jérôme Lang[1] Jérôme Mengin[2]
lang@lamsade.dauphine.fr mengin@irit.fr

[1] LAMSADE
Université Paris-Dauphine
75775 Paris Cedex 16

[2] IRIT
Université de Toulouse
31062 Toulouse Cedex

**Résumé :**
Nous nous intéressons à l'apprentissage de relations de préférences sur des domaines multi-attributs (ou combinatoires), en faisant une hypothèse très simple d'indépendance entre les attributs: nous supposons que les préférences sur les différents attributs sont séparables. Etant donné un ensemble d'exemples, consistant chacun en une comparaison entre deux alternatives, nous voulons produire un CP-net séparable, consistant en une collection de préférences locales, une par attribut, qui soit compatible avec les exemples. Nous considérons trois formes de compatibilité entre un CP-net et un ensemble d'exemples ; pour chacune nous donnons une caractérisation ainsi que des résultats de complexité.

**Mots-clés :** Apprentissage de préférences ; CP-nets

**Abstract:**
We address the problem of learning preference relations on multi-attribute (or combinatorial) domains. We do so by making a very simple hypothesis about the dependence structure between attributes that the preference relation enjoys, namely separability (no preferential dependencies between attributes). Given a set of examples consisting of comparisons between alternatives, we want to output a separable CP-net, consisting of local preferences on each of the attributes, that fits the examples. We consider three forms of compatibility between a CP-net and a set of examples, and for each of them we give useful characterizations as well as complexity results.

**Keywords:** Preference learning ; CP-nets

## 1 Introduction

Pour de nombreuses applications, comme le commerce électronique, il est important d'apprendre les préférences d'un utilisateur sur un ensemble d'alternatives (au sens anglais du terme) qui a une structure combinatoire (ou multi-attributs) : chaque alternative est un n-uplet de valeurs, une pour chaque élément d'un ensemble donné d'attributs. Supposons par exemple qu'on observe une utilisatrice faire les choix suivants entre des billets d'avions : (a) elle préfère un vol KLM de nuit atterrissant à Heathrow à un vol Aeroflot de jour atterrissant à Gatwick ; (b) elle préfère un vol Aeroflot de nuit atterrissant à Heathrow à un vol KLM de jour atterrissant à Heathrom ; et (c) elle préfère un vol KLM de jour atterrissant à Heathrow à un vol KLM de nuit atterrissant à Heathrow. Une explication intuitvement correcte (parmi d'autres) de ses choix est qu'elle préfère (inconditionnellement) Aeroflot à KLM, les vols de jour aux vols de nuit (encore uncondtionnellement), et Heathrow à Gatwick, toujours unconditionnellement. Cette explication a la propriété cruciale de *séparabilité* : les préférences sur les valeurs de chaque attribut sont indépendantes des valeurs des autres attributs. Cette explication permet de prédire qu'elle préfère un vol Aeroflot de jour atterrissant à Heathrow à un vol Aeroflot de jour atterrissant à tout autre vol, et un vol Aeroflot de jour atterrissant à Heathrow à un vol Aeroflot de jour atterrissant à Gatwick (mais cela ne permet pas de prédire si elle préfère un vol Aeroflot de jour atterrissant à Gatwick ou un vol Aeroflot de nuit atterrissant à Heathrow).

Si on observe de plus que l'utilisatrice préfère un vol Aeroflot de nuit atterrissant à Gatwick plutôt qu'un vol Aeroflot de jour atterrissant à Gatwick, l'explication précédente n'est plus valable ; et plus généralement, aucune explication où ses préférences seraient séparables n'est valable. Une explication possible (parmi d'autres) pourrait être qu'elle préfère Aeroflot à KLM, Heathrow à Gatwick, les vols de jour aux vols nocturnes avec KLM et *vice-versa* avec Aeroflot.

Cela soulève plusieurs questions intéressantes : comment peut-on décider s'il est possible que les préférences de l'utilisatrice sont séparables ? Si oui, quelles sont les préférences locales sur chaque variable ? Sinon, peut-on trouver une relation de préférence séparable « maximalement compatible » avec l'ensemble d'exemples ? Cet article essaye de répondre à ces questions (mais nous n'abordons pas la recherche de relations de préférence *non séparables* compatibles avec un

ensemble d'exemples).

Dans cet article, nous nous concentrons sur l'*apprentissage passif* : le système a en entrée un ensemble de préférences entre alternatives, que l'utilisatrice a déjà exprimées (sans nécessairement qu'on lui ait explicitement demandé de le faire), et on veut raisonner sur ces préférences afin de prédire ses choix sur d'autres alternatives. Cela est différent de l'*élicitation* de préférences (voir par exemple [Chen et Pu, 2004; Braziunas et Boutilier, 2006]), qui consiste à interagir avec l'utilisatrice en lui soumettant des requêtes spécifiques, ou en lui suggérant de nouvelles alternatives, afin de la conduire à exprimer ses préférences sur les valeurs de certains attributs [Viappiani *et al.*, 2006], jusqu'à ce qu'elle ait trouvé l'objet qu'elle préfère ou quitté le système.

Les préférences sur des domaines combinatoires ont été étudiées en détail en théorie de la décision multi-critères (notamment depuis Keeney et Raiffa 1976) et en intelligence artificielle. Les travaux en théroie de la décision multi-critères ont surtout porté sur la *modélisation* des préférences, alors qu'en intelligence artificielle les chercheurs ont plutôt cherché à définir des langages de *représentation* de préférences aussi succincts que possibles et à proposer des algorithmes permettant de trouver les alternatives optimales aussi rapidement que possible.

Les modèles et langages pour les préférences peuvent être différenciés d'abord selon qu'ils manipulent des préférences *ordinales* (des relations d'ordre entre alternatives) ou des préférences *numériques* (par exemple les fonctions d'utilité). Nous nous intéressons ici aux préférences ordinales. Elles ont l'avantage d'être souvent plus facile à obtenir d'un utilisateur.

Alors qu'un certain nombre de travaux récents ont porté sur l'apprentissage de fonctions d'ordre et de préférences sur des paires dans le cas de domaines non combinatoires (voir par exemple Cohen *et al.* 1999, Hüllermeier *et al.* 1999), et que l'apprentissage et l'élicitation de fonctions d'utilité sur des domaines multi-critères ont suscité quelques travaux (Ha et Haddawy 1997 ; Gonzales et Perny 2004,entre autres), il y a eu moins de tentatives d'apprentissage de préférences à partir de comparaisons ordinales, à savoir : (a) l'apprentissage de préférences lexicographiques [Schmitt et Martignon, 2006; Dombi *et al.*, 2007; Yaman *et al.*, 2008] ; (b) quelques travaux sur l'apprentissage de CP-nets [Athienitou et Dimopoulos, 2007; Koriche et Zanuttini, 2009], (c) des préférences *ceteris paribus* plus générales [Sachdev, 2007] ou (d) des fonctions d'utilité apprises à partir de comparaisons ordinales entre formules logiques [Domshlak et Joachims, 2007]. Nous reviendrons sur ces approches en Section 6.

Pour (a) et (b), la relation de préférence sur un domaine multi-critères est construite à partir de relations de préférences « locales » sur les domaines de chaque attribut ; ces ordres locaux sont ensuite combinés soit de manière lexicographique pour (a), soit, pour (b), en faisant l'hypothèse que les comparaisons locales peuvent être généralisées à des comparaisons entre vecteurs de valeurs quand tous les autres attributs ont la même valeur (*ceteris paribus*). Les CP-nets permettent d'exprimer, à l'aide d'un graphe, des dépendances entres les variables, et ensuite de représenter des préférences locales sur les valeurs de chaque attribut, en fonction des valeurs d'autres attributs. Afin d'apprendre des CP-nets de structure quelconque, étant donné un ensemble d'exemples, il faudrait identifier la structure des dépendances entre attributs *et* les tables de préférences conditionnelles locales. On pourrait alors chercher le plus petit CP-net compatible avec les exemples, ou le meilleur compromis entre simplicité et compatibilité avec les exemples. Toutefois, dans cet article, nous nous restreignons à un problème plus simple en ne considérant que la classe des CP-nets les plus simples : les CP-nets séparables, dont le graphe des dépendances entre variables est vide. Il y a plusieurs raisons justifiant ce choix. D'abord, les préférences séparables forment une classe importante de relations de préférences sur des domaines multi-critères, par exemple en économie ou en choix social (voir par exemple [Bradley *et al.*, 2005]), et l'apprentissage de préférences séparables mérite une attention particulière. Deuxièmement, comme nous le verrons, le problème de l'apprentissage de préférences séparables a une complexité de calcul élevée – plus qu'on pourrait le penser *a priori* étant donné qu'on peut *raisonner* en temps linéaire sur des CP-nets séparables. Troisièmement, la mise au point de méthodes pour l'apprentissage de préférences non séparables est une question importante mais aussi très délicate, à laquelle on ne peut sûrement pas répondre en un seul article. Se concentrer d'abord sur des préférences séparables est une étape nécessaire et non triviale vers l'apprentissage de préférences plus complexes. Enfin, il est bien connu qu'une structure simple peut avoir de meilleures propriétés de générali-

sation qu'une structure plus compliquée qui serait plus en adéquation avec les exemples.

La section 2 rappelle des notions élémentaires sur les préférences sur des domaines multi-attributs, et sur les CP-nets (séparables). Nous nous intéressons ensuite successivement à trois formes de compatibilité entre un CP-net et un ensemble d'exemples : la *compatibilité faible* en section 3, la *compatibilité forte* en section 4, et la *compatibilité implicative* en section 5. Nous prouvons qu'alors qu'il est possible de tester en temps polynomial s'il existe un CP-net séparable qui implique un ensemble d'exemples, déterminer s'il en existe un qui soit faiblement (ou fortement) compatible avec les exemples est NP-complet. La section 6 discute des liens avec des travaux connexes, et la section 7 conclut l'article.

## 2 Le problème

On veut apprendre un ordre sur des objets, ou *alternatives*, définis par leurs valeurs pour les attributs d'un ensemble $\mathcal{V} = \{X_1, \ldots, X_n\}$. Chaque attribut $X_i$ a un domaine fini $D_i$ de valeurs possibles. On note $D = D_1 \times \ldots \times D_n$ l'ensemble des alternatives possibles. Si $\vec{x} \in D$, alors $(\vec{x})_i$ représente la valeur de $\vec{x}$ pour le $i$-ème attribut $X_i$. Un attribut $X_i$ est *binaire* si $D_i$ a deux éléments, que nous notons par convention $x_i$ et $\overline{x_i}$.

Une *relation de préférence (stricte)* $\succ$ est un ordre partiel strict sur $D$, c'est-à-dire une relation binaire irréflexive et transitive (et donc asymétrique). Si de plus $\succ$ est totale (si $\vec{x} \neq \vec{y}$ alors $\vec{x} \succ \vec{y}$ ou $\vec{y} \succ \vec{x}$) alors $\succ$ est une *relation de préférence linéaire*, ou plus simplement un *ordre linéaire*. On peut remarquer que si on considère la relation transitive $\succ$ comme un graphe sur l'ensemble de sommets $D$ (avec un arc de $\vec{x}$ à $\vec{y}$ si $x \succ y$), alors $\succ$ est irréflexive si et seulement si le graphe est acyclique.

Supposons maintenant qu'on a un ensemble fini d'*exemples* $\mathcal{E}$, où chaque exemple est une paire d'alternatives distinctes $(\vec{x}, \vec{y})$ (que nous noterons souvent $\vec{x} \succ \vec{y}$) telle que $\vec{x}$ est préféré à $\vec{y}$. Dans un système de recommandation, ces exemples pourraient avoir été enregistrés durant l'interaction d'une utilisatrice avec le système. Idéalement, on voudrait apprendre à ordonner toute paire non ordonnée d'alternatives distinctes $\{\vec{x}, \vec{y}\}$. Notre cible est donc une relation de préférence linéaire. [1]

Cet article étudie l'apprentissage de *structures de préférence ceteris paribus séparables*, ou *structure SCP* dans la suite. Les structures SCP sont des cas particuliers des *réseaux de préférences conditionnelles*, ou CP-nets, de [Boutilier *et al.*, 2004], lorsque le graphe de dépendance entre les attributs est vide, c'est-à-dire lorsque chaque attribut est préférentiellement indépendant de tous les autres attributs (les CP-nets permettent de faire dépendre les préférences sur les valeurs d'un attribut des valeurs d'autres attributs). Une structure SCP sur $\mathcal{V}$ est une collection $\mathcal{N}$ d'ordres linéaires $\succ_i$, un sur chaque $D_i$. Un *"swap"* est une paire d'alternatives $\{\vec{x}, \vec{y}\}$ qui diffèrent seulement par la valeur d'un attribut. Une structure SCP $\mathcal{N}$ induit une relation de préférence $\succ_{\mathcal{N}}$ de la manière suivante : on dit d'abord qu'un ordre linéaire $D$ satisfait $\mathcal{N}$ si pour tout attribut $X_i$ et pour tous $\vec{z}, \vec{z'}$ tels que $\vec{z}$ et $\vec{z'}$ coïncident sur tous les attributs excepté $X_i$, $\vec{z} \succ \vec{z'}$ si et seulement si $(\vec{z})_i \succ_i (\vec{z'})_i$. Ensuite, $\succ_{\mathcal{N}}$ est l'intersection de toutes les relations de préférences qui satisfont $\mathcal{N}$. On remarque que $\succ_{\mathcal{N}}$ est un ordre partiel sur $D$. Tout ordre linéaire $\succ$ qui satisfait une structure SCP est *séparable*. La séparabilité signifie que les attributs sont mutuellement indépendants en ce qui concerne les préférences. On peut remarquer que si $\mathcal{N}$ est une structure SCP, alors $\succ_{\mathcal{N}}$ est irréflexive, et peut donc être étendue à au moins un ordre linéaire sur $D$. [Boutilier *et al.* 2004, Th. 1].

Pour simplifier la présentation, on suppose dans la suite (sauf lorsqu'on indique explicitement le contraire) que tous les attributs sont binaires ; les résultats peuvent toutefois être généralisés à des attributs non binaires, comme on le montre à la fin de la section 3. Dans le cas d'attributs binaires, pour tout paire d'alternatives $(\vec{x}, \vec{y})$, on définit

$$\begin{aligned} \mathit{Diff}(\vec{x}, \vec{y}) &= \{x_i \mid (\vec{x})_i = x_i \text{ et } (\vec{y})_i = \overline{x_i}\} \\ &\cup \{\overline{x_i} \mid (\vec{x})_i = \overline{x_i} \text{ et } (\vec{y})_i = x_i\}. \end{aligned}$$

On a alors la caractérisation suivante de $\succ_{\mathcal{N}}$ (un corollaire des Théorèmes 7 et 8 de Boutilier *et al.*, 2004) :

[1] La méthode et les résultats présentés ici peuvent être facilement généralisés au problème de l'apprentissage de relations de préférences non strictes, lorsqu'on autorise l'indifférence entre deux alternatives. Nous nous restreignons à des relations strictes ici car la présentation est plus simple.

**Lemme 1** *Soit $\mathcal{N}$ une structure SCP sur des variables binaires, et soit $\vec{y} \neq \vec{x}$. Alors $\vec{x} \succ_{\mathcal{N}} \vec{y}$ si et seulement si $\mathcal{N}$ contient $x_i \succ \overline{x_i}$ pour tout $x_i \in Diff(\vec{x}, \vec{y})$ et $\overline{x_i} \succ x_i$ pour tout $\overline{x_i} \in Diff(\vec{x}, \vec{y})$.*

**Exemple 1** *Soient trois attributs binaires $A$, $B$ et $C$, de domaines respectifs $\{a, \overline{a}\}$, $\{b, \overline{b}\}$ et $\{c, \overline{c}\}$, et soit $\mathcal{N} = \{a \succ \overline{a}, b \succ \overline{b}, \overline{c} \succ c\}$. On a $ab\overline{c} \succ_{\mathcal{N}} \overline{a}\overline{b}c$, alors que $a\overline{b}c$ et $\overline{a}bc$ sont incomparables pour $\succ_{\mathcal{N}}$.*

On a écrit plus haut que la cible de notre processus d'apprentissage devrait idéalement être un ordre linéaire sur $D$, mais nous venons de voir qu'une structure SCP $\mathcal{N}$ ne correspond en général pas à un tel ordre, puisque la relation de préférence $\succ_{\mathcal{N}}$ induite n'est en général pas complète. En fait, $\succ_{\mathcal{N}}$ peut être vue comme l'ensemble de toutes ses complétions, c'est-à-dire qu'une structure SCP exprime un *ensemble de relations de préférences linéaires*.

Si un exemple $(\vec{x}, \vec{y})$ est un swap, alors $\vec{x} \succ_{\mathcal{N}} \vec{y}$ ou $\vec{y} \succ_{\mathcal{N}} \vec{x}$ : dans ce cas, on voudrait évidemment que $\mathcal{N}$ soit en accord avec l'exemple en ce qui concerne $\{\vec{x}, \vec{y}\}$. Mais si $(\vec{x}, \vec{y})$ n'est pas un swap, il peut y avoir des complétions $\succ$ et $\succ'$ de $\succ_{\mathcal{N}}$ telles que $\vec{x} \succ \vec{y}$ et $\vec{y} \succ' \vec{x}$. Il nous faut donc commencer par envisager les différentes manières d'estimer dans quelle mesure une structure SCP donné généralise un ensemble d'exemples donné.

Athienitou et Dimopoulos (2007) proposent des algorithmes pour apprendre un CP-net dont l'ordre partiel associé contient tous les exemples – nous dirons dans ce cas qu'il *implique* les exemples. Nous affirmons ici que cette exigence peut être trop forte, comme sur l'exemple suivant :

**Exemple 2** *Soient deux attributs binaires $X_1$ et $X_2$, de domaines respectifs $\{x_1, \overline{x_1}\}$ et $\{x_2, \overline{x_2}\}$, et l'ensemble d'exemples $\mathcal{E} = \{x_1x_2 \succ x_1\overline{x_2},\ x_1\overline{x_2} \succ \overline{x_1}x_2,\ \overline{x_1}x_2 \succ \overline{x_1}\overline{x_2}\}$.*

Que peut-on espérer apprendre de l'ensemble d'exemples $\mathcal{E}$ ci-dessus ? La fermeture transitive de $\mathcal{E}$ est la relation de préférence complète $x_1x_2 \succ x_1\overline{x_2} \succ \overline{x_1}x_2 \succ \overline{x_1}\overline{x_2}$. Cette relation de préférence est *séparable* : l'agent préfère inconditionnellement $x_1$ à $\overline{x_1}$ et $x_2$ à $\overline{x_2}$. Le fait que $x_1\overline{x_2}$ soit préféré à $\overline{x_1}x_2$ indique simplement qu'à choisir entre $X_1$ et $X_2$, l'agent préfère abandonner $X_2$ (on peut par exemple interpréter $X_1$ comme signifiant « devenir riche » et $X_2$ comme signifiant « il fera beau demain »). Intuitivement, puisque $\mathcal{E}$ est séparable, on peut espérer renvoyer une structure $\mathcal{N}$ qui contienne $x_1 \succ \overline{x_1}$ et $x_2 \succ \overline{x_2}$. Pourtant, aucune structure SCP n'implique $\mathcal{E}$ (en fait, aucun CP-net n'implique $\mathcal{E}$, quelles que soient les dépendances). La structure $\mathcal{N}$ induit une relation de préférence partielle dans laquelle $x_1\overline{x_2}$ et $\overline{x_1}x_2$ sont incomparables. Plus généralement, aucune structure *ceteris paribus* ne peut « expliquer » que $x_1 \succ \overline{x_1}$ est « plus important que » $x_2 \succ \overline{x_2}$. Ainsi, si on cherche une structure qui *implique* tous les exemples, on renverra simplement « échec ». Par contre, si on cherche une structure SCP qui est simplement *consistante* avec les exemples, c'est-à-dire qui n'implique par le contraire des exemples, alors on renverra $\mathcal{N}$.

L'explication est que quand un agent exprime un CP-net, la relation de préférence induite par ce CP-net n'est pas censée représenter la relation de préférence de l'agent dans sa totalité, mais un sous-ensemble de celle-ci. En d'autres termes, quand un agent exprime le CP-net $\mathcal{N}$, il exprime simplement qu'il préfère $x_1$ à $\overline{x_1}$ *ceteris paribus* – c'est-à-dire pour une valeur fixée de $X_2$ – et qu'il préfère aussi $x_2$ à $\overline{x_2}$ *ceteris paribus* ; le fait que $x_1\overline{x_2}$ et $\overline{x_1}x_2$ soient incomparables dans $\mathcal{N}$ ne veut sûrement pas dire que l'agent les considère comme effectivement incomparables, mais provient du fait que, techniquement, les CP-nets ne sont pas suffisamment expressifs pour représenter la préférence manquante $x_1\overline{x_2} \succ \overline{x_1}x_2$.[2]

Cet exemples suggèrent que, plutôt que de chercher un CP-net qui implique les exemples, on devrait chercher un CP-net qui induit une relation de préférence consistante avec les exemples. Une première manière de définir cette notion de consistance est d'imposer que le CP-net appris $\mathcal{N}$ soit tel que les exemples soient consistants avec au moins une relation de préférence (totale) qui étend $\mathcal{N}$. Dans certains cas, exiger qu'au moins une complétion de $\mathcal{N}$ contienne tous les exemples peut être trop fort, notamment lorsque les exemples proviennent de plusieurs utilisateurs (lorsqu'on veut apprendre les préférences génériques d'un groupe d'utilisateurs), ou d'un seul utilisateur, mais dans plusieurs contextes:

[2] Des langages plus expressifs, comme les TCP-nets [Brafman *et al.*, 2006] ou les théories de préférences conditionnelles [Wilson, 2004] permettraient de représenter ces préférences.

**Exemple 3** *Supposons qu'on observe que tous les utilisateurs d'un même groupe préfèrent inconditionnellement $x_1$ à $\overline{x_1}$ et $x_2$ à $\overline{x_2}$, alors que leurs préférences entre $x_1\overline{x_2}$ et $\overline{x_1}x_2$ diffèrent (on peut interpréter $x_1$ et $x_2$ comme signifiant respectivement « être invité à un bon dîner » et « recevoir une prime de 50 euros »). Alors $\mathcal{E} \supseteq \{\overline{x_1}x_2 \succ x_1\overline{x_2},\ x_1\overline{x_2} \succ \overline{x_1}x_2\}$ : il est clair que $\mathcal{E}$ est inconsistent, et il ne peut donc pas y avoir de structure de préférence qui induise un ordre qu'on pourrait compléter en une relation de préférence linéaire qui contienne $\mathcal{E}$. Pourtant, si $\mathcal{N} = \{x_1 \succ \overline{x_1}, x_2 \succ \overline{x_2}\}$, alors chaque exemple de $\mathcal{E}$ est (individuellement) contenu dans au moins une complétion of $\succ_{\mathcal{N}}$.*

On définit maintenant trois notions de compatibilité entre une structure SCP et un ensemble d'exemples. (Ces définitions seraient aussi valables pour le cas de CP-nets de structures quelconques.)

**Définition 1** *Soit $\mathcal{N}$ une structure SCP sur $\mathcal{V}$. Un exemple $(\vec{x}, \vec{y})$ est* impliqué par *$\mathcal{N}$ si $\vec{x} \succ \vec{y}$ pour toute complétion $\succ$ de $\succ_{\mathcal{N}}$ ; $(\vec{x}, \vec{y})$ est* consistant avec *$\mathcal{N}$ s'il existe une complétion $\succ$ de $\succ_{\mathcal{N}}$ telle que $\vec{x} \succ \vec{y}$. De plus, nous dirons qu'un ensemble d'exemples $\mathcal{E}$ est :*
- impliqué par *$\mathcal{N}$ si tout exemple de $\mathcal{E}$ est impliqué par $\mathcal{N}$ ;*
- globalement (ou fortement) compatible avec *$\mathcal{N}$ s'il existe une complétion $\succ$ de $\succ_{\mathcal{N}}$ telle que pour tout $(\vec{x}, \vec{y}) \in \mathcal{E}$, $\vec{x} \succ \vec{y}$ ;*
- faiblement compatible avec *$\mathcal{N}$ si tout exemple $(\vec{x}, \vec{y}) \in \mathcal{E}$ est individuellement consistent avec $\mathcal{N}$.*

*Enfin, nous dirons que $\mathcal{E}$ est implicativement / fortement / faiblement* séparable *s'il est impliqué par / fortement compatible avec / faiblement compatible avec au moins une structure SCP.*

Il est clair que la compatibilité forte implique la compatibilité faible, et que si $\mathcal{E}$ est impliqué par $\mathcal{N}$, alors $\mathcal{E}$ est aussi fortement compatible avec $\mathcal{N}$.

## 3 Compatibilité faible

On commence par décrire comment la recherche d'une structure SCP qui soit faiblement compatible avec un ensemble d'exemples donné peut être ramenée à un problème de satisfaisabilité propositionnelle. Dans le cas d'attributs binaires, on propose la traduction suivante d'un ensemble d'exemples en un ensemble de clauses propositionnelles, dont, nous le verrons ensuite, les modèles correspondent à des structures SCP faiblement consistantes avec les exemples.

À chaque exemple $\vec{x} \succ \vec{y}$ on associe la clause $C_{\vec{x}\succ\vec{y}}$ qui contient $x_i$ si et seulement si $x_i \in \mathit{Diff}(\vec{x}, \vec{y})$ et $\neg x_i$ si et seulement si $\overline{x_i} \in \mathit{Diff}(\vec{x}, \vec{y})$. Par exemple, si $\vec{x} = x_1\overline{x_2}x_3x_4$ et $\vec{y}{=}\overline{x_1}x_2x_3\overline{x_4}$ alors $\mathit{Diff}(\vec{x}, \vec{y}) = \{x_1, \overline{x_2}, x_4\}$ et $C_{\vec{x}\succ\vec{y}} = x_1 \vee \neg x_2 \vee x_4$. Cette clause exprime le fait que $x_1$ est préféré à $\overline{x}_1$, ou $\overline{x}_2$ est preferred à $x_2$, ou $x_4$ est préféré à $\overline{x}_4$. (L'explication à cela est que si une structure SCP $\mathcal{N}$ contient $\{\overline{x}_1 \succ x_1, x_2 \succ \overline{x}_2, \overline{x}_4 \succ x_4\}$, alors d'après le lemme 1 on a $\vec{y} \succ_{\mathcal{N}} \vec{x}$, et donc $\mathcal{N}$ n'est pas consistent avec $\vec{x} \succ \vec{y}$.) Si $\mathcal{E}$ est un ensemble d'exemples, alors on pose $\Phi_{\mathcal{E}} = \{C_e \mid e \in \mathcal{E}\}$.

On peut maintenant définir une correspondance bijective entre les interprétations de $\{x_1, \ldots, x_n\}$ et les structures SCP sur $\mathcal{V}$. Si $M$ est une telle interprétation, soit $\mathcal{N}_M$ la structure qui contient $x_i \succ \overline{x_i}$ pour tout $i$ tel que $M \models x_i$ et la préférence $\overline{x_i} \succ x_i$ pour tout $i$ tel que $M \models \neg x_i$. Ainsi, si $M(x_1) =$ vrai, $M(x_2) =$ faux, $M(x_3) =$ faux et $M(x_4) =$ vrai alors $\mathcal{N}_M$ contient les tables de préférence $\{x_1 \succ \overline{x_1},\ \overline{x_2} \succ x_2,\ \overline{x_3} \succ x_3,\ x_4 \succ \overline{x_4}\}$.

**Proposition 1** *Soit $M$ une interprétation. Un ensemble d'exemples $\mathcal{E}$ est faiblement consistant avec $\mathcal{N}_M$ si et seulement si $M \models \Phi_{\mathcal{E}}$.*

**Proof :** On montre que $\mathcal{N}_M \not\models \vec{y} \succ \vec{x}$ si et seulement si $M \models C_{\vec{x}\succ\vec{y}}$. On peut supposer sans perte de généralité que $\vec{x} = x_1 \ldots x_n$ et $\vec{y} = \overline{x_1} \ldots \overline{x_i}x_{i+1} \ldots x_n$. Alors $M \not\models C_{\vec{x}\succ\vec{y}}$ si et seulement si $\mathcal{N}_M$ contient $\overline{x_1} \succ x_1, \ldots, \overline{x_i} \succ x_i$, ce qui, d'après le lemme 1, est équivalent à $\mathcal{N}_M \models \vec{y} \succ \vec{x}$. ∎

**Corollaire 1** *$\mathcal{E}$ est faiblement séparable si et seulement si $\Phi_{\mathcal{E}}$ est satisfaisable.*

En conséquence, la séparabilité faible se ramène à un test de satisfaisabilité.

**Exemple 4** *Soient les trois attributs binaires $A, B, C$, et l'ensemble d'exemples*

$$\mathcal{E} = \left\{ \begin{array}{l} abc \succ \overline{a}bc, \overline{a}\overline{b}c \succ ab\overline{c}, \\ \overline{a}b\overline{c} \succ a\overline{b}c, \overline{a}bc \succ \overline{a}\overline{b}\overline{c} \end{array} \right\}$$

*$\Phi_{\mathcal{E}}$ a un modèle unique, qui correspond à la structure SCP $\mathcal{N} = \{a \succ \overline{a}, b \succ \overline{b}, c \succ \overline{c}\}$. $\mathcal{N}$ est donc l'unique structure SCP faiblement compatible avec $\mathcal{N}$, ce qui montre que $\mathcal{E}$ est faiblement séparable.*

**Proposition 2** *Décider si un ensemble d'exemples sur des attributs binaires est faiblement séparable est* NP*-complet.*

**Proof :** L'appartenance à NP est une conséquence directe du corollaire 1. Pour montrer la NP-difficulté, on utilise une réduction depuis 3SAT : soit $\Phi$ un ensemble de clauses à trois littéraux non tautologiques. Pour toute clause $C = l_1 \vee l_2 \vee l_3$ de $\Phi$ on crée un exemple $e_C = (\vec{x} \succ \vec{y})$ avec
– $\vec{x} = \varepsilon_1.x_1\varepsilon_2.x_2 \ldots \varepsilon_n.x_n$ and
– $\vec{y} = \varepsilon'_1.x_1\varepsilon'_2.x_2 \ldots \varepsilon'_n.x_n$,
où, pour tout $i$, $\varepsilon_i.x_i = \varepsilon'_i.x_i = x_i$, sauf si $l_j = \neg x_i$ pour un certain $j$, auquel cal $\varepsilon_i.x_i = \neg x_i$ ; et si $l_j = x_i$ pour un certain $j$, alors $\varepsilon'_i.x_i = \neg x_i$. Soit alors $\mathcal{E}_\Phi = \{e_C \mid C \in \Phi\}$. Par exemple, si $\Phi = \{x_1 \vee \neg x_2 \vee x_3, \neg x_1 \vee x_2 \vee x_4, x_2 \vee x_3 \vee x_4\}$ alors $\mathcal{E}_\Phi = \{x_1\overline{x_2}x_3x_4 \succ \overline{x_1}x_2\overline{x_3}x_4, \overline{x_1}x_2x_3x_4 \succ x_1\overline{x_2}x_3\overline{x_4}, x_1x_2x_3x_4 \succ x_1\overline{x_2x_3x_4}\}$. On vérifie facilement que $\Phi_{\mathcal{E}_\Phi} = \Phi$, et donc, d'après le corollaire 1 on conclut que $\Phi$ est satisfiable si et seulement si $\mathcal{E}_\Phi$ est faiblement séparable. ∎

La généralisation à des domaines non binaires n'est pas difficile. Au lieu d'avoir un symbole propositionnel par attribut, on a un symbole propositionnel par paire de valeurs de chaque attribut. Par exemple, dans le cas d'un attribut $X$ de domaine $\{d_1, d_2, d_3\}$, on a trois symboles $d_1 \succ d_2$, $d_1 \succ d_3$ et $d_2 \succ d_3$ ($d_2 \succ d_1$ étant équivalent à $\neg(d_1 \succ d_2)$, etc.). Par exemple, si $Y$ est un attribut binaire, la clause correspondant à l'exemple $d_1y \succ d_2\overline{y}$ est $(d_1 \succ d_2) \vee y$. La différence principale avec le cas binaire est qu'il faut maintenant des clauses pour imposer la transitivité des préférences locales. Soit $Trans = \bigwedge_{X_i \in \mathcal{V}} Trans_{X_i}$ la formule propositionnelle exprimant cette transitivité – par exemple, avec $D_1 = \{d_1, d_2, d_3\}$ on a

$$Trans_{X_1} = (d_1 \succ d_2 \wedge d_2 \succ d_3 \rightarrow d_1 \succ d_3) \wedge \ldots$$

On remarque que $Trans$ est de taille polynomiale. On a maintenant une correspondance bijective entre les structures SCP et les interprétations qui satisfont $Trans$: $M \models \Phi_{\mathcal{E}} \wedge Trans$ si et seulement si $\mathcal{N}_M$ est faiblement consistent avec $\mathcal{N}_M$. En conséquence, $\mathcal{E}$ est faiblement séparable si et seulement si $\Phi_{\mathcal{E}} \wedge Trans$ est satisfaisable.

Cette traduction montre que la consistance faible pour des attributs non binaires est dans NP, et donc que le problème reste NP-complet.

Dès que $\mathcal{E}$ devient grand par rapport au nombre d'attributs $n$, la probabilité que $\mathcal{E}$ soit faiblement séparable devient faible. Dans ce cas, on pourrait vouloir déterminer une structure SCP qui soit faiblement compatible avec le plus grand nombre d'exemples de $\mathcal{E}$ possible. Le corollaire 1 appliqué à des sous-ensembles de $\mathcal{E}$ montre que ce problème revient à un problème de type MAXSAT. La structure SCP qui correspond le mieux à un ensemble d'exemples donné (au sens de la compatibilité faible) correspond à l'interprétation qui maximise le nombre de clauses de $\Phi_{\mathcal{E}}$ satisfaites. Ce résultat peut être étendu au cas de variables non binaires, avec la différence que les clauses de $Trans$ sont protégées. La propositions 1 indique aussi qu'on peut utiliser des méthodes d'approximation polynomiale ; et que, dans le cas binaire, si pour chaque exemple $(\vec{x}, \vec{y})$ de $\mathcal{E}$, $\vec{x}$ et $\vec{y}$ diffèrent sur au plus deux attributs, alors on peut tester si $\mathcal{E}$ est faiblement séparable en temps polynomial (mais le problème d'optimisation reste NP-difficile dans ce cas).

## 4 Compatibilité forte

Il est moins facile de caractériser la compatibilité forte. La différence entre compatibilité faible et forte tient à ceci : alors que pour la compatibilité faible on cherche une structure SCP consistante avec chaque exemple de $\mathcal{E}$ pris individuellement, pour la compatibilité forte on cherche une structure SCP consistante avec tout l'ensemble d'exemples $\mathcal{E}$, pris globalement.

**Exemple 4, suite** *$\mathcal{E}$ n'est pas fortement compatible avec $\mathcal{N}$, parce que $\mathcal{E} \cup \succ_{\mathcal{N}}$ possède le cycle suivant :*

$$a\overline{b}c \succ_{\mathcal{N}} \overline{a}\overline{b}c \succ_{\mathcal{E}} ab\overline{c} \succ_{\mathcal{N}} \overline{a}b\overline{c} \succ_{\mathcal{E}} a\overline{b}c$$

*Puisque $\mathcal{E}$ n'est fortement compatible avec aucune autre structure SCP que $\mathcal{N}$ (et ceci parce que $\mathcal{N}$ est la seule structure SCP faiblement compatible avec $\mathcal{N}$), $\mathcal{E}$ n'est pas fortement séparable.*[3]

[3] Remarquons que $\mathcal{E}$ est à la fois faiblement consistant et sans cycles, comme on a vu à l'example 3, et pourtant il n'est pas fortement consistent.

Remarquons que toutes les alternatives contenues dans le cycle de l'exemple ci-dessus apparaissent dans $\mathcal{E}$. Plus généralement, si on note $D(\mathcal{E})$ l'ensemble des alternatives de $D$ qui apparaissent dans $\mathcal{E}$, alors on a la caractérisation suivante de la compatibilité forte :

**Proposition 3** *$\mathcal{E}$ est fortement compatible avec $\mathcal{N}$ si et seulement si la restriction de $\succ_{\mathcal{N}} \cup \mathcal{E}$ à $D(\mathcal{E})$ est acyclique.*

**Proof :** Par définition, si $\mathcal{E}$ est fortement compatible avec $\mathcal{N}$ alors $\succ_{\mathcal{N}}$ peut être complété en un ordre linéaire contenant $\mathcal{E}$, donc $\succ_{\mathcal{N}} \cup \mathcal{E}$ est acyclique, et sa restriction à $D(\mathcal{E})$ aussi. Réciproquement, supposons que $\mathcal{E}$ n'est pas fortement compatible avec $\mathcal{N}$: $\succ_{\mathcal{N}} \cup \mathcal{E}$ ne peut pas être étendu en un ordre linéaire, ce qui signifie qu'elle n'est pas irreflexive, donc qu'elle contient un cycle $\Gamma$. Puisque les structures SCP sont consistantes, $\Gamma$ doit contenir au moins un arc de $\mathcal{E}$. On peut transformer $\Gamma$ en un cycle $\Gamma'$ dont les sommets sont tous dans $D(\mathcal{E})$: il suffit de remplacer, tant que c'est possible, toute séquence de sommets dans $\Gamma$ de la forme $x \succ_{\mathcal{N}} y \succ_{\mathcal{N}} z$ par $x \succ_{\mathcal{N}} z$ (ce qui est possible parce que $\succ_{\mathcal{N}}$ est transitive). Or, tout arc dans $\Gamma'$ est soit dans $\mathcal{E}$, soit adjacent à deux arcs de $\mathcal{E}$ (il ne peut pas être réduit à un arc réflexif simple de $\succ_{\mathcal{N}}$, parce que $\succ_{\mathcal{N}}$ est irreflexive), donc ses extrémités sont dans $D(\mathcal{E})$. Donc $\Gamma'$ est un cycle de la restriction de $\succ_{\mathcal{N}} \cup \mathcal{E}$ à $D(\mathcal{E})$. ∎

Puisque la restriction de $\succ_{\mathcal{N}} \cup \mathcal{E}$ à $D(\mathcal{E})$ a au plus $2.|\mathcal{E}|$ sommets, on peut déterminer s elle possède un cycle en temps polynomial, donc déterminer si $\mathcal{E}$ est fortement compatible avec $\mathcal{N}$ est dans P. Puisque la taille d'une structure SCP est linéaire par rapport au nombre de variables, déterminer si $\mathcal{E}$ est fortement séparable est dans NP.

**Proposition 4** *Déterminer si $\mathcal{E}$ est fortement séparable est* NP*-complet.*

**Proof :** La difficulté vient de la réduction suivante depuis le problème de SÉPARABILITÉ FAIBLE. Soit $\mathcal{E} = \{\vec{x}_1 \succ \vec{y}_1, \ldots, \vec{x}_m \succ \vec{y}_m\}$ un ensemble d'exemples, construit sur un ensemble de symboles propositionnels $X$. À $\mathcal{E}$ on associe l'ensemble d'examples $\mathcal{E}'$, construit sur l'ensemble d'attributs $X \cup \{P_1, \ldots, P_m\}$. Pour tout $e_i = \vec{x}_i \succ \vec{y}_i$ in $\mathcal{E}$ on crée l'exemple $e'_i$ suivant:

$$\vec{x}_i\overline{p}_1 \ldots \overline{p}_{i-1}p_i\overline{p}_{i+1} \ldots \overline{p}_m \succ \vec{y}_i\overline{p}_1 \ldots \overline{p}_{i-1}p_i\overline{p}_{i+1} \ldots \overline{p}_m.$$

On a alors que $\mathcal{E}' = \{e'_i \mid e_i \in \mathcal{E}\}$ est fortement séparable si et seulement si $\mathcal{E}$ est faiblement séparable. Supposons que $\mathcal{E}$ est faiblement séparable : il existe alors une structure SCP $\mathcal{N}$ consistante avec chaque $e_i$ in $\mathcal{E}$. Soit $\mathcal{N}' = \mathcal{N} \cup \{p_1 \succ \overline{p}_1, \ldots, p_n \succ \overline{p}_n\}$. Pour tout $i$, parce que $\mathcal{N}$ est consistante avec $\vec{x}_i \succ \vec{y}_i$, il existe un ordre linéaire $\succ_i$ étendant $\succ_{\mathcal{N}}$ et contenant $\vec{x}_i \succ \vec{y}_i$. Soit $\succ$ la relation de préférence définie par: $\vec{x}\vec{p} \succ \vec{y}\vec{p'}$ si et seulement si l'une de ces conditions est vérifiée :

(a) $\vec{p}$ est lexicographiquement plus grand que $\vec{p'}$ (ce que nous notons $\vec{p} >_{lex} \vec{p'}$), c'est-à-dire, il existe un $i \leq m$ tel que $\vec{p}$ contient $p_i$, $\vec{p'}$ contient $\overline{p}_i$, et pour tout $j < i$, $\vec{p}$ contient $p_j$ si et seulement si $\vec{p'}$ contient $p_j$.

(b) $\vec{p}=\vec{p'}=\overline{p}_1 \ldots \overline{p}_{i-1}p_i\overline{p}_{i+1} \ldots \overline{p}_m$ pour un $i$, et $\vec{x} \succ_i \vec{y}$;

(c) $\vec{p} = \vec{p'}$, $\vec{p}$ n'est pas de la forme $\overline{p}_1 \ldots \overline{p}_{i-1}p_i\overline{p}_{i+1} \ldots \overline{p}_m$, et $\vec{x} \succ^* \vec{y}$, où $\succ^*$ est une relation de préférence étendant $\succ_N$.

On peut vérifier que $\succ$ est une relation de préférence complète $\succ$ étendant $\succ_{\mathcal{N}'}$, et contenant $\mathcal{E}'$, ce qui implique que $\mathcal{E}'$ est fortement consistant avec $\mathcal{N}'$, et que $\mathcal{E}'$ est fortement séparable .

Réciproquement, supposons que $\mathcal{E}'$ soit fortement séparable . Soit $\mathcal{N}'$ fortement consistant avec $\mathcal{E}'$, et $\succ$ étendant $\succ_{\mathcal{N}'}$ et contenant $\mathcal{E}'$. Pour tout $i \leq m$, définissons $\succ_i$ comme suit : $\vec{x} \succ_i \vec{y}$ *iff* $\vec{x}\overline{p}_1 \ldots \overline{p}_{i-1}p_i\overline{p}_{i+1} \ldots \overline{p}_m \succ \vec{y}\overline{p}_1 \ldots \overline{p}_{i-1}p_i\overline{p}_{i+1} \ldots \overline{p}_m$. Alors, pour tout $i$: $\succ_i$ est un ordre linéaire sur $2^X$ ; $\succ_i$ étend $\succ_{\mathcal{N}}$ ; et $\succ_i$ contient $\vec{x}_i \succ \vec{y}_i$, parce que $\succ$ contient $\vec{x}_i\overline{p}_1 \ldots \overline{p}_{i-1}p_i\overline{p}_{i+1} \ldots \overline{p}_m \succ \vec{y}_i\overline{p}_1 \ldots \overline{p}_{i-1}p_i\overline{p}_{i+1} \ldots \overline{p}_m$. Donc, pour tout $i$ on a trouvé une extension $\succ_i$ de $\succ_{\mathcal{N}}$ contenant $\vec{x}_i \succ \vec{y}_i$, ce qui implique that $\mathcal{E}$ est faiblement compatible avec $\mathcal{N}$, et que $\mathcal{E}$ est faiblement séparable . ∎

Remarquons que bien que la séparabilité faible et la séparabilité forte aient la même complexité, la séparabilité faible a la propriété intéressante

qu'il existe une traduction simple en SAT préservant les solutions (les modèles de $\Phi_{\mathcal{E}}$ correspondent bijectivement aux structures SCP-nets faiblement consistantes avec $\mathcal{E}$), ce qui permet à la consistance faible d'être calculée en pratique en utilisant des algorithmes pour SAT. Pour calculer une structure SCP fortement consistante avec $\mathcal{E}$, on peut générer des structures $\mathcal{N}$ faiblement consistantes avec $\mathcal{E}$, et tester ensuite l'acyclicité de $\succ_{\mathcal{N}} \cup \mathcal{E}$.

## 5 Compatibilité implicative

Il est facile de déterminer s'il existe une structure SCP qui implique $\mathcal{E}$. Dans le cas de variables binaires, à chaque exemple $\vec{x} \succ \vec{y}$ on associe la conjonction de littéraux $\Gamma_{\vec{x}\succ\vec{y}} = \neg C_{\vec{y}\succ\vec{x}}$. Soit $\Gamma_{\mathcal{E}} = \bigwedge\{\Gamma_e \mid e \in \mathcal{E}\}$. À l'aide du Lemme 1, on peut prouver que $M \models \Gamma_{\mathcal{E}}$ si et seulement si $\mathcal{N}_M$ implique $\mathcal{E}$. La recherche d'un modèle d'une conjonction de clauses se fait en temps polynomial, donc déterminer si un ensemble d'exemples est implicativement séparable est dans P.

Bien que la séparabilité implicative soit plus facile que les séparabilités forte et faible, comme nous l'avons expliqué plus haut, nous pensons que c'est une notion bien trop forte.

## 6 Travaux connexes

L'apprentissage de préférences est un champ de recherches actif, qui s'intéresse à divers aspects de l'apprentissage de préférences, notamment l'arrangement de labels [Hüllermeier *et al.*, 2008] ou l'arrangement d'objets [Cohen *et al.*, 1999], mais jusqu'à présent ces recherches se sont focalisées sur des domaines simples, non combinatoires[4].

L'apprentissage de fonctions numériques sur des domaines combinatoires a été considéré, notamment dans [Ha et Haddawy, 1997; Gonzales et Perny, 2004]. Récemment, Domshlak et Joachims [2007] ont proposé une méthode pour apprendre une fonction d'utilité à partir de préférences entre formules logiques. En ce qui concerne l'apprentissage de préférences *ordinales* sur des domaines multiattributs, une approche qui suit une méthodologie proche de la nôtre (mais pour une classe très différente de structures de préférences) est le travail sur l'apprentissage de préférences lexicographiques sur des domaines multiattributs [Schmitt et Martignon, 2006; Dombi *et al.*, 2007; Yaman *et al.*, 2008]. Une différence importante avec notre travail est qu'ils apprennent la relation de préférence en entier, alors que nous n'apprenons qur les préférences locales. Bien entendu, apprendre la relation de préférence en entier donne une information plus riche, mais cela a un prix : l'hypothèse que les préférences sont lexicographiques est extrêmement restrictive. Les préférences lexicographiques sont une forme très spécifique de préférences, que l'on ne rencontre que rarement en pratique. Il faut remarquer que toute relation de préférence lexicographique est séparable, mais que l'ensemble des relations de préférence séparables est bien plus grand. Il est intéressant de noter que les résultats de complexité sont similaires.

Koriche et Zanuttini [2009] montrent qu'il est possible d'éliciter efficacement des CP-nets en utilisant des requêtes simples de la forme $\vec{x} \succ^{?} \vec{y}$, où $\vec{x}$ et $\vec{y}$ diffèrent seulement sur la valeur d'une variable. L'apprentissage passif de CP-nets a été considéré par Athienitou et Dimopoulos [2007]. Comme nous l'avons expliqué plus haut, ils ne s'intéressent qu'à la compatibilité implicative, ce qui limite considérablement l'applicabilité de leur approche. Sachdev [2007] propose d'apprendre des "théories de préférence" (au sens de Doyle *et al.* [1991]) consistantes avec un ensemble d'exemples. Il suppose qu'on a accès à l'ensemble complet des exemples correspondant à une théorie de préférence existante, ce qui limite l'applicabilité pratique de son approche.

## 7 Conclusion et travaux futurs

Notre contribution est double: nous avons montré que la recherche d'une relation de préférence *ceteris paribus* séparable (ou de manière équivalente, la recherche d'un CP-net séparable) compatible avec un ensemble d'exemples n'est pas immédiate à définir en raison du fait que, au contraire des préférences lexicographiques, les préférences séparables ne permettent pas d'induire une relation de préférence complète sur le domaine multiattributs à partir des relations de préférence locales. Nous avons proposé et

[4] Des domaines multiattributs sont parfois considérés, mais avec un rôle très différent : par exemple, en arrangement de labels on apprend une relation de préférence sur un domaine simple, étant donnée une information sous la forme d'un vecteur de valeurs d'attributs (par exemple, on cherche à prédire la façon dont un utilisateur va ordonner un ensemble de candidats à une élection, étant donnés son sexe, son âge et sa profession).

discuté trois formes de compatibilité entre un CP-net séparable et un ensemble d'exemples. Pour chacune d'entre elles, nous avons identifié sa complexité et donné des caractérisations qui peuvent être utilisées pour calculer un CP-net séparable qui correspond au mieux à un ensemble d'exemples. Bien que la séparabilité implicative soit plus facile à déterminer que la séparabilité faible et la séparabilité forte, elle est toutefois bien moins intéressante que les deux autres notions.

Plus généralement, nos définitions de compatibilités implicative, forte et faible sont pertinentes pour n'importe quel CP-net. Jusqu'à présent nous n'avons des caractérisations et des résultats de complexité pour ces trois formes de consistance que dans le cas séparable, c'est-à-dire le cas où le graphe de dépendence n'a pas d'arc. Une question à long terme consisterait à trouver des méthodes pour apprendre des CP-nets avec n'importe quelle structure, en se focalisant d'abord sur les structures simples, par exemples les hyperarbres. Une première étape consisterait à apprendre des CP-nets avec un graphe *fixé*: étant donné un ensemble d'exemples $\mathcal{E}$ et un graphe de dépendance $G$, renvoyer le CP-net maximisant la $G$-compatibilité (pour chacune des trois formes de compatibilité) avec $\mathcal{E}$. Malheureusement, cela n'est pas si simple. Notre traduction d'exemples en clauses propositionnelles peut être généralisée, en utilisant pour chaque attribut binaire une variable propositionnelle pour toute combinaison de valeurs de ses parents. Par exemple, si $A$ n'a aucun parent et $B$ n'a comme parent que $A$, on utilise trois variables propositionnelles : $a$, $(a{:}b)$ et $(\overline{a}{:}b)$, où, par exemple, l'affectation de $(a{:}b)$ à vrai correspond à l'entrée $a{:}b{\succ}\overline{b}$ dans la table de préférence de $B$. La compatibilité faible peut alors être exprimée en utilisant, pour chaque exemple $\vec{x}{\succ}\vec{y}$, une clause pour chaque séquence de swaps de $\vec{y}$ à $\vec{x}$, exprimant qu'au moins une des étapes de la séquence est bloquée. Ceci garantit qu'on n'a pas $\vec{y}{\succ_{\mathcal{N}}}\vec{x}$. Mais il peut y avoir un nombre exponentiel de telles clauses, même dans le cas où le nombre de parents dans le graphe de dépendance est borné, ce qui limite considérablement l'applicabilité de cette traduction.

## Remerciements

Un grand merci à Richard Booth, Yann Chevaleyre, Kevin Garcia, Peter Haddawy, Frédéric Koriche, Mathieu Serrurier, Chattrakul Sombatheera et Bruno Zanuttini pour de nombreuses discussions au sujet de l'apprentissage de CP-nets. Cette recherche a partiellement bénéficié du soutien du projet ANR-05-BLAN-0384 "Preference Handling et Aggregation in Combinatorial Domains".

## References

F. Athienitou et Y. Dimopoulos. Learning CP-networks: a preliminary investigation. In *Proc. 3rd Multidisciplinary Work. on Advances in Preference Handling (PREF'07)*, 2007.

C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, et D. Poole. CP-nets: a tool for representing et reasoning avec conditional ceteris paribus préférence statements. *JAIR*, 21:135–191, 2004.

W. J. Bradley, J. K. Hodge, et D.M. Kilgour. Separable discrete preferences. *Mathematical Social Science*, 49(3):335–353, 2005.

R. Brafman, C. Domshlak, et S. Shimony. On graphical modeling of préférence et importance. *JAIR*, 25:389–424, 2006.

D. Braziunas et C. Boutilier. Elicitation of factored utilities. *AI Magazine*, 29 (4), 2008.

L. Chen et P. Pu. Survey of préférence elicitation methods. Tech. Rep. 200467, EPFL, 2004.

W. Cohen, R. Schapire, et Y. Singer. Learning to order things. *JAIR*, 10:243–270, 1999.

J. Dombi, C. Imreh, et N. Vincze. Learning lexicographic orders. *Eur. J. of Operational Research*, 183:748–756, 2007.

C. Domshlak et T. Joachims. Efficient et non-parametric reasoning over user preferences. *User Modeling et User-Adapted Interaction (UMUAI)*, 17(1-2):41–69, April 2007.

J. Doyle, Y. Shoham, et M. Wellman. A logic of relative desire (preliminary report). In *Proc. ISMIS '91*, 16–31. 1991.

C. Gonzales et P. Perny. GAI networks for utility elicitation. In *Proc. KR'04*, pages 224–233. AAAI Press, 2004.

V. Ha et P. Haddawy. Problem-focused incremental elicitation of multi-attribute utility models. In *Proc. UAI-97*, pages 215–222, 1997.

E. Hüllermeier, J. Fürnkranz, W. Cheng, et K. Brinker. Label ranking by learning pairwise preferences. *AIJ*, 172:1897–1917, 2008.

R. Keeney et H. Raiffa. *Decision avec Multiple Objectives: Preferences et Value Trade-offs*. Wiley, 1976.

F. Koriche et B. Zanuttini. Learning conditional préférence networks avec queries, 2009. In *Proc. IJCAI'09*.

M. Sachdev. On learning of ceteris paribus préférence theories. Master's thesis, Graduate Faculty of North Carolina State University, 2007.

M. Schmitt et L. Martignon. On the complexity of learning lexicographic strategies. *J. Mach. Learn. Res.*, 7:55–83, 2006.

P. Viappiani, B. Faltings, et P. Pu. Preference-based search using example-critiquing avec suggestions. *JAIR*, 27:465–503, 2006.

N. Wilson. Extending CP-Nets avec Stronger Conditional Preference Statements. In *AAAI-04*, pages 735–741, 2004.

F. Yaman, T. Walsh, M. Littman, et M. desJardins. Democratic approximation of lexicographic préférence models. In *Proc. ICML '08*, pages 1200–1207, 2008.

# La bienveillance et les préférences comme paramètres de l'interprétation des recommandations vagues

S. Leblois⋆ J. F. Bonnefon⋆
leblois@univ-tlse2.fr bonnefon@univ-tlse2.fr

⋆Université de Toulouse
CLLE (Cognition, Langues, Langage et Ergonomie); UTM, EPHE, CNRS;
Maison de la Recherche, Université de Toulouse le Mirail,
5 allées Antonio Machado, 31058 Toulouse Cedex 9, FRANCE

**Résumé :**
Aussi fréquentes qu'elles puissent être dans la vie quotidienne, les recommandations vagues sont beaucoup moins étudiées que leurs homologues précises. Cet article propose un modèle de l'impact de trois variables sur l'utilisation de recommandations vagues : le format de l'imprécision, la bienveillance entre juge et conseiller et le détenteur des préférences menacées par le conseil. Les résultats empiriques rapportés dans l'article montrent que la bienveillance a un effet indépendant positif sur le poids accordé à une recommandation vague, et que les deux autres variables interagissent dans le sens d'un biais de désirabilité, qui prédit une moindre utilisation du conseil verbal contrapréférentiel lorsque les préférences menacées sont celles de la personne conseillée. Cette étude est la première étape d'un programme de recherche qui se consacre aux facteurs individuels et situationnels intervenant dans l'utilisation de conseils vagues, qui sera exposé dans la conclusion.

**Mots-clés :** recommandation vague, bienveillance, désirabilité, tact, expérimentation

**Abstract:**
However common they are in everyday life, vague recommendations are much less studied than their precise counterparts. This paper addresses the impact of three variables on the uptake of vague recommendations : vagueness format, benevolence of the advisor, and preference locus. Results show that the weight given to the recommendation is greater when the advisor is expectedly benevolent ; the interaction between the vagueness format and the preference locus agrees with a wishful thinking framework that predicts a lower uptake of verbal counterpreferential advice when the preferences at stake are that of the advisee. This study is a first step of a larger research programme about individual and situational factors affecting the uptake of vague recommendations, which is delineated in the conclusion.

**Keywords:** vague recommendation, benevolence, desirability, tact, experimentation

Imaginez que votre conjoint vous dise les choses suivantes : « Tu devrais manger un peu moins gras, faire du sport 2 à 3 fois par semaine et marcher environ une heure par jour.» Toutes ces recommandations sont vagues dans le sens où elles ne vous donnent pas d'informations précises quant à la quantité exacte de gras qu'il faut soustraire de votre régime alimentaire, à la régularité de la pratique sportive à adopter ou encore à la durée exacte conseillée de marche quotidienne. Au lieu de donner des prescriptions précises à appliquer à la lettre, ces recommandations ne font que fournir des informations vagues (ou imprécises) à l'aide de quantificateurs (un peu moins), d'intervalles de valeurs (de 2 à 3 fois par semaine) ou encore de tendance centrale (environ une heure) [1]. Alors que ce type de conseil est largement utilisé dans la vie quotidienne, peu de travaux s'y sont déjà intéressés ; il existe une littérature riche sur les conseils mais celle-ci se concentre uniquement sur les recommandations précises.

Cet article propose un modèle psychologique de l'interprétation des conseils vagues, construit autour de trois paramètres : la bienveillance du conseiller, le format de l'imprécision, et le détenteur des préférences éventuellement menacées par le conseil. Différentes interactions sont théoriquement envisageables entre ces variables. Ces différentes possibilités sont mises à l'épreuve d'une expérimentation dont les résultats sont rapportés dans l'article.

## 1 Les Conseils Vagues

Recevoir un conseil vague peut être frustrant. La plupart des gens préfèrent en effet recevoir des informations quantitatives sous un format numérique et précis [5, 16]. Cependant, plusieurs raisons peuvent pousser un conseiller à utiliser des recommandations vagues, ce qui explique leur usage fréquent dans la vie de tous les jours. Tout d'abord, le conseiller peut être contraint d'être vague tout simplement parce qu'il ne peut pas faire mieux. Il peut ne pas disposer des connaissances requises pour formuler un conseil précis : votre conjoint ne possède sans doute pas

1. Notre notion d'une recommandation «vague» recouvre différentes notions qu'une analyse conceptuelle plus fine devrait distinguer. Ainsi, l'expression «de 10 à 20%» spécifie un intervalle précis, au sein duquel le destinataire peut choisir. Le vague en question est donc une forme de généralité. Cette forme de vague serait à distinguer de la notion de sous-spécification sémantique qui concerne des expressions de quantification comme «un peu» ou «beaucoup».

les connaissances d'un préparateur physique qui lui permettrait de vous dire précisément à partir de combien de séances de sport par semaine votre forme s'améliorera ; ou encore la situation peut ne pas se prêter à des recommandations précises et correctes : existe-t-il réellement une fréquence précise idéale de pratique sportive ?

Être vague ne répond pas seulement à une telle contrainte. Cela peut être également une stratégie visant à réduire le risque d'offenser l'autre, que nous appellerons décideur. En effet, un conseil peut être aussi intrusif qu'un ordre pour celui qui le reçoit. Peut-être réagiriez-vous mal si votre conjoint vous adressait les recommandations données en exemple au début de cet article. Cependant, leur formulation vague les rend bien moins dérangeante que si elles étaient formulées avec précision, par exemple : « Tu devrais réduire de 30% ton apport de gras, faire du sport 2 fois par semaine (pas plus, pas moins), et marcher exactement 80 minutes par jour.» Formulés de cette façon, les conseils ressemblent dangereusement à des ordres, qui peuvent être difficiles à entendre lorsque l'on a ni l'envie de faire un régime ni l'envie de faire du sport. Donner une formulation vague à un conseil peut donc être une stratégie visant à en atténuer le caractère intrusif.

Enfin, les conseils vagues peuvent servir une autre stratégie ne visant plus à ménager le décideur mais plutôt à promouvoir les intérêts propres du conseiller, tout en lui évitant de voir son honnêteté mise en doute. Assurément, votre conjoint aimerait que vous fassiez le plus de marche qu'il vous est possible de faire. Imaginons qu'il ait lu dans un magazine de santé que la marche devenait bénéfique à partir de trois quart d'heure de pratique. En vous conseillant de faire environ une heure de marche par jour, il assure ses propres intérêts (vous voir plus rapidement svelte) sans réellement être malhonnête ou vous trahir. De manière générale, être vague nous permet de communiquer la vérité qui nous arrange.

Toutes ces possibilités (le conseiller est incertain, le conseiller s'exprime avec tact, le conseiller promeut ses propres intérêts) rendent complexe et flexible l'interprétation d'un conseil vague. De plus, la personne qui interprète le conseil peut elle-même être en proie à des biais d'interprétation tel que le biais de désirabilité : puisque le conseil vague offre un large éventail d'interprétations possibles, autant choisir celle qui correspond le mieux à ses préférences personnelles. Si vous détestez marcher, vous êtes susceptible interpréter le conseil « Marche environ une heure par jour» comme signifiant 40 minutes plutôt que 80.

Dans les sections qui suivent, nous allons revenir plus en détail sur les paramètres que nous pensons déterminants pour l'utilisation de conseils vagues. Le premier est un facteur interpersonnel que nous appelons bienveillance (Section 2). Il reflète la croyance que le conseiller a sincèrement à cœur les intérêts du décideur. La bienveillance devrait donc limiter l'appréhension d'être induit en erreur par le conseiller et, influencer ainsi l'utilisation qui est faite du conseil. Dans la Section 3, nous nous tournerons vers deux autres paramètres, le format de l'imprécision et la personne qui détient les préférences éventuellement menacées par le conseil.

## 2 Bienveillance

Comme nous l'avons déjà évoqué, le conseiller peut être stratégiquement imprécis pour promouvoir ses intérêts propres, sans pour autant être objectivement malhonnête. Face à ce risque, le décideur dispose d'une solution : il peut rechercher les conseils des personnes en qui il a *confiance.* Le rôle de la confiance a souvent été étudié dans les situations de dilemmes sociaux (tels que le jeu du prisonnier), où le risque de tromperie est justement important. L'établissement d'une relation de confiance permet dans de telles situations d'augmenter la probabilité de coopération, en réduisant la tentation de tromper ou de faire défaut à l'autre [13].

La confiance est une notion complexe, du fait de sa multidimensionnalité [8, 10, 11, 12]. Toutes les dimensions de la confiance décrites dans la littérature ne sont pas pertinentes pour le problème traité dans cet article. L'expertise par exemple, est une dimension dont l'importance est largement reconnue ; pour avoir la confiance du décideur, le conseiller a besoin de maîtriser le domaine sur lequel portent les conseils. Mais même si la compétence du conseiller est déterminante, elle ne l'immunise pas contre l'envie de tromper l'autre pour assurer ses intérêts.

Une autre dimension de la confiance est plus spécifiquement liée au soupçon de tromperie : il s'agit de la bienveillance. C'est une qualité que nous attendons d'un conseiller pour pouvoir lui accorder notre confiance[11], et qui reflète son désir sincère d'agir au mieux

pour nous, en sacrifiant éventuellement ses intérêts propres : même si les intérêts du décideur et ceux du conseiller sont incompatibles, un conseiller bienveillant devrait donner le conseil optimal pour le décideur. Des études ont déjà montré l'influence que peut avoir la bienveillance sur l'utilisation du conseil précis. Pour manipuler la bienveillance, des auteurs ont utilisé les relations personnelles préexistantes entre conseiller et décideur [15], d'autres ont rendu explicites les pensées du décideur qui traduisaient sa perception du niveau de bienveillance du conseiller [17].

En parallèle, certaines émotions dites contingentes (dans le sens où elles sont provoquées par un événement qui n'est pas lié à la situation de conseil) peuvent affecter la confiance [6] : la colère contingente a tendance à diminuer la propension à la confiance alors que la gratitude contingente agit positivement sur la confiance. Colère et gratitude sont des émotions susceptibles d'être ressenties respectivement vis à vis de conseillers malveillants ou bienveillants. On peut alors imaginer que la colère et la gratitude fonctionnent comme des marqueurs somatiques de la bienveillance perçue, affectant donc la confiance ressentie vis à vis du conseiller et consécutivement l'usage qui sera fait de ses recommandations.

Des résultats antérieurs suggèrent donc que la bienveillance joue un rôle dans l'évaluation de la confiance que l'on peut accorder à des conseillers, et dans le fait de suivre ou non leurs conseils. Mais ces études ont été réalisées uniquement avec des recommandations précises. Notre analyse suggère que la bienveillance est encore plus importante dans les situations de conseils vagues, du fait du risque de tromperie qu'autorise l'imprécision. Nous nous attendons donc à ce que la bienveillance ait un effet critique sur l'utilisation de recommandations vagues. Cette hypothèse fera l'objet d'un test expérimental, rapporté dans la section empirique de cet article.

# 3 Préférences

Nous abordons maintenant un autre aspect des situations de conseils vagues, qui concerne le rôle joué par les préférences du décideur vis à vis des événements futurs. Deux perspectives théoriques permettent de rendre compte de ce rôle, l'une est basée sur la notion de tact et l'autre sur la notion de biais de désirabilité. Dans cette section, nous allons montrer comment ces deux perspectives aboutissent à des prédictions contrastées qui seront mises à l'épreuve dans la section empirique de l'article.

## 3.1 Tact

Un conseil est un acte menaçant pour la face (FTA : Face-threatening act) dans le sens de la théorie de la politesse [2, 3]. Un FTA est un énoncé qui peut menacer les deux aspects de la « face », que l'on peut définir comme l'identité publique positive que nous essayons de projeter et de maintenir dans nos interactions sociales. Le premier aspect de la face (la face négative) correspond au désir de montrer de l'indépendance dans nos décisions, de la liberté dans nos actions. Le second aspect de la face (la face positive) correspond au désir de voir nos préférences et jugements approuvés par les agents avec lesquels nous sommes en interaction.

Par nature, les conseils menaçent la face négative, dans le sens où leur but inhérent est d'influencer les actions d'autrui, diminuant ainsi l'indépendance du décideur dans sa prise de décision. Les conseils peuvent aussi menacer la face positive, s'ils recommandent des actions qui vont contre les préférences du décideur : formuler un conseil contrapréférentiel signifie au décideur que ses préférences spontanées ne sont pas celles qui semblent les plus appropriées vis à vis de la situation, du point de vue du conseiller. Comme l'ont expérimenté la plupart des parents, suggérer à un adolescent de réduire le temps qu'il passe devant les jeux vidéos provoque une réaction indignée, non seulement parce que cela enfreint son désir d'indépendance, mais également parce que ce conseil véhicule un jugement négatif à l'égard de ses préférences.

Un conseiller qui doit formuler un FTA (comme par exemple donner un conseil contrapréférentiel) est susceptible, voire même contraint par les attentes sociales, d'avoir recours à une stratégie de politesse dans le but de ménager la face de l'interlocuteur. Parmi les stratégies disponibles, l'une consiste à édulcorer la recommandation menaçante ; pour cela, le conseiller doit utiliser une formulation suffisamment précise pour qu'elle traduise sa pensée et conserve son caractère informatif, mais suffisamment imprécise pour ne pas désapprouver ouvertement les préférences du décideur. Reprenons l'exemple des jeux vidéos. Au lieu de déclarer précisément « Tu pourrais passer deux fois moins de temps

à jouer », un parent empreint de tact se montrera plus vague pour réduire la menace de face : « Tu pourrais passer un peu moins de temps à jouer ». Bien entendu, l'adolescent n'est pas sans connaître les stratégies de politesse communes, et possède les compétences sociales nécessaires pour comprendre que le parent a minimisé sa pensée et donc qu'«un peu moins » peut signifier beaucoup. Les décideurs pourraient donc percevoir l'imprécision d'un conseil contrapréférentiel comme une marque de tact, et par conséquent les surinterpréter, c'est à dire ajuster l'interprétation en en exagérant le caractère contrapréférentiel.

Un phénomène de cet ordre a été mis en évidence dans le domaine de la communication médicale [1]. Les participants devaient évaluer la probabilité associée à un prognosic médical qualifié de « possible » par le médecin. Les résultats montrent que les pronostics sévères ont reçu des probabilités plus hautes que les prognostics plus anodins, bien que les deux aient été décrits comme « possible ». Cet effet s'explique par le fait que de nombreux participants ont perçu le mot « possible » comme une marque de tact dans le cas des prognostics sévères (« Il est possible que vous deveniez sourd »), mais pas dans le cas de prognostics bénins (« Il est possible que vous développiez des insomnies »).

Dans cet article, nous nous intéressons à des recommandations vagues qui sont des actes menaçants : tout d'abord, parce que ce sont des recommandations qui menacent par nature la face négative ; mais également, parce qu'elles sont contrapréférentielles et menacent donc la face positive. Toutefois, ces recommandations menaçantes doivent réunir deux conditions nécessaires pour que le décideur puisse soupçonner que le conseiller a eu recours au tact et pour qu'il ajuste ainsi son interprétation. Premièrement, la recommandation doit utiliser une imprécision verbale plutôt que numérique. En effet, alors que les quantificateurs vagues verbaux comme « possible » ou « un peu moins » sont notoirement impliqués dans des stratégies de politesse, rien ne nous permet de suggérer que des expressions vagues numériques comme « 10 à 20% de chance » ou « 10 à 20% de moins » le sont aussi.

Deuxièmement, le conseil doit menacer la face positive du décideur et non celle d'un autre agent : la recommandation doit être contrapréférentielle pour le décideur mais pas pour d'autres agents. Pour illustrer cette variable que nous appelons *le détenteur des préférences*, supposons que la préférence d'un étudiant est généralement d'écrire une thèse courte. Imaginons maintenant que nous disions à un étudiant que sa thèse devrait être un peu plus longue qu'elle ne l'est actuellement. Dans cette situation, notre recommandation va contre les préférences de la personne que nous conseillons : cette personne est détentrice des préférences. Comparons maintenant cette situation avec celle où nous disons au directeur de thèse de l'étudiant que la thèse devrait être un peu plus longue. Ici, notre recommandation ne va pas contre les préférences de la personne conseillée (le directeur), mais contre celle d'un autre agent, l'étudiant : la personne conseillée n'est pas détentrice des préférences. Selon la théorie de la politesse, la surinterprétation ne devrait pas avoir lieu dans ce deuxième cas.

Pour résumer, la théorie de la politesse prédit une interaction entre le détenteur de la préférence (les préférences en jeu sont-elles celles de la personne conseillée, ou celles d'un autre agent ?) et le format de l'imprécision (verbal ou numérique) : l'utilisation de la recommandation contrapréférentielle devrait être plus importante quand les préférences en jeu sont celles de la personne conseillée et quand la recommandation est verbale. Le biais de désirabilité, que nous allons maintenant aborder, fait une prédiction exactement opposée.

### 3.2 Biais de désirabilité

Ce que l'on désigne en anglais comme le *wishful thinking* est la tendance à évaluer la probabilité d'un événement futur en fonction de notre préférence concernant cet événement. On parlera également de *biais de désirabilité* pour désigner ce comportement [9]. Le biais de désirabilité peut influencer l'interprétation de probabilités verbales vagues dans la direction opposée à celle de la perception de tact [14]. Les expressions verbales décrivant la probabilité d'un événement contrapréférentiel sont sous-interprétés quand les participants sont concernés par l'événement négatif (c'est à dire, quand leurs préférences sont menacées). Par exemple, les participants à qui l'on demande de traduire numériquement le mot « probablement » dans la phrase « Tu vas probablement divorcer » donnent une probabilité plus faible que quand ce même mot est présenté dans l'énoncé : « [un autre étudiant] va probablement divorcer. »

À l'inverse, les expressions verbales décrivant la probabilié d'un événement souhaité sont surinterprétées quand les participants sont les détenteurs des préférences. Par exemple, les participants à qui on demande de donner une interprétation numérique du mot « probablement » dans « Tu vas probablement avoir ta licence avec mention » donnent une probabilité plus forte que lorsque l'énoncé est « [un autre étudiant] va probablement avoir sa licence avec mention. »

Ainsi, le biais de désirabilité en tant que force motivationnelle prédit que les recommandations vagues contrapréférentielles devraient être sous-interprétées quand la personne conseillée est détentrice des préférences, mais pas quand c'est un autre agent qui l'est. Cependant, comme pour le tact, nous pouvons supposer que le format verbal est nécessaire pour autoriser le biais de désirabilité. En effet, à notre connaissance, le biais de désirabilité n'a jamais été observé avec des informations numériques vagues comme des intervalles de valeurs.

Pour résumer, le biais de désirabilité prédit une interaction entre le détenteur de la préférence (les préférences en jeu sont-elles celles de la personne conseillée, ou celles d'un autre agent ?) et le format de l'imprécision (verbal ou numérique) ; mais dans la direction opposée à celle prédite par la théorie de la politesse : l'utilisation du conseil contrapréférentiel devrait être plus faible quand les préférences en jeu sont celles de la personne conseillée et lorsque la recommandation est verbale.

# 4 Méthode

Nous allons maintenant rapporter une expérience visant à identifier le meilleur modèle de l'impact de nos trois variables sur l'interprétation d'un conseil vague. Premièrement, nous souhaitons tester l'hypothèse selon laquelle la bienveillance a un effet déterminant sur l'utilisation de conseils vagues (contrapréférentiels). Deuxièmement, nous souhaitons déterminer lequel du biais de désirabilité ou du tact est la principale force motivationnelle influençant l'utilisation de recommandations vagues et contrapréférentielles. Pour cela, nous manipulons dans cette expérience le format de l'imprécision (verbal vs numérique) d'une recommandation contrapréférentielle, la bienveillance attendue de la part du conseiller ainsi que le détenteur des préférences vis à vis du conseil. Nous évaluons l'utilisation du conseil en mesurant la révision que font les participants de leur opinion initiale après avoir reçu la recommandation vague. Nous nous attendons à un effet indépendant de la bienveillance, et à un effet d'interaction entre le format de l'imprécision et le détenteur de la préférence, dont la direction devrait nous permettre de départager les prédictions faites par la théorie de la politesse et le biais de désirabilité.

## 4.1 Participants

Un total de 120 étudiants de l'Université Toulouse le Mirail a participé à cette étude : 41 hommes et 79 femmes. La moyenne d'âge est de 23 ans (écart-type = 4,7). Les données de 6 participants ont été exclues sur la base de leur distance de Mahalanobis avec le reste de l'échantillon. (La distance de Mahalanobis permet de repérer les vecteurs de réponses extrêmes afin de les écarter de l'échantillon. Une grande distance de Mahalanobis est l'équivalent multivarié d'être, par exemple, à plus de trois écart-types de la moyenne d'une distribution.)

## 4.2 Matériel et Procédure

L'expérience est construite sur un plan expérimental mixte $2 \times 2 \times 2$. Le format de l'imprécision (verbal vs. numerique) et le détenteur des préférences (personne conseillée vs. autre) sont manipulées en inter-groupes (la valeur de ces variables est différentes dans différents sous-échantillons de l'échantillon expérimental). La bienveillance du conseiller est une variable intra-sujet (chaque participant à l'expérience sera confronté tour à tour aux deux valeurs de cette variable). L'expérience consiste en la passation d'un questionnaire proposant un scénario présenté ci-dessous :

> Vous êtes entraîneur de football de l'équipe de Valenciennes. Cette année, votre équipe est montée en ligue 1. Vous comptez de ce fait demander une augmentation pour un joueur évoluant dans votre équipe depuis un an. Il gagne actuellement 20 000 euros par mois. Quel est le montant de l'augmentation que vous comptez demander ?

Cette première partie manipule en intersujet la variable *détention des préférences*. Dans ce groupe expérimental, l'augmentation concerne une personne autre que le décideur. Dans l'autre groupe expérimental, l'augmentation concerne

l'entraîneur lui-même. Notons $e_1$ la première estimation de l'augmentation.

Un conseil recommandait ensuite aux participants de diminuer le montant qu'ils comptaient demander. Cette recommandation provenait soit d'un conseiller bienveillant (avec lequel ils étaient supposés avoir « d'excellentes relations »), soit d'un conseiller non bienveillant (avec lequel ils étaient supposés avoir « des relations concflictuelles »). La recommandation était formulée soit verbalement (« Á mon avis, tu devrais demander *un peu moins* ») soit numériquement (« Á mon avis, tu devrais demander *10 à 30% de moins*). L'exemple suivant correspond à une recommandation formulée verbalement par un conseiller bienveillant :

> Juste avant d'aller solliciter cette augmentation, vous croisez un collègue avec qui vous avez toujours eu *excellentes relations*. Vous vous mettez à discuter. La conversation se porte sur l'augmentation de votre joueur. Vous demandez à votre collègue ce qu'il pense du montant de l'augmentation que vous allez demander. Il vous répond : « Á mon avis, tu devrais demander un peu moins ». Compte tenu de cette opinion, quel est le montant en euros, de l'augmentation que vous comptez demander ?

Notons $e_2$ cette estimation révisée. À partir de $e_1$ et $e_2$, nous obtenons un taux de révision de la décision défini comme $\frac{e_1-e_2}{e_1}$.

**Prétest** Nous avons utilisé la recommandation « 10 à 30% de moins » comme la forme numérique de l'expression verbale « un peu moins ». Cette décision se base sur un prétest que nous avons réalisé auprès d'un autre groupe de participants, qui devaient traduire numériquement l'expression « un peu moins » dans un même contexte que celui utilisé dans l'expérience, à savoir celui d'une augmentation salariale sur la base d'un salaire de 20000 euros.

Les participants au prétest ont du répondre à la question suivante :

> Deux personnes gagnent un salaire de 20 000 euros par mois chacune. Elles reçoivent toutes les deux une augmentation. L'une reçoit une augmentation de $[x_1]$ euros ($[x_1]$ euros de plus par mois). L'autre reçoit un peu moins. Selon vous, quel peut- être le montant de l'augmentation de la deuxième personne ?

Le montant $x_1$ prend différentes valeurs selon la condition expérimentale : 1000, 2000, 3000 ou 4000 euros. L'augmentation de la deuxième personne $x_2$ est estimée par le sujet.

Soit r le rapport du montant de la deuxième augmentation sur le montant de la première :

$$r = \frac{x_2}{x_1}$$

L'expression « un peu moins » est traduite par 18% de moins en moyenne. Comme le montre la Figure 1, la plupart des réponses (plus de 70%) se situent dans un intervalle allant de 10 à 30%, quelque soit la valeur initiale $x_1$. Nous avons donc considéré l'expression « 10 à 30% de moins » comme étant un équivalent acceptable de l'expression « un peu moins » dans le contexte d'une augmentation salariale.

## 5 Résultats

Les données ont été analysées à l'aide d'une analyse de variance à mesures répétées utilisant comme variable dépendante le rapport $\frac{e_1-e_2}{e_1}$. La Table 1 présente les statistiques descriptives, dont la Figure 2 offre une représentation graphique.

Comme nous nous y attendions, la bienveillance a un effet massif sur l'utilisation du conseil : $F(1, 113) = 53.5$, $p < .001$, $\eta^2_{sp} = .10$. Les conseils donnés par des conseillers bienveillants conduisent les participants à diminuer leur estimation initiale de 19% en moyenne (écart-type = 18 points de pourcentage). Cette valeur correspond quasi-parfaitement à ce que le prétest identifiait comme étant la traduction numérique de l'expression « un peu moins ». À l'inverse, quand le conseiller n'est pas bienveillant, les participants écartent presque complètement la recommandation, avec une diminution moyenne de la première estimation de l'ordre de 6% (écart-type = 22 points de pourcentage) ; pour être plus précis, la diminution moyenne est inférieure à 5% dans 3 des 4 conditions sans bienveillance. Ces résultats montrent sans ambiguité le rôle critique que joue la bienveillance dans l'utilisation de conseils vagues.

Ni le format de l'imprécision ni le détenteur de la préférence n'ont eu d'effet principal. Cependant, l'interaction de ces deux variables affecte l'utilisation du conseil : $F(3, 113) = 3.8$,$p = .05$, $\eta^2_{sp} = .03$. L'effet obtenu est celui prédit par le biais de désirabilité. Les recommandations reçoivent un poids plus important quand elles vont à l'encontre des préférences d'une tierce personne, plutôt qu'à l'encontre de celles du déci-

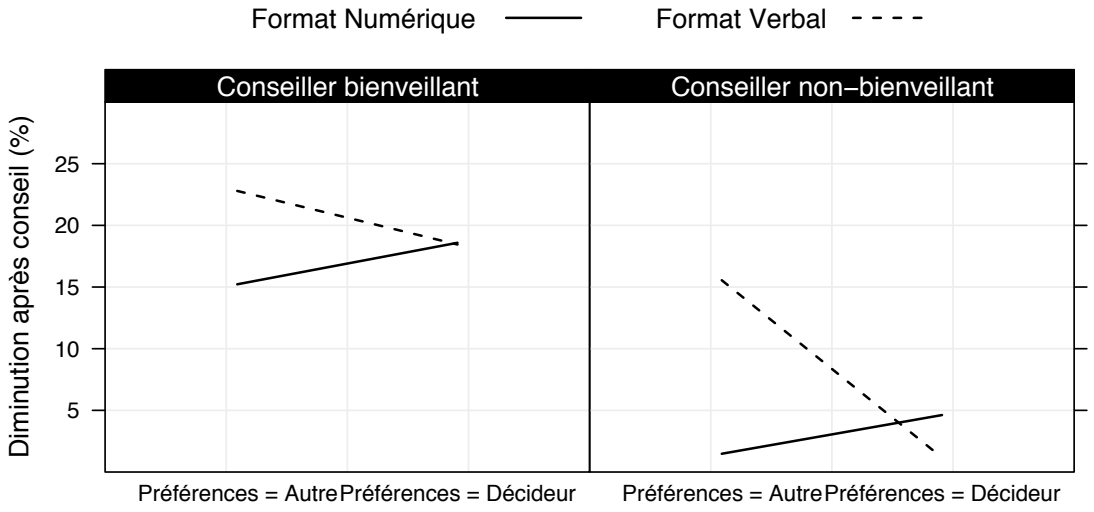



FIGURE 2 – Révision moyenne (en points de pourcentage) après avoir reçu un conseil vague contrapréférentiel. Le graphique de gauche présente les données obtenues avec la recommandation d'un conseiller bienveillant, le graphique de droite présente celles obtenues avec un conseil provenant d'un conseiller non bienveillant. Sur les deux graphiques, l'interaction entre le format de l'imprécision et le détenteur des préférences est claire. La variable détenteur des préférences n'a d'effet que lorsque le conseil vague est exprimé verbalement (« un peu moins »). Dans ce cas, la recommandation a l'impact le plus fort lorsque les préférences menacées ne sont pas celles de la personne conseillée.

TABLE 1 – Révision moyenne (points de pourcentage) après avoir reçu un conseil vague contrapréférentiel, en fonction de la bienveillance du conseiller, du format de l'imprécision et du détenteur des préférences (décideur ou autre). Les écarts-types figurent entre parenthèses.

| | Conseiller bienveillant | | Conseiller non bienveillant | |
|---|---|---|---|---|
| | Format verbal | Format numérique | Format verbal | Format numérique |
| Personne conseillée | 19 (18) | 19 (16) | 01 (24) | 05 (16) |
| Autre agent | 23 (18) | 15 (19) | 16 (22) | 01 (23) |

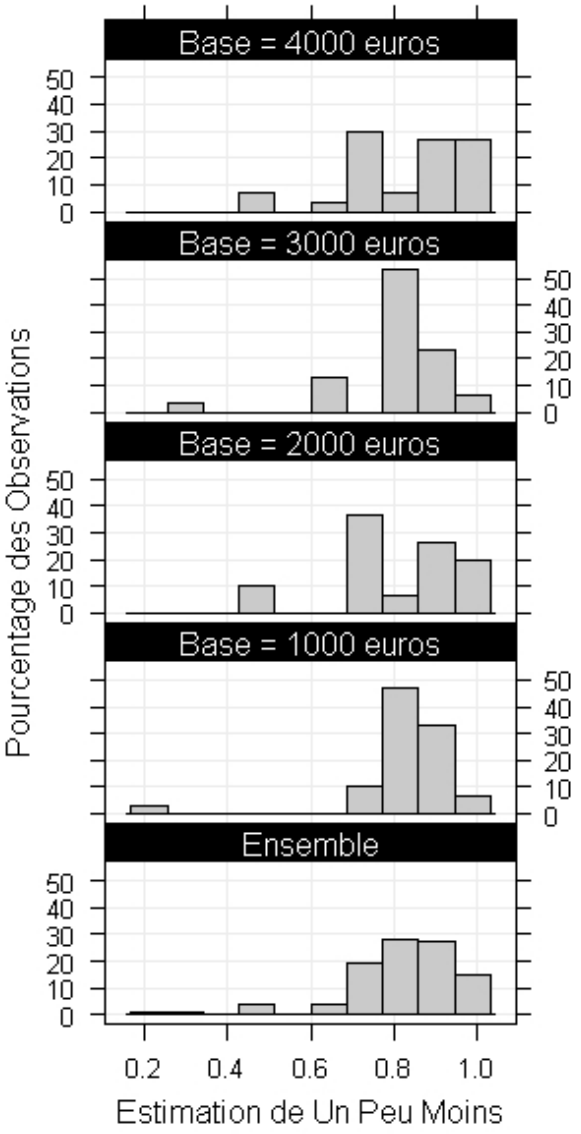



FIGURE 1 – Distribution des ratios $r$ pour l'interprétation numérique de l'expression « un peu moins » utilisée dans un contexte d'augmentation salariale et pour chacune des valeurs de $x_1$. La plupart des réponses se situent dans l'intervalle [.70 ;.90], ce qui signifie que un peu moins que $e_1$ correspond en moyenne à une valeur de $e_2$ comprise entre 70 et 90% de $e_1$. Un peu moins que $e_1$ signifie en fait réduire $e_1$ de 10 à 30%.

deur, mais cet effet nécessite que la recommandation soit exprimée verbalement et non avec un intervalle de valeurs.

## 6 Conclusion

Au début de cet article, nous avons examiné les multiples raisons qui expliquent l'utilisation répandue de conseils vagues dans la vie de tous les jours. Nous avons suggéré que l'imprécision pouvait permettre au conseiller d'induire stratégiquement en erreur le décideur sans pour autant être objectivement malhonnête. Pour éviter ce problème, les décideurs sont enclins à aller rechercher des conseils auprès de personnes en qui ils ont confiance et plus précisément, qu'ils pensent bienveillantes. En se basant sur les travaux antérieurs démontrant l'effet de la bienveillance sur l'utilisation du conseil précis, nous avons supposé que le rôle de la bienveillance serait critique pour les recommandations vagues, du fait de ce risque de tromperie.

En plus de ce facteur interpersonnel, nous avons envisagé les forces motivationnelles qui sous-tendent l'utilisation de recommandations vagues. Notre analyse a permis de confronter deux modèles possibles. Selon la théorie de la politesse, les conseillers cherchent à maintenir la face du décideur quand ils sont amenés à donner des recommandations contrapréférentielles, les décideurs sont conscients de cette stratégie de tact et ils ont par conséquence tendance à surinterpréter les recommandations contrapréférentielles quand les préférences en jeu sont les leurs. Selon le biais de désirabilité, les décideurs choisissent l'interprétation la moins dommageable vis à vis de leurs propres préférences, et sous-interprètent donc les recommandations contrapréférentielles quand les préférences en jeu sont les leurs.

Le biais de désirabilité offre la meilleure explication des résultats de notre expérience. Ces résultats prolongent certaines observations sur l'interprétation des propriétés verbales [14], mais sont en contradiction avec d'autres résultats indiquant une surinterprétation des termes vagues ou ambigus en situations contrapréférentielles [4, 1, 7].

Ces données divergentes nous amènent à nous demander pourquoi la tension entre le tact et le biais de désirabilité se résout différemment selon les études. De futures recherches auront à expliciter les paramètres individuels et situationnels qui favorisent une interprétation plutôt

qu'une autre. Quelques modifications mineures à notre manipulation expérimentale pourraient suffir à favoriser l'interprétation en termes de tact. Par exemple, rendre le conseil non sollicité ou public peut favoriser la perception de tact. Dans notre expérience, le conseil était à la fois sollicité et privé. La conjonction de ces deux facteurs peut avoir encouragé les participants à juger les considérations de tact comme non pertinentes, parce que (a) une sollicitation peut être une invitation à donner une réponse directe et claire, même si elle s'avère contrapréférentielle ; et (b) la menace de face d'une recommandation contrapréférentielle peut être largement atténuée lorsqu'elle n'est pas donnée publiquement. En parallèle des facteurs situationnels, des caractéristiques individuelles peuvent encourager les participants à privilégier une interprétation plutôt qu'une autre. Par exemple, les participants sociables peuvent être plus enclin à détecter l'usage de tact chez leur interlocuteur, alors que des participants présentant un optimisme irréaliste peuvent être plutôt sujets au biais de désirabilité. La sociabilité comme l'optimise irréaliste sont des traits de caractère qui peuvent être mesurés par des tests de personnalité familiers aux psychologues. De telles mesures pourraient avec profit être introduites dans une étude expérimentale telle que celle que nous avons menée.

Nos résultats concernant la bienveillance sont clairs. La bienveillance est une condition nécessaire à l'utilisation d'un conseil vague contrapréférentiel, en particulier lorsque les préférences en jeu sont celles de la personne conseillée. Ce résultat invite certains raffinements méthodologiques et théoriques que nous nous proposons d'explorer dans la suite de notre programme de recherche.

Premièrement, rappelons que nous avons utilisé la qualité de la relation entre conseiller et décideur pour opérationnaliser la bienveillance. Bien que cette opérationnalisation ait déjà été utilisée [15], cette manipulation ne coincide pas parfaitement avec la définition de confiance que nous utilisons [11]. Pour nous rapprocher de cette définition, nous projetons d'utiliser des situations de conflit d'intérêt entre le conseiller et la personne conseillée. La bienveillance sera opérationnalisée comme le degré auquel le conseiller se montre prêt à favoriser les intérêts de la personne conseillée aux dépens des siens. Cette procédure facilitera l'étude de la dynamique de la bienveillance perçue au cours d'une série de situations de conseil et plus précisément, du temps nécessaire pour se contruire une réputation de bienveillance, pour la détruire, et ultimement pour la réparer.

Deuxièmement, nous sommes conscients qu'avoir de bonnes relations avec un conseiller implique, en dehors de toutes considérations de bienveillance, une sorte d'engagement, celui de les maintenir. Le conseil d'un ami pourrait donc être davantage suivi, non pas parce qu'il est perçu comme plus fiable, mais plutôt parce que le rejeter reviendrait à remettre en cause l'amitié entre le conseiller et la personne qui reçoit le conseil. La prochaine étape de notre programme de recherche sera de contrôler cette possibilité, en donnant une information supplémentaire à la personne conseillée : lui préciser si le conseiller sera informé ou non de l'emploi de son conseil.

Enfin, en dehors de ces prolongements empiriques, une tâche reste à aborder, qui consiste à dépasser la modélisation purement statistique que nous avons employée dans cet article, et de développer une modélisation formelle de la situation d'interaction que constitue la réception d'une recommandation vague. Les paramètres que nous avons identifiés, ainsi que la structure de leurs interactions, nous semblent ouvrir une voie prometteuse à une telle formalisation.

## Références

[1] J. F. Bonnefon and G. Villejoubert. Tactful or doubtful ? expectations of politeness explain the severity bias in the interpretation of probability phrases. *Psychological Science*, 17 :747–751, 2006.

[2] P. Brown and S. Levinson. *Universals in language usage : Politeness phenomena*, pages 56–289. Cambridge University Press, Cambridge, 1978.

[3] P. Brown and S. Levinson. *Politeness : Some Universals in Language Usage*. Cambridge University Press, 1987.

[4] V. Demeure, J. F. Bonnefon, and E. Raufaste. Politeness and conditional reasoning : Interpersonal cues to the indirect suppression of deductive inferences. *Journal of Experimental Psychology : Learning, Memory, Cognition*, 35 :260–266, 2009.

[5] I. Erev and B. L. Cohen. Verbal versus numerical probabilities : Efficiency, biases, and the preference paradox. *Organizatio-*

*nal Behavior and Human Decision Processes*, 45 :1–18, 1990.

[6] F. Gino and M. E. Schweitzer. Blinded by anger or feeling the love : How emotions influence advice taking. *Journal of Applied Psychology*, 93 :1165–1173, 2008.

[7] T. Holtgraves. Social psychology, cognitive psychology, and linguistic politeness. *Journal of Politeness Research*, 1 :73–93, 2005.

[8] C. E. Johnson-George and W. C. Swap. Measurement of specific interpersonal trust : Construction and validation of a scale to assess trust in a specific other. *Journal of Personality and Social Psychology*, 43 :1306–1317, 1982.

[9] Z. Krizan and P. D. Windschiltl. The influence of outcome desirability on optimism. *Psychological Bulletin*, 133 :95–121, 2007.

[10] J. D. Lewis and A. Weigert. Trust as social reality. *Social Forces*, 63 :967–985, 1985.

[11] R. C. Mayer, J. H. Davis, and F. D. Schoorman. An integrative model of organizational trust. *Academy of Management Review*, 20 :709–734, 1995.

[12] J. K. Rempel, J. G. Holmes, and M. D. Zanna. Trust in close relationships. *Journal of Personality and Social Psychology*, 49 :95–112, 1985.

[13] K. W. Rockmann and G. B. Northcraft. To be or not to be trusted : The influence of media richness on defection and deception. *Academy of Management Review*, 107 :106–122, 2008.

[14] T. Smits and V. Hoorens. How probable is "probably" ? it depends on whom you're talking about. *Journal of Behavioral Decision Making*, 18 :83–96, 2005.

[15] L. M. Van Swol and J. A. Sniezek. Trust, confidence and expertise in a judge-advisor system. *Organizational Behavior and Human Decision Processes*, 84 :288–307, 2001.

[16] T. S. Wallsten, D. V. Budescu, R. Zwick, and S. M. Kemp. Preference and reasons for communicating probabilistic information in numerical or verbal terms. *Bulletin of the Psychonomic Society*, 31 :135–138, 1993.

[17] T. B. White. Consumer trust and advice acceptance : The moderating roles of benevolence, expertise, and negative emotions. *Journal of Consumer Psychology*, 15 :141–148, 2005.

# On Agent Programming Language Support for Rational Communication

Yves Lespérance
lesperan@cse.yorku.ca

Steven Shapiro
steven@cs.toronto.edu

Dept. of Computer Science and Engineering, York University, Toronto, ON M3J 1P3, Canada

**Résumé :**
It is generally thought that agents in a multiagent system should be capable of communicating in a high-level, speech acts-based, agent communication language (ACL), for instance FIPA ACL. Many agent programming languages (APLs) such as Jack, JASON, JADE, etc., support communication in such ACLs, i.e., the production and reception/processing of ACL messages. They also support the specification of complex behaviors for agents, typically by allowing the definition of a library of hierarchical plans that are selected and executed at run time based on external or internal events. However, rational communication requires more than the ability to send and receive ACL messages ; the agent should understand the semantics and pragmatics of the associated communication acts. Without this, agents can only interact when they follow rigid protocols, which is difficult in open systems. Most APLs do not support any of this. In this paper, we argue that this is a serious deficiency that needs to be addressed and discuss the requirements for this. There is one tool that does support the processing of FIPA ACL message semantics and pragmatics, the JADE Semantic Add-On (JSA). But, JSA provides only limited support for the specification of complex agent behaviors. To gain a better understanding of the problem, we explore how one could combine JSA-like semantics and pragmatics processing capabilities with the complex behavior specification capabilities of a typical APL.

## 1 Introduction and Motivation

It is generally thought that agents in a multiagent system should be capable of communicating in a high-level, speech acts-based, agent communication language (ACL), for instance the ACL standardized by the Foundation for Intelligent Physical Agents (FIPA) [9] or KQML [11]. This allows agents not only to query/inform each other about what they know and request the performance of actions, but to make commitments to one another by making proposals and agreeing to a proposal or request. Such languages have a communication act layer separate from the content layer that can be used to make the illocutionary force (performative) of a message explicit. This allows some processing of messages, for example by a broker, without requiring detailed understanding of their content. FIPA ACL provides a semantic specification of the meaning of the available communication acts in the FIPA Semantic Language (SL), which is essentially a modal logic dealing with beliefs, intentions, and action.

Many frameworks for programming multiagent systems support the production, reception and processing of FIPA ACL messages, for instance Jack [5], JASON [2], JADE [1], IndiGolog [7] with the IG-JADE-PKSlib library [19], etc. However in nearly all cases, these tools process messages only in terms of their syntax and do not support the semantic interpretation of messages and reasoning about their illocutionary/rational effects and the associated agent beliefs and intentions. In fact, agents are generally assumed to interact only according to specified protocols (e.g., contract net, query-reply, etc.) where the types of performatives that may be exchanged in a conversation are rigidly specified (many such interaction protocols have also been standardized by FIPA). The agents only accept messages with the performatives (and content languages/ontologies) allowed by the protocol, and process them based on their performative label and content, without access to their semantics. As pointed out in [17, 18], this means that agents can only interact when they have been designed to interact and that it is very difficult to get agents in an open system to work together.

To solve this problem, one needs agent programming facilities that support rational communication, i.e., the semantic/pragmatic interpretation of FIPA ACL messages as well as reasoning over the resulting semantic representations. This would allow agents to understand the meaning/pragmatics of the messages they receive and to respond to them in a reasonable way, even when messages do not come according to a rigid protocol. Support for semantic/pragmatic interpretation of messages also has many other benefits. There are many ways to perform the same communication act, e.g., one can make an implicit request by informing someone that one has a particular intention. An agent that really understands the meaning of messages will be able to see beyond the difference in surface form. When applying cooperation principles, an agent with rational communication capabilities can exploit its understanding of the pragmatics of the exchanged messages.

For instance, it can detect lying/deception when receiving messages whose claims are inconsistent with what is known about the sender's beliefs/intentions. Furthermore, an agent with rational communication capabilities can in principle deal with various forms of communication failure. When it detects a false presupposition in a message, it can correct it. If a question or request is unclear, it can ask for a clarification. When a direct answer to a query is unlikely to achieve the ultimate goal of the requester, it can provide a helpful answer addressing the latter. One can easily imagine this sort of capabilities being very advantageous/profitable in an e-commerce agent for example.

There is one tool that attempts to provide the kind of rational communication facilities discussed above, the JADE Semantic Add-On (JSA) [17, 18]. JSA provides a library for manipulating FIPA ACL messages, and FIPA SL representations of the semantics of these messages. Semantic interpretation and pragmatic inference, including cooperative belief/intention adoption, is performed in a rule-based framework that can be customized. Facilities are also provided for storing SL semantic representations in a knowledge base and defining forward chaining and backward chaining inference rules over them.

Another requirement for agent programming tools is support for the specification of complex behaviors. The main approach to this is belief-desire-intention (BDI) agent programming languages/architectures, such as PRS [10] and its various successors, such as AgentSpeak [23], JASON [2], Jack [5], JAM [14], and CAN [30, 28], as well as the closely related 3APL [13]. These BDI agent programming languages were conceived as a simplified and operationalized version of the BDI (Belief, Desire, Intention) model of agency, which is rooted in philosophical work such as Bratman's [4] theory of practical reasoning and Dennett's theory of intentional systems [8].

An important feature of BDI-style programming languages and platforms is their interleaved account of sensing, deliberation, and execution [22]. In BDI systems, *abstract plans* written by programmers are combined and executed in real-time. By executing as they reason, BDI agents reduce the likelihood that decisions will be made on the basis of outdated beliefs and remain responsive to the environment by making adjustments in the steps chosen as they proceed. Unlike in classical planning-based architectures, *execution* happens at each step. The assumption is that the careful crafting of plans' preconditions to ensure the selection of appropriate plans at execution time, together with a built-in mechanism for retrying alternative options, will usually ensure that a successful execution is found, even in the context of a changing environment. There is also work on using learning to improve the way plans are selected [12, 15].

Another popular approach to agent programming is the Golog [16] family of high-level programming languages. In contrast to BDI APLs, programming languages in the Golog line aim for a middle ground between classical planning and normal programming. The idea is that the programmer may write a *sketchy* nondeterministic program involving domain specific actions and test conditions and that the interpreter will reason about these and search for a valid execution. The semantics of these languages is defined on top of the *situation calculus*, a popular predicate logic framework for reasoning about action [20, 25]. The interpreter for the language uses an action theory representing the agent's beliefs about the state of the environment and the preconditions and effects of the actions to find a provably correct execution of the program. By controlling the amount of nondeterminism in the program, the high-level program execution task can be made as hard as classical planning or as easy as deterministic program execution. In ConGolog [6], the language was extended to support concurrent programming. Finally in IndiGolog [7], the framework was generalized to allow the programmer to control planning/lookahead and support online execution and sensing the environment.

There are also other approaches to agent programming languages, for instance approaches based on process algebra or temporal logic. Bordini et al. [3] discusses many of these.

Clearly, it would be very useful to have an agent programming framework that provides both (JSA-like) communication and semantics/pragmatics processing capabilities as well as the complex behavior specification capabilities of a typical BDI or Golog-style agent programming language. However, developing such a tool presents very significant challenges. To be able to represent the illocutionary/perlocutionary effects of communication acts, one needs a very expressive logical framework with quantifiers and modal op-

erators for beliefs, intentions, time/action, etc. (e.g., FIPA SL). Entailment in such a logic is undecidable. Even if we restrict attention to the propositional fragment, deciding entailment has very high complexity. Thus using a complete reasoner for such a logic is incompatible with the reactivity requirements of agents operating in highly dynamic environments. One possible approach is to try to identify fragments of such logics where reasoning is decidable and practically feasible, along the lines of work on description logics. But as far as we know, no formalism of this type has ever been used to model the effects of a wide range of communication acts. An alternative approach is to use an incomplete set of inference rules and/or search control heuristics in performing inference. This means that the user must assume responsibility for the range of inferences handled and must often craft customized inference rules/proof strategies for his application. One example of this approach is the ARTIMIS [26] rational agent framework, where a modal logic theorem prover with customized proof strategies is used to implement cooperative dialogue systems (many elements of the ARTIMIS framework have been incorporated into JSA).

To get a better understanding of the issues involved in developing an agent programming framework that supports complex behavior specification facilities and rational communication, we examine in this preliminary project report how one could go about combining JSA with a typical BDI or Golog-style agent programming language. In the next section, we go over JSA and IndiGolog in a bit more detail. Then, we outline a generic architecture that combines JSA and an APL. We keep it generic so that it remains applicable to many different APLs. After that, we examine how the model could be realized with IndiGolog as the APL. We conclude by summarizing our results so far and discussing the work that remains.

## 2 Background

### 2.1 JSA

As mentioned earlier, agent communication languages (ACLs) are languages shared between agents that are used to formulate messages sent between them. In this paper, we focus on FIPA ACL. A FIPA ACL message contains a performative and may (and usually does) include other parameters, such as a sender, a receiver, and the content of the message. The performative indicates what the message is intended to achieve. Examples of performative are : Inform, Request, Inform Ref (which informs the recipient of the referent of a descriptor), Call for Proposal, Agree, Cancel, etc. Examples of FIPA ACL messages are given below.

FIPA ACL identifies a set of primitive and composite communication actions (such as Inform and Request) and gives them a semantics using a first-order, multimodal, multiagent logic called the FIPA Semantic Language (FIPA SL). FIPA SL has the following modalities :
- $(B\ i\ \phi)$ : agent $i$ believes $\phi$
- $(U\ i\ \phi)$ : agent $i$ is uncertain whether $\phi$ holds
- $(I\ i\ \phi)$ : agent $i$ intends $\phi$
- $(\textit{done}\ a\ \phi)$ : action $a$ has just occurred and $\phi$ held beforehand
- $(\textit{feasible}\ a\ \phi)$ : action $a$ can possibly occur and $\phi$ would hold if it did

The semantics of communicative acts in FIPA ACL are defined using *feasibility conditions* and *rational effects*. A feasibility condition specifies the condition that must hold for an act to be performed. A rational effect is the effect an agent intends to bring about by performing the act.

The main activity of a JSA agent is interpreting incoming FIPA ACL messages. The interpretation loop has access to a belief base which stores facts and maintains a history of previously executed actions (using FIPA SL as its representation language), and a planner. There are two main processes in the JSA interpretation loop, one is a *production function* and the other is a *consumption function*. The objects that these functions operate over (i.e., produce or consume) are FIPA SL sentences which are called *semantic representations* (SRs). When a message comes in, the production function reads the message and produces SRs according to the type and content of the message. The consumption function reads these SRs and acts on these SRs by, e.g., asserting beliefs, executing actions, and/or producing more SRs.

The production and consumption functions are implemented using *semantic interpretation principles* (SIPs). The SIPs are rules that operationalize the FIPA ACL semantics (via predefined SIPs) but can also be used to customize the behavior of agents in a particular domain, implement planning or complex activities, etc. (via user-defined SIPs).

As an example of the interpretation process

[17], suppose a *son* agent wants to know the current temperature from a *display* agent. There are different ways to formulate this request in FIPA ACL. One way is for the son to inform the display that it wants to know the current temperature. This is expressed by the following action which we denote *Act1*.

(*inform :sender son :receiver* (*:set display*)
*:content* ("(*I son* (*exists* ?*t*
(*B son* (*temperature* ?*t*)))))").

This is an inform action from the son to a set of recipients, which in this case is just the display agent. The content of the action is that the son intends that there be a $t$ such that the son believes that $t$ is the current temperature. When the display agent receives this message, the fact that the son agent sent the message is asserted in the belief base of the display agent :

(*B display* (*done* (*action son Act1*) *true*)).

This belief triggers the ActionFeature SIP which asserts (among other beliefs) the belief in the rational effect of the action. In this case, the rational effect is that the son intends that the display believes that the son intends to know the temperature.

(*B display* (*I son*
(*B display* (*I son*
(*exists* ?*t* (*B son* (*temperature* ?*t*)))))))).

A belief of this form, i.e., that the agent believes that another agent intends the agent to believe something, triggers the BeliefTransfer SIP. This is a user-customizable SIP which when fired causes the agent to adopt a belief of another agent. In this case, it causes the display to adopt the belief intended by the son :

(*B display* (*I son*
(*exists* ?*t* (*B son* (*temperature* ?*t*))))).

This type of belief, i.e., that the agent believes another agent has a particular intention triggers an IntentionTransfer SIP. This user-customizable SIP, when fired, causes the agent to adopt the intention of the other agent.

(*I display*
(*exists* ?*t* (*B son* (*temperature* ?*t*)))).

At this point, the RationalityPrinciple SIP kicks in. This SIP searches the agent's base of actions for an action whose rational effect matches the intention. If the content of the intention is to achieve a goal in the agent's environment, then the PlanningSIP could be triggered, which would try to create a plan to achieve the goal. In this case, the result of the SIP will be to produce an action which informs the son of the current temperature.

(*Inform-ref :sender display :receiver* (*:set son*)
*:content* "((*any* ?*t* (*temperature* ?*t*)))").

JSA can be customized in various ways. In addition to writing SIPs or modifying the default ones, users can write assert and query filters. These are called when a sentence is asserted or queried (resp.) in the belief base. A filter contains a trigger which matches certain sentences. If the trigger matches, then the filter is applied to the sentence. The conditions for belief and intention transfer are also customizable. These are the conditions under which an agent will adopt the beliefs or intentions of other agents. Intention transfer is the main mechanism to implement cooperation among agents in this framework.

### 2.2 The Situation Calculus and IndiGolog

The *situation calculus* [20, 25] is a predicate logic language for representing dynamically changing worlds in which all changes are the result of named *actions*. The constant $S_0$ is used to denote the *initial situation*, namely that situation in which no actions have yet occurred. There is a distinguished binary function symbol *do*, where *do*$(a, s)$ denotes the successor situation to $s$ resulting from performing the action $a$. Relations (resp. functions) whose values vary from situation to situation, are called *fluents*, and are denoted by predicate (resp. function) symbols taking a situation term as their last argument. There is a special predicate *Poss*$(a, s)$ used to state that action $a$ is executable in situation $s$.

Within this language, one can formulate action theories that describe how the world changes as the result of the available actions as in [25] :
- Axioms describing the initial situation, $S_0$.
- Action precondition axioms, one for each primitive action $a$, characterizing *Poss*$(a, s)$.
- Successor state axioms, one for each relational fluent $F$ (resp. functional fluent $f$), which characterize the conditions under which $F(\vec{x}, do(a, s))$ holds (resp. $f(\vec{x}, do(a, s)) = v$) in terms of what holds

in situation $s$ ; these axioms may be compiled from effects axioms, but provide a solution to the frame problem [24].
- Unique names axioms for the primitive actions.
- A set of foundational, domain independent axioms for situations $\Sigma$.

On top of situation calculus action theories, logic-based programming languages can be defined, which, in addition to the primitive actions, allow the definition of complex actions. The ConGolog language [6], an extension of Golog [16], provides the following rich set of programming constructs :

| | |
|---|---|
| $\alpha$, | primitive action |
| $\phi?$, | wait for a condition |
| $\delta_1; \delta_2$, | sequence |
| $\delta_1 \mid \delta_2$, | nondeterministic branch |
| $\pi\, x.\, \delta$, | nondet. choice of argument |
| $\delta^*$, | nondeterministic iteration |
| **if** $\phi$ **then** $\delta_1$ **else** $\delta_2$ **endIf**, | conditional |
| **while** $\phi$ **do** $\delta$ **endWhile**, | while loop |
| $\delta_1 \parallel \delta_2$, | concurrency with equal priority |
| $\delta_1 \rangle\!\rangle \delta_2$, | prioritized concurrency |
| $\delta^{\parallel}$, | concurrent iteration |
| $\langle\, \vec{x} : \phi \rightarrow \delta \,\rangle$, | interrupt |
| $p(\vec{\theta})$, | procedure call |

Among these constructs, we notice the presence of nondeterministic constructs. These include $(\delta_1 \mid \delta_2)$, which nondeterministically chooses between programs $\delta_1$ and $\delta_2$, $\pi\, x.\, \delta$, which nondeterministically picks a binding for the variable $x$ and performs the program $\delta$ for this binding of $x$, and $\delta^*$, which performs $\delta$ zero or more times. Also notice that ConGolog includes constructs for dealing with concurrency. In particular $(\delta_1 \parallel \delta_2)$ expresses the concurrent execution (interpreted as interleaving) of the programs $\delta_1$ and $\delta_2$. Beside $(\delta_1 \parallel \delta_2)$, ConGolog includes other constructs for dealing with concurrency, such as prioritized concurrency $(\delta_1 \rangle\!\rangle \delta_2)$, where $\delta_1$ runs at higher priority than $\delta_2$, and interrupts $\langle\, \vec{x} : \phi \rightarrow \delta \,\rangle$. We refer the reader to [6] for a detailed account of ConGolog.

In [6], a single step transition semantics in the style of [21] is defined for ConGolog programs. Two special predicates *Trans* and *Final* are introduced. $\mathit{Trans}(\delta, s, \delta', s')$ means that by executing program $\delta$ starting in situation $s$, one can get to situation $s'$ in one elementary step with the program $\delta'$ remaining to be executed. $\mathit{Final}(\delta, s)$ means that program $\delta$ may successfully terminate in situation $s$.

*Offline executions* of programs, which are the kind of executions originally proposed for Golog and ConGolog [16, 6], are characterized using the $\mathit{Do}(\delta, s, s')$ predicate, which means that there is an execution of program $\delta$ that starts in situation $s$ and terminates in situation $s'$ :

$$\mathit{Do}(\delta, s, s') \stackrel{\text{def}}{=} \exists \delta'.\mathit{Trans}^*(\delta, s, \delta', s') \wedge \mathit{Final}(\delta', s'),$$

where $\mathit{Trans}^*$ is the reflexive, transitive closure of *Trans*. An offline execution of $\delta$ from $s$ is a sequence of actions $a, \ldots, a_n$ such that : $\mathcal{D} \cup \mathcal{C} \models \mathit{Do}(\delta, s, do(a_n, \ldots, do(a_1, s) \ldots))$, where $\mathcal{D}$ is an action theory as mentioned above, and $\mathcal{C}$ is a set of axioms defining the predicates *Trans* and *Final* and the encoding of programs as first-order terms [6].

Golog and ConGolog programs were intended to be executed *offline*, that is, a complete execution was obtained before committing even to the first action. However, IndiGolog [7, 27], [1] the latest language within the Golog family, provides a formal logic-based account of interleaved planning, sensing, and action by executing programs *online* and using a specialized new construct $\Sigma(\delta)$, the *search operator*, to perform local offline planning when required.

Roughly speaking, an *online* execution of a program finds a next possible action, executes it in the real world, then obtains sensing information, and repeats the cycle until the program is completed. Formally, an online execution is a sequence of so-called online configurations of the form $(\delta, \sigma)$, where $\delta$ is a high-level program and $\sigma$ is a history (see [7] for its formal definition). A history contains the sequence of actions executed so far as well as the sensing information obtained.

## 3 Generic Architecture

In this section, we describe the architecture we propose for combining JSA with APLs. There were two guiding principles we followed when designing our generic architecture for combining JSA with APLs. 1) We strove for modularity, i.e., we wanted to avoid the duplication of control structures and information storage in JSA and the APL. This facilitates the design and modification of a system by keeping information and control either in the JSA component or the APL component but not both. 2) We wanted to insulate the APL programmer as much as possible from having to deal with JSA programming. That is, we aimed to develop the JSA side

[1] `http ://sourceforge.net/projects/indigolog/`

of the system so that the APL programmer can avoid JSA programming altogether.

Generally, we kept the conversational aspects of the system on the JSA side and the information storage and planning aspects on the APL side. In particular, let the *objective belief base* of an agent be the set of beliefs the agent has about the world. These beliefs cannot contain belief or intention operators but may include (for ACLs that support them, such as IndiGolog) the temporal operators, i.e., *done* and *feasible*. The objective belief base is maintained on the APL side but is also used by JSA. However *subjective beliefs*, i.e., beliefs whose content contains beliefs and/or intentions are handled by JSA, since the APLs of which we are aware do not handle subjective beliefs. Planning and plan execution to bring about intentions are also handled on the APL side. The JSA component is used to handle communication with other agents. The default behavior of JSA is intended to be customized to suit the needs of a particular application using JSA and JADE. This customization includes what amounts to agent programming. Instead, we propose that the application-specific customization be done in the APL since APLs provide more sophisticated tools for programming agents.

To this end, we propose modifying JSA so that the objective beliefs of an agent that would be asserted by the JSA default behavior (e.g., as a result of an inform action) are passed on to the APL and stored in the belief base of the APL. If JSA has a query about an objective belief, the belief is passed on to the APL which answers the query if it can and the answer is passed back to JSA. This would be handled by adding an assert filter which checks if the asserted sentence is objective and if so it sends the sentence to be asserted in the APL belief base. The sentence is prevented from being asserted in the JSA belief base. Similarly, a query filter is put in place so that objective queries are sent to the APL and the answer is sent back to JSA. However, sentences that are not objective are still asserted and queried within JSA, such as beliefs about beliefs or beliefs about intentions.

The APL's intention base would be used to store the agent's intentions (using assert filters) and queries about the agent's intentions would be handled by the APL. The APL would be used for planning or using precompiled plans according to the capabilities of the APL.

The JSA process and the APL process would be running independently. However, the APL process must be responsive to the JSA process in order to handle assertions to the belief base and belief queries. The APL may also be called upon by the JSA process to plan or execute precompiled plans in order to generate actions to perform. The architecture is illustrated in Fig. 1.

The APL process might also want to send out messages on its own, i.e., independently of the JSA process interpreting received messages. This could be handled with a separate JADE or JSA process which takes as input messages to be sent by the APL, handles the translation into FIPA ACL, and sends the message to the appropriate agents. JSA has predefined methods that initiate communication (using FIPA ACL) with other agents that could be adapted for this purpose.

The architecture will have to be tailored to each APL since different APLs use different languages and have different capabilities. For example, IndiGolog has temporal operators and so can handle the *done* and *feasible* operators, but some BDI APLs do not have temporal operators. On the other hand, IndiGologdoes not have a built-in intention base nor does it have a built-in facility for updating beliefs about a particular fluent due to an exogenous action, so these would have to be added. Also, the form of the formulas that could be asserted in the belief base of both IndiGolog and BDI agents would have to be restricted to conjunctions of literals.

## 4 JSA+IndiGolog Architecture

In this section, we discuss how the generic architecture described above could be instantiated and implemented with the APL being IndiGolog. There are several issues that need to be addressed to obtain such an implemented architecture : how to handle 1) assertions of objective beliefs, 2) belief queries, 3) assertions of intentions, and 4) sending FIPA ACL messages from IndiGolog using JSA. Other issues, such as how to handle Call for Proposals performatives, intention and belief transfer, supporting queries from IndiGolog to the JSA belief base, etc., are left for future work.

As discussed above, objective beliefs will be stored in the IndiGolog belief base. A simple way to handle belief update is to introduce two special actions for each fluent $F$ : $add_F(\vec{x})$, which makes the fluent $F(\vec{x})$ true, and $delete_F(\vec{x})$,

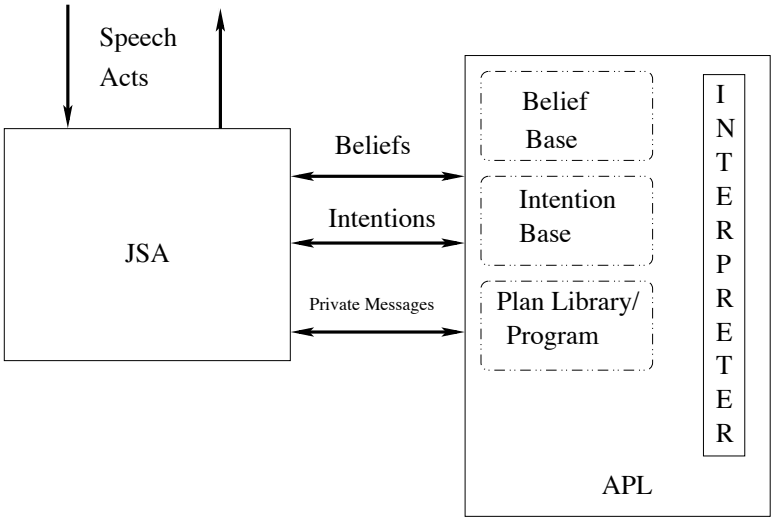



FIG. 1 – Diagram of JSA-APL architecture

which makes $F(\vec{x})$ false. We assume that the objective formulas to be asserted in the IndiGolog belief base are literals, however it is straightforward to generalize the account to handle conjunctions of literals. We also assume that the implementation of the generic architecture on the JSA side translates assertions of positive literals into $add_F$ actions and assertions of negative literals into $delete_F$ actions. Then for each fluent $F$, we add the $add_F$ and $delete_F$ actions to the successor state axiom for $F$. That is, if the successor state axiom for $F$ is of the form :

$$F(\vec{x}, do(a, s)) \equiv \\ (\gamma^+(\vec{x}, s) \vee (F(\vec{x}, s) \wedge \neg\gamma^-(\vec{x}, s))),$$

we modify it as follows :

$$F(\vec{x}, do(a, s)) \equiv \\ [a = add_F(\vec{x}) \vee \gamma^+(\vec{x}, s) \vee \\ (F(\vec{x}, s) \wedge \neg(a = delete_F(\vec{x}) \vee \gamma^-(\vec{x}, s)))].$$

Note that this only performs a very simple form of belief update. In response to a belief being updated, one may want to revise related beliefs (e.g., to satisfy some state constraints), or hypothesize the occurrence of exogenous actions. This could be handled using programmer provided rules that can be triggered by belief updates. We leave the details for future work.

Handling belief base queries is a little more involved since it involves action on the part of the IndiGolog agent. When a query comes in, the agent checks its belief base to see if the query is true or false and replies accordingly. For simplicity, we assume here that the belief base is complete. But note there has been some work on allowing incomplete knowledge in IndiGolog [29], which can be used to relax this assumption.

Queries are implemented with the $query(\phi)$ exogenous action, which is executed by the JSA agent when it wants to know the value of the objective formula $\phi$. The IndiGolog agent replies using the $reply(\phi)$ action, if $\phi$ currently holds, or $reply(\neg\phi)$, if $\neg\phi$ currently holds.

We introduce a new fluent $Queried(\phi)$ which holds, if $\phi$ has been queried but no reply has been issued yet :

$$Queried(\phi, do(a, s)) \equiv \\ (a = query(\phi) \vee \\ (Queried(\phi, s) \wedge a \neq reply(\phi) \wedge \\ a \neq reply(\neg\phi))).$$

Next we define a complex action which processes queries :

$$ProcessQueries \stackrel{\text{def}}{=} \\ \langle\, \phi : Queried(\phi) \rightarrow \textbf{if } \phi \textbf{ then } reply(\phi) \textbf{ else } \\ reply(\neg\phi) \textbf{ endIf} \,\rangle.$$

Now, given an IndiGolog program $\delta$, we can enable query processing in $\delta$ by executing *ProcessQueries* concurrently with and at a higher priority than $\delta$, i.e., $ProcessQueries \gg \delta$.

IndiGolog, unlike BDI languages, does not have a built-in intention base, and control in IndiGolog is driven by the agent's program as opposed to

its intentions. Thus, in general it would be up to the IndiGolog programmer to decide how intentions should be incorporated into the program.

We can implement an intention base in IndiGolog using a fluent, *Requested*($\phi$), where $\phi$ is a state formula representing an achievement goal (intentions to perform a complex action $\delta$ could be represented as *Done*($\delta$)). The exogenous action *request*($\phi$) adds $\phi$ to the intention base (i.e., causes *Requested*($\phi$) to hold), while the exogenous action *cancelReq*($\phi$) drops $\phi$ from the intention base.

The successor state axiom for *Requested* is as follows :

$$\textit{Requested}(\phi, \textit{do}(a, s)) \equiv (a = \textit{request}(\phi) \vee (\textit{Requested}(\phi, s) \wedge a \neq \textit{cancelReq}(\phi))).$$

The IndiGolog program can access the intention base (using the *Requested* fluent) in order to work towards achieving the intentions.

There will be cases where an IndiGolog agent wants to initiate FIPA ACL messages rather than just reacting to messages received from other agents. Since JSA already has a library for creating and sending FIPA ACL messages, we propose to use that rather than recreating it for IndiGolog (and for other ACLs using our generic architecture). To do this, we propose a special action *sendJSA*(*perf*, *list*), which takes as its first argument the performative of the message, and as its second argument, a list containing the remainder of the components of the message, which would include the recipient list for the message and whatever other components are appropriate for the given performative. The JSA agent would have a thread waiting for such a message, and upon receipt, it would create the appropriate FIPA ACL message for the performative and component list, and send the message to the recipients.

To implement the communication between JSA and IndiGolog, we would use TCP/IP sockets. The current IndiGolog implementation allows users to specify device managers, which listen on a specified socket for messages which can be custom translated into exogenous actions and then inserted into the history of actions. Similarly, device managers handle the implementation of physical actions, which would include messages sent from the IndiGolog agent to the JSA agent. It would not be difficult to implement a device manager to handle the communication between IndiGolog and JSA.

# 5 Discussion and Conclusion

In this paper, we outlined an approach to combining the sophisticated communication capabilities of JSA with the well-developed agent programming features of APLs. We believe the combination of these languages/tools can lead to a very flexible framework for implementing agent systems with complex behavior and communication capacities. First, we outlined a generic architecture that could be applicable to a variety of APLs, and then we discussed how the architecture could be implemented for IndiGolog.

As this work is still in its preliminary stages, there are many issues that remain to be resolved. One issue is where to handle the specification of when an agent will accept intention and belief transfers. In keeping with our strategy of isolating the APL programmer from having to customize JSA, we would suggest using special predicates on the APL side to specify under which conditions such transfers would be allowed. If the decisions about these transfers are static, i.e., the decisions could be made in advance, then this would be possible. However, if an agent is allowed to change its mind about whether to accept belief or intention transfers from another agent at run-time, then this would be difficult to implement on the APL side, since the beliefs about the beliefs and intentions of other agents (which would be needed to make these decisions) are stored on the JSA side and the APL does not have access to them. However, we note that APLs generally do not allow the representation of beliefs about the beliefs and intentions of other agents, so this would seem to be a reasonable restriction.

Another issue still to be resolved is how to accommodate Call for Proposal messages. A Call for Proposal is a FIPA ACL performative that is used to initiate negotiations between agents and is therefore somewhat more difficult to implement in an APL than simpler performatives such as Inform and Request (the recipient must decide what proposal to make, if any). We reserve the handling of this performative for future work. In future work, we also plan to implement the architecture for IndiGolog and a BDI APL, and evaluate the resulting platforms.

# Références

[1] F.L. Bellifemine, G. Claire, and D. Greenwood. *Developing Multi-Agent Systems*

*with JADE*. Wiley, West Sussex, England, 2007.

[2] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming Multi-agent Systems in AgentSpeak Using Jason*. Wiley Series in Agent Technology. Wiley, West Sussex, England, 2007.

[3] R.H. Bordini, M. Dastani, J. Dix, and A. El Fallah-Seghrouchni, editors. *Multi-Agent Programming : Languages, Platforms and Applications*. Springer, New York, NY, 2005.

[4] Michael E. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press, 1987.

[5] Paolo Busetta, Ralph Rönnquist, Andew Hodgson, and Andrew Lucas. JACK intelligent agents : Components for intelligent agents in Java. *AgentLink News*, 2 :2–5, January 1999.

[6] Giuseppe De Giacomo, Yves Lespérance, and Hector J. Levesque. ConGolog, a concurrent programming language based on the situation calculus. *Artificial Intelligence Journal*, 121(1–2) :109–169, 2000.

[7] Giuseppe De Giacomo and Hector J. Levesque. An incremental interpreter for high-level programs with sensing. In Hector J. Levesque and Fiora Pirri, editors, *Logical Foundations for Cognitive Agents : Contributions in Honor of Ray Reiter*, pages 86–102. Springer, Berlin, 1999.

[8] Daniel Dennett. *The Intentional Stance*. The MIT Press, 1987.

[9] The Foundation for Intelligent Physical Agents, an IEEE Computer Society Standards Committee. `www.fipa.org`.

[10] Michael P. Georgeff and Amy L. Lansky. Reactive reasoning and planning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 677–682, Seattle, USA, 1987.

[11] The DARPA Knowledge Sharing Initiative External Interfaces Working Group. Specification of the KQML agent-communication language – plus example agent policies and architectures. Available at `http ://www.cs.umbc.edu/kqml/`.

[12] Alejandro Guerra-Hernández, Amal El Fallah-Seghrouchni, and Henry Soldano. Learning in BDI multi-agent systems. In Jürgen Dix and João Alexandre Leite, editors, *Computational Logic in Multi-Agent Systems, 4th International Workshop, CLIMA IV*, volume 3259 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 218–233. Springer, 2004.

[13] Koen V. Hindriks, Frank S. de Boer, Wiebe van der Hoek, and John-Jules Ch. Meyer. Agent programming in 3APL. *Autonomous Agents and Multi-Agent Systems*, 2 :357–401, 1999.

[14] Marcus J. Huber. JAM : A BDI-theoretic mobile agent architecture. In *Proceedings of the Annual Conference on Autonomous Agents (AGENTS)*, pages 236–243, New York, NY, USA, 1999. ACM Press.

[15] Samin Karim, Budhitama Subagdja, and Liz Sonenberg. Plans as products of learning. In *IEEE/ACM International Conference on Intelligent Agent Technology (IAT 2006)*, pages 139–145. IEEE Computer Society, 2006.

[16] Hector J. Levesque, Ray Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl. GOLOG : A logic programming language for dynamic domains. *Journal of Logic Programming*, 31 :59–84, 1997.

[17] Vincent Louis and Thierry Martinez. The JADE semantic agent : towards agent communication oriented midleware. *AgentLink News*, 18 :16–18, August 2005.

[18] Vincent Louis and Thierry Martinez. Operational model for the FIPA-ACL semantics. In Frank Dignum, Rogier M. van Eijk, and Roberto Flores, editors, *Agent Communication II : International Workshops on Agent Communication, AC 2005 and AC 2006, Utrecht, Netherlands, July 25, 2005, and Hakodate, Japan, May 9, 2006, Selected and Revised Papers*, volume 3859 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 1–14. Springer-Verlag, 2006.

[19] Erick Martinez and Yves Lespérance. IG-JADE-PKSlib : an agent-based framework for advanced web service composition and provisioning. In *Proc. of the AAMAS 2004 Workshop on Web-services and Agent-based Engineering*, pages 2–10, New York, NY, USA, July 2004.

[20] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4 :463–502, 1969.

[21] Gordon D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI-FN-19, Computer Science Department, Aarhus University, Denmark, 1981.

[22] Martha E. Pollack. The uses of plans. *Artificial Intelligence Journal*, 57(1) :43–68, 1992.

[23] Anand S. Rao. AgentSpeak(L) : BDI agents speak out in a logical computable language. In W. Vander Velde and J. W. Perram, editors, *Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World. (Agents Breaking Away)*, volume 1038 of *Lecture Notes in Computer Science (LNCS)*, pages 42–55. Springer-Verlag, 1996.

[24] Ray Reiter. The frame problem in the situation calculus : A simple solution (sometimes) and a completeness result for goal regression. In Vladimir Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation : Papers in Honor of John McCarthy*, pages 359–380. Academic Press, San Diego, CA, 1991.

[25] Ray Reiter. *Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press, 2001.

[26] D. Sadek and P. Bretier. ARTIMIS : Natural dialogue meets rational agency. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1030–1035, 1997.

[27] Sebastian Sardina, Giuseppe De Giacomo, Yves Lespérance, and Hector J. Levesque. On the semantics of deliberation in IndiGolog – from theory to implementation. *Annals of Mathematics and Artificial Intelligence*, 41(2–4) :259–299, August 2004.

[28] Sebastian Sardina and Lin Padgham. Goals in the context of BDI plan failure and planning. In Edmund H. Durfee, Makoto Yokoo, Michael N. Huhns, and Onn Shehory, editors, *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 16–23, Hawaii, USA, May 2007. ACM Press.

[29] Stavros Vassos and Hector Levesque. Progression of situation calculus action theories with incomplete information. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2024–2029, 2007.

[30] Michael Winikoff, Lin Padgham, James Harland, and John Thangarajah. Declarative & procedural goals in intelligent agent systems. In *Proceedings of Principles of Knowledge Representation and Reasoning (KR)*, pages 470–481, Toulouse, France, April 2002.

# Assumption-based argumentation for the minimal concession strategy of agents engaged in resource negotiation[1]

M. Morge[†] morge@di.unipi.it
P. Mathieu[⋆] Philippe.Mathieu@lifl.fr
P. Mancarella[†] paolo@di.unipi.it

[†]Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo, 3
I-56127 Pisa, Italy

[⋆]Laboratoire d'Informatique Fondamentale de Lille
Université des Sciences et Technologies de Lille
F-59655 Villeneuve d'ascq cédex

**Résumé :**
De récents travaux s'intéressent au modèle informatique de l'argumentation pour la négociation multi-agents. Toutefois, cette approche logique ne propose pas de stratégie d'agent efficace pour la négociation. Dans cet article, nous présentons une réalisation de la stratégie de concession minimale et nous l'illustrons à l'aide d'un exemple intuitif de négociation de ressources. Nous affirmons ici que le résultat d'une négociation, dont on garantit l'arrêt, est un accord optimal (si possible), lorsque les agents adoptent notre stratégie.

**Mots-clés :** Comportement d'agent, Négociation, Argumentation, Théorie des jeux, Allocation multi-agents de ressources

**Abstract:**
Several recent works in the area of Artificial Intelligence focus on computational models of argumentation-based negotiation. However, even if computational model of arguments are used to encompass the reasoning of interacting agents, this logical approach does not come with an effective strategy for agents engaged in negotiations. In this paper we present a realisation of the Minimal Concession (MC) strategy. The main contribution of this paper is the realisation of this intelligent strategy by means of assumption-based argumentation illustrated by an intuitive scenario of resource negotiation. Moreover, we claim here that the outcome of negotiations, which are guaranteed to terminate, is an optimal agreement (when possible) if the agents adopt the MC strategy.

**Keywords:** Agent behaviour, Negotiation, Argumentation, Game theory, Multiagent Resource Allocation

## 1 Introduction

In negotiations, the aim for all parties is to "make a deal" while bargaining over their interests, typically seeking to maximise their "good" (welfare), and prepared to concede some aspects while insisting on others. Negotiations can be delegated to a multiagent system responsible for reaching agreements automatically [9]. As pointed by [15], there is a need for a solid theoretical foundation for negotiation that covers algorithms and protocols, while determining which strategies are most effective under what circumstances.

Several recent works in the area of Artificial Intelligence focus on computational models of argumentation-based negotiation [12, 1, 5]. In these works, argumentation logic serves as a unifying medium to provide a model for agent-based negotiation systems, in that it can support : the reasoning and decision-making process of agents [12], the inter-agent negotiation process to reach an agreement [1] and the definition of contracts emerging from the negotiation [5]. However, even if computational model of arguments are used to encompass the reasoning of interacting agents, few works are concerned by the strategy of agents engaged in negotiations and its properties. A first attempt in this direction is the Minimal Concession (MC) strategy proposed by [7]. However, the latter does not show how to fill the gap between the argumentation-based decision-making mechanism and its realisation for computing this negotiation strategy. Moreover, some assumptions are too strong such as the fact the agents know the preferences and the reservation values of the other agents. In this paper we present our realisation of the MC strategy [17]. Moreover, we consider it for the specific case of resource negotiation. Argumentation logic is used to support the intelligent strategy of negotiating agents, to guide and empower negotiation amongst agents and to allow them to reach agreements. With the support of assumption-based argumentation, agents select the "optimal" utterances to fulfil the preferences/constraints of users and the requirements imposed by the other agents.

The paper is organised as follows. Section 2 introduces the basic notions of assumption-based

[⋆]This work is partially supported by the Sixth Framework IST programme of the EC, under the 035200 ARGUGRID project.

argumentation in the background of our work. Section 3 introduces the walk-through example. Section 3 outline the dialogue-game protocol we use. Section 5 defines our framework for decision making. Section 6 presents our realisation of the MC strategy. Section 7 highlights some properties of our protocol and strategy. Section 8 discusses some related works. Section 9 concludes.

## 2 Assumption-based argumentation

Assumption-based argumentation [3] (ABA) is a general-purpose computational framework which allows to reason with incomplete information since certain literals are assumptions, meaning that they can be assumed to hold as long as there is no evidence to the contrary. Moreover, ABA concretise Dung's abstract argumentation [6] (AA). Actually, all the semantics used in AA, which captures various degrees of collective justification for a set of arguments, can be applied.

An ABA framework considers a deductive system augmented by a non-empty set of assumptions and a (total) mapping from assumptions to their contraries. In order to perform decision making, we consider here the generalisation of the original assumption-based argumentation framework and its computational mechanism, whereby multiple contraries are allowed [8].

**Definition 1 (ABA)** *An **assumption-based argumentation framework** is a tuple* $\mathtt{ABF} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}sm, \mathcal{C}on \rangle$ *such that :*
- $(\mathcal{L}, \mathcal{R})$ *is a deductive system where,*
  - $\mathcal{L}$ *is a formal language consisting of countably many sentences,*
  - $\mathcal{R}$ *is a countable set of inference rules of the form* $r : \alpha \leftarrow \alpha_1, \ldots, \alpha_n$ $(n \geq 0)$ *where* $\alpha \in \mathcal{L}$*, called the **head** of the rule (denoted* $\mathtt{head}(r)$*),* $\alpha_1, \ldots, \alpha_n \in \mathcal{L}$ *, called the **body** (denoted* $\mathtt{body}(r)$*), and* $n \geq 0$ *;*
- $\mathcal{A}sm \subseteq \mathcal{L}$ *is a non-empty set of **assumptions**. If* $x \in \mathcal{A}sm$*, then there is no inference rule in* $\mathcal{R}$ *such that* $x$ *is the head of this rule ;*
- $\mathcal{C}on : \mathcal{A}sm \rightarrow 2^{\mathcal{L}}$ *is a (total) mapping from assumptions into set of sentences in* $\mathcal{L}$*, i.e. their **contraries**.*

In the remainder of the paper, we restrict ourselves to finite deduction systems, i.e. with finite languages and finite set of rules. For simplicity, we restrict ourselves to flat frameworks [3], i.e. whose assumptions do not occur as conclusions of inference rules, such as logic programming or the frameworks considered in this paper.

We adopt here a tree-like structure for arguments.

**Definition 2 (Argument)** *Let* $\mathtt{ABF} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}sm, \mathcal{C}on \rangle$ *be an ABA framework. An **argument** $\bar{a}$ deducing the **conclusion** $c \in \mathcal{L}$ (denoted* $\mathtt{conc}(\bar{a})$*) supported by a set of **assumptions** $\mathtt{A}$ in $\mathcal{A}sm$ (denoted* $\mathtt{asm}(\bar{a})$*) is a tree where the root is $c$ and each node is a sentence of $\mathcal{L}$. For each node :*
- *if the node is a leaf, then it is either an assumption of* $\mathtt{A}$ *or* $\top$[1] *;*
- *if the node is not a leaf and it is $\alpha \in \mathcal{L}$, then there is an inference rule $\alpha \leftarrow \alpha_1, \ldots, \alpha_n$ in $\mathcal{R}$ and,*
  - *either $n = 0$ and $\top$ is its only child,*
  - *or $n > 0$ and the node has $n$ children, $\alpha_1, \ldots, \alpha_n$.*

*We write it* $\bar{a} : \mathtt{A} \vdash c$*. The set of arguments built upon* $\mathtt{ABF}$ *is denoted by* $\mathcal{A}(\mathtt{ABF})$*.*

In the remainder of the paper, we restrict ourselves to finite deduction systems, i.e. with finite languages and finite set of rules. For simplicity, we also restrict ourselves to flat frameworks [3], in which assumptions do not occur as conclusions of inference rules.

In an assumption-based argumentation framework, the attack relation amongst arguments comes from the contraries which capture the notion of conflicts.

**Definition 3 (Attack relation)** *An argument* $\bar{a} : \mathtt{A} \vdash \alpha$ ***attacks** an argument* $\bar{b} : \mathtt{B} \vdash \beta$ *iff there is an assumption* $x \in \mathtt{B}$ *such as* $\alpha \in \mathcal{C}on(x)$*. Similarly, we say that the set $\bar{S}$ of arguments attacks $\bar{b}$ when $\bar{a} \in \bar{S}$.*

According to the two previous definitions, ABA is clearly a concrete instantiation of AA where arguments are deductions and the attack relation comes from the contrary relation.

We have defined the attack-relation to adopt Dung's calculus of opposition [6].

[1] $\top$ denotes the unconditionally true statement.

| Move | Speaker | Proposal |
|---|---|---|
| $\mathtt{mv}_1$ | $\mathrm{ag}_1$ | grab $\mathrm{r}_3$ |
| $\mathtt{mv}_2$ | $\mathrm{ag}_2$ | grab $\mathrm{r}_1$ and $\mathrm{r}_2$ |
| $\mathtt{mv}_3$ | $\mathrm{ag}_1$ | none swap |
| $\mathtt{mv}_4$ | $\mathrm{ag}_2$ | swap all the resources |
| $\mathtt{mv}_5$ | $\mathrm{ag}_1$ | swap $\mathrm{r}_2$ and $r_3$ |
| $\mathtt{mv}_6$ | $\mathrm{ag}_2$ | grab $r_2$ |
| $\mathtt{mv}_7$ | $\mathrm{ag}_1$ | swap $r_1$ and $r_3$ |
| $\mathtt{mv}_8$ | $\mathrm{ag}_2$ | swap $r_2$ and $r_3$ |

TAB. 1 – Negotiation dialogue

**Definition 4 (Semantics)** *Let* $\mathtt{AF} = \langle \mathcal{A}(\mathtt{ABF}), \mathtt{attacks} \rangle$ *be our argumentation framework built upon the ABA framework* $\mathtt{ABF} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}sm, \mathcal{C}on \rangle$. *A set of arguments* $\bar{S} \subseteq \mathcal{A}(\mathtt{ABF})$ *is :*

- ***conflict-free*** *iff* $\forall \bar{a}, \bar{b} \in \bar{S}$ *it is not the case that* $\bar{a}$ *attacks* $\bar{b}$ *;*
- ***admissible*** *iff* $\bar{S}$ *is conflict-free and* $\bar{S}$ *attacks every argument* $\bar{a}$ *such that* $\bar{a}$ *attacks some arguments in* $\bar{S}$.

For simplicity, we restrict ourselves to admissible semantics.

## 3 Walk-through example

In order to present informally our negotiation strategy, we consider a scenario where two agents seek to swap discrete, non-divisible and non-shareable resources. The negotiation of the allocation is a complex task due to the number of possible allocations, their characteristics and the preferences of the users. It makes this use-case interesting enough for the evaluation of our strategy.

In our scenario, the initial allocation is such that the agent $\mathrm{ag}_1$ has the resources $\mathrm{r}_1$ and $\mathrm{r}_2$ while the agent $\mathrm{ag}_2$ has the resource $\mathrm{r}_3$. While the agent $\mathrm{ag}_1$ is empowered to collect as much resources as possible whatever the resources are, the agent $\mathrm{ag}_2$ is responsible for collecting $\mathrm{r}_2$ and eventually as much other resources as possible. Taking into account these goals/preferences, the agent $\mathrm{ag}_1$ (resp. the $\mathrm{ag}_2$) needs to interactively solve a decision-making problem where the decision amounts to a deal it can agree on. Moreover, some decisions amount to the moves they can utter during the negotiation.

We consider the negotiation performed through the moves in Tab. 1. A move at time $t$ has an identifier $\mathtt{mv}_t$ and it is uttered by a speaker. According to the MC strategy, the agents start with the proposals which are "optimal" for themselves. Each of them suggests to take all the resources. In the third step of the negotiation, the agent $\mathrm{ag}_1$ adopting the MC strategy must concede minimally : either with the empty deal where none resource is exchanged, or the swap of the resources $\mathrm{r}_2$ and $\mathrm{r}_3$, or the swap of the resources $\mathrm{r}_1$ and $\mathrm{r}_3$. Arbitrarily, it suggests the empty deal, and so implicitly it rejects the previous proposal of the agent $\mathrm{ag}_2$. It is rational for the agent $\mathrm{ag}_2$ to reject the empty deal since it does not allow the agent $\mathrm{ag}_2$ to take the resource $\mathrm{r}_2$. The agent $\mathrm{ag}_2$ is ready to concede $\mathrm{r}_3$ in order to get $\mathrm{r}_2$. Adopting the MC strategy, the agent $\mathrm{ag}_2$ concedes minimally by suggesting to swap all the resources. It is rational for the agent $\mathrm{ag}_1$ to reject this deal for which the number of its resources decreases. The agent $\mathrm{ag}_1$ implicitly rejects this deal by suggesting the swap of the resources $\mathrm{r}_2$ and $\mathrm{r}_3$. The agent $\mathrm{ag}_2$ prefers and so suggests to take the resource $\mathrm{r}_2$. Since this deal is still irrational for the agent $\mathrm{ag}_1$, the latter suggests the swap of the resources $\mathrm{r}_1$ and $\mathrm{r}_3$. Since the previous proposal put forward by the agent $\mathrm{ag}_2$ has been previously (implicitly) rejected, the agent $\mathrm{ag}_2$ must concede minimally. For this purpose, the agent $\mathrm{ag}_2$ suggest the proposal which is preferred and which has not been yet rejected, i.e. the swap of the resources $\mathrm{r}_2$ and $\mathrm{r}_3$. Since this proposal has been previously put forward by the agent $\mathrm{ag}_1$, the agent $\mathrm{ag}_2$ accepts it and the dialogue is closed.

## 4 Protocol

A negotiation is a social interaction between self-interested parties intended to resolve a dispute by verbal means and to produce an agreements upon a course of action. In this section, we briefly present our game-based social model to handle the collaborative operations of agents. In particular, we present a dialogue-game protocol for bilateral bargaining.

According to the game metaphor for social interactions of [20], agents are players which utter moves according to social rules.

**Definition 5 (Dialogue-game)** *Let us consider* $\mathcal{L}$ *a common object language and* $\mathcal{ACL}$ *a common agent communication language. A* ***dialogue-game*** *is a tuple* $\mathtt{DG} = \langle \mathtt{P}, \Omega_M, \mathtt{H}, \mathtt{T}, \mathtt{proto}, \mathtt{Z} \rangle$ *where :*

- $\mathtt{P}$ *is a set of agents called* ***players*** *;*

– $\Omega_M \subseteq \mathcal{ACL}$ *is a set of well-formed moves ;*
– $H$ *is a set of histories, the sequences of well-formed moves s.t. the speaker of a move is determined at each stage by the turn-taking function* $T$ *and the moves agree with the protocol* $proto$*;*
– $T : H \rightarrow P$ *is the turn-taking function ;*
– $proto : H \rightarrow 2^{\Omega_M}$ *is the function determining the legal moves which are allowed to expand an history ;*
– $Z$ *is the set of dialogues, i.e. the terminal histories.*

DG allows social interaction between agents. During a dialogue-game, players utter moves. Each dialogue is a maximally long sequence of moves. Let us now specify informally the elements of DF for bilateral bargaining.

In bilateral bargainings, there are two players, the initiator init and the partner part, which utter moves each in turn. The **syntax** of moves is in conformance with a common **agent communication language**, $\mathcal{ACL}$. A move at time $t$ : has an identifier, $\mathtt{mv}_t$ ; is uttered by a speaker ($\mathtt{sp}_t \in \mathtt{P}$) and the speech act is composed of a locution $\mathtt{loc}_t$ and a content $\mathtt{content}_t$. The possible locutions are : assert, reply, standstill, concede, accept and reject. The content consists of a sentence in the common object language, $\mathcal{L}$.

Given an history, the players share a **dialogue state**, depending on their previous moves. Considering the step $t \in \mathbb{N}$, the dialogue state is a tuple $\mathtt{DS}_t = \langle \mathtt{ll}_t, \mathtt{lo}_t(\mathtt{init}), \mathtt{lo}_t(\mathtt{part}), \mathtt{nbss}_t \rangle$ where :
– $\mathtt{ll}_t$ is the last locution which has been uttered, eventually none ;
– $\mathtt{lo}_t(\mathtt{init})$ (resp. $\mathtt{lo}_t(\mathtt{part})$) represents the last offer of the initiator (resp. partner), i.e. the content of its last move ;
– $\mathtt{nbss}_t$ is the number of consecutive standstill in the last moves.

Fig. 1 represents our dialogue-game protocol with the help of a deterministic finite-state automaton. A dialogue begins with a first offer when a participant (the initiator or the partner) makes an assert. The legal responding speech act is reply. After that, the legal responding moves are standstills, concessions, acceptations and rejections. The legal responding moves to a concession/standstill are the same. An history is final and : i) the dialogue is a failure if it is closed by a reject ; ii) the dialogue is a success if it is closed by an accept.

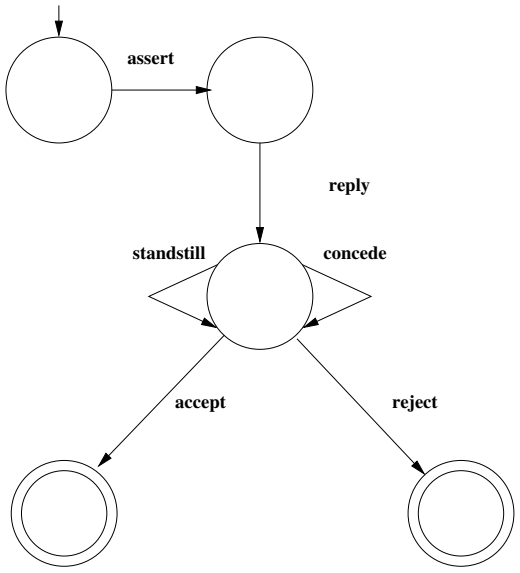



FIG. 1 – bilateral bargaining protocol

The strategy interfaces with the dialogue-game protocol through the condition mechanism of utterances for a move. For example, at a certain point in the dialogue the agent is able to send standstill or concede. The choice of which locution and which content to send depends on the agent's strategy.

## 5 Decision making

Taking into account the goals/preferences of the user, an agent needs to solve a decision-making problem where the decision amounts to a service it can agree on. This agent uses argumentation in order to assess their suitability and identify "optimal" services. It argues internally to link the deals and the benefits that these deals guarantee under possibly incomplete knowledge. This section presents our framework to perform decision making, illustrated by the agent $\mathrm{ag}_1$.

**Definition 6 (Decision framework)** *A* ***decision framework*** *is a tuple* $DF = \langle \mathcal{L}, \mathcal{G}, \mathcal{D}, \mathcal{B}, \mathcal{R}, \mathcal{A}sm, \mathcal{C}on, \mathcal{P} \rangle$ *such that :*
– $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}sm, \mathcal{C}on \rangle$ *is an ABA framework as defined in Def. 1 and* $\mathcal{L} = \mathcal{G} \cup \mathcal{D} \cup \mathcal{B}$ *where,*
  – $\mathcal{G}$ *is a set of literals in* $\mathcal{L}$ *called* ***goals****,*
  – $\mathcal{D}$ *is a set of assumptions in* $\mathcal{A}sm$ *called* ***decisions****,*
  – $\mathcal{B}$ *is a set of literals in* $\mathcal{L}$ *called* ***beliefs*** *;*
– $\mathcal{P} \subseteq \mathcal{G} \times \mathcal{G}$ *is a strict partial order over* $\mathcal{G}$*, called the* ***preference*** *relation.*

$\mathtt{o}([u,v,w],[x,y,z]) \leftarrow$ $\mathtt{d}([0,0,0],[0,0,0]),$ $\mathtt{control}(\mathrm{ag}_1,[u,v,w]),$ $\mathtt{control}(\mathrm{ag}_2,[x,y,z])$

$\mathtt{o}([u,v,1],[x,y,0]) \leftarrow$ $\mathtt{d}([0,0,0],[0,0,1]),$ $\mathtt{control}(\mathrm{ag}_1,[u,v,0]),$ $\mathtt{control}(\mathrm{ag}_2,[x,y,1])$

$\mathtt{o}([0,v,1],[1,y,0]) \leftarrow$ $\mathtt{d}([1,0,0],[0,0,1]),$ $\mathtt{control}(\mathrm{ag}_1,[1,v,0]),$ $\mathtt{control}(\mathrm{ag}_2,[0,y,1])$

$\mathtt{o}([u,0,1],[x,1,0]) \leftarrow$ $\mathtt{d}([0,1,0],[0,0,1]),$ $\mathtt{control}(\mathrm{ag}_1,[u,1,0]),$ $\mathtt{control}(\mathrm{ag}_2,[x,0,1])$

$\mathtt{control}(\mathrm{ag}_1,[1,1,0])$

$\mathtt{control}(\mathrm{ag}_2,[0,0,1])$

TAB. 2 – The inference rules of the agent $\mathrm{ag}_1$

In the object language $\mathcal{L}$, we distinguish three disjoint components : a set of **goals** representing the objectives the agent wants to be fulfilled, i.e. the possible resource allocations (e.g. the situation where the agent $\mathrm{ag}_1$ has the first resource and the third resource, $\mathtt{o}([1,0,1],[0,1,0])$) ; a set of **decisions** representing the possible deals (e.g. swapping the second resource with the third resource (denoted $\mathtt{d}([0,1,0],[0,0,1])$)) ; a set of **beliefs**, representing the initial resource allocation (e.g. the fact that the agent $\mathrm{ag}_1$ initially has the first resource and the second resource $\mathtt{control}(\mathrm{ag}_1,[1,1,0])$). Decisions are **assumptions**. The multiple **contraries** capture the mutual exclusion of alternatives.

The inference rules of the agent $\mathrm{ag}_1$ are depicted in Tab. 2. All variables occurring in an inference rule are implicitly universally quantified over the whole rule. A rule with variables is a scheme standing for all its ground instances. It is worth noticing that the agent only consider the **rational** deals (s.t the output is an allocation at least as preferred to the initial allocation). The inference rules of the agent $\mathrm{ag}_2$ are similar.

We consider the **preference** relation $\mathcal{P}$ over the goals in $\mathcal{G}$, which is transitive, irreflexive and asymmetric. $g_1 \mathcal{P} g_2$ can be read "$g_1$ is preferred to $g_2$". From the agent $\mathrm{ag}_1$ viewpoint, $\mathtt{o}([1,1,1],[0,0,0])\mathcal{P}\mathtt{o}([1,0,1],[0,1,0])$, $\mathtt{o}([1,1,1],[0,0,0])\mathcal{P}\mathtt{o}([1,1,0],[0,0,1])$ and $\mathtt{o}([1,1,1],[0,0,0])\mathcal{P}\mathtt{o}([0,1,1],[1,0,0])$.

Formally, given an argument $\bar{\mathrm{a}}$, let

$$\mathtt{dec}(\bar{\mathrm{a}}) = \mathtt{asm}(\bar{\mathrm{a}}) \cap \mathcal{D}$$

be the set of decisions supported by the argument $\bar{\mathrm{a}}$.

Decisions are suggested to reach a goal if they are supported by arguments.

**Definition 7 (Decisions)** *Let* $DF = \langle \mathcal{L}, \mathcal{G}, \mathcal{D}, \mathcal{B}, \mathcal{R}, \mathcal{A}sm, \mathcal{C}on, \mathcal{P} \rangle$ *be a decision framework,* $g \in \mathcal{G}$ *be a goal and* $D \subseteq \mathcal{D}$ *a set of decisions.*

- *The decisions* $D$ ***argue for*** $g$ *iff there exists an argument* $\bar{a}$ *such that* $conc(\bar{a}) = g$ *and* $dec(\bar{a}) = D$.
- *The decisions* $D$ ***credulously argue for*** $g$ *iff there exists an argument* $\bar{a}$ *in an admissible set of arguments such that* $conc(\bar{a}) = g$ *and* $dec(\bar{a}) = D$.
- *The decisions* $D$ ***skeptically argue for*** $g$ *iff for all admissible set of arguments* $\bar{S}$ *such that for some arguments* $\bar{a}$ *in* $\bar{S}$ $conc(\bar{a}) = g$, *then* $dec(\bar{a}) = D$.

*We denote* $val(D)$, $val_c(D)$ *and* $val_s(D)$ *respectively the set of goals in* $\mathcal{G}$ *for which the set of decisions* $D$ *argues, credulously argues and skeptically argues respectively.*

Due to the uncertainties, some decisions satisfy goals for sure if they skeptically argue for them, or some decisions can possibly satisfy goals if they credulously argue for them. While the first case is required for convincing a risk-averse agent, the second case is enough to convince a risk-taking agent. We focus here on risk-taking agents.

Since agents can consider multiple objectives which may not be fulfilled all together by a set of non-conflicting decisions, high-ranked goals must be preferred to low-ranked goals.

**Definition 8 (Preferences)** *Let* $DF = \langle \mathcal{L}, \mathcal{G}, \mathcal{D}, \mathcal{B}, \mathcal{R}, \mathcal{A}sm, \mathcal{C}on, \mathcal{P} \rangle$ *be a decision framework. We consider* $G$, $G'$ *two set of goals in* $\mathcal{G}$ *and* $D$, $D'$ *two set of decisions in* $\mathcal{D}$. $G$ *is* ***preferred*** *to* $G$ *(denoted* $G\mathcal{P}G'$*) iff*

1. $G \supseteq G'$, *and*
2. $\forall g \in G \setminus G'$ *there is no* $g' \in G'$ *such that* $g' \mathcal{P} g$.

$D$ *is* ***preferred*** *to* $D'$ *(denoted* $D\mathcal{P}D'$*) iff* $val_c(D)\mathcal{P}val_c(D')$.

The reservation value (denoted $\mathtt{RV}$) is the minimal set of goals which needs to be reached by a set of decisions to be acceptable. Formally, given a reservation value $\mathtt{RV}$, let

$$\mathtt{ad} = \{\mathtt{d}(x,y) \mid \exists \mathtt{D} \in \mathcal{D} \text{ s.t. } \mathtt{d}(x,y) \in \mathtt{D} \text{ and } \mathtt{val}_c(\mathtt{D})\mathcal{P}\mathtt{RV}\}$$

be the deal which can be accepted by the agent.

The set of decisions $\{\mathtt{d}([0,0,0],[0,0,1])\}$ is the only one which skeptically argues for having all the resources while both $\{\mathtt{d}([0,0,0],[0,0,1])\}$ and $\{\mathtt{d}([0,0,0],[0,0,0])\}$ credulously argue for having the two first resources. Due to the preferences of the agent $\mathrm{ag}_1$ over the goals, it prefers $\mathtt{d}([0,0,0],[0,0,1])$ to the other deals.

## 6 Minimal concession strategy

Taking into account the preferences/goals of the user and the dialogue state, an agent needs to solve some decision-making problems where the decision amounts to a move it can utter. This agent uses argumentation in order to assess the suitability of moves and identify "optimal" moves. It argues internally to link the current dialogue state, the legal moves (their speech acts and contents), and the resulting dialogue states of these moves under possibly incomplete knowledge. This section presents how our argumentation approach realizes the Minimal Concession (MC) strategy, illustrated by the agent $\mathrm{ag}_1$.

A dialogue strategy is a plan that specifies the moves chosen by a player to achieve a particular goal. We consider here the MC strategy which specifies the move chosen by the player for every history when it is his turn to move. For this purpose, an agent adopts a decision framework $\mathtt{DF} = \langle \mathcal{L}, \mathcal{G}, \mathcal{D}, \mathcal{B}, \mathcal{R}, \mathcal{A}sm, \mathcal{C}on, \mathcal{P} \rangle$. The latter, as illustrated in the previous section, allows to perform decision making where the decision amounts to the deal it can agree on. This DF must be extended to perform the MC strategy. For this purpose, we incorporate in the object language $\mathcal{L}$ :

- the goal respond (resp. optimal) in $\mathcal{G}$ representing the objective of the agent which consists of responding (resp. uttering the "optimal" move) ;
- the decisions in $\mathcal{D}$ representing the possible locutions (e.g. $\mathtt{loc}(\mathtt{concede})$). Obviously, the multiple contraries capture the mutual exclusion of the corresponding alternatives (e.g. $\{\mathtt{loc}(\mathtt{concede}), \mathtt{loc}(\mathtt{accept}), \mathtt{loc}(\mathtt{reject})\} = \mathcal{C}on(\mathtt{loc}(\mathtt{standstill}))$) ;
- a set of beliefs in $\mathcal{B}$, related to the dialogue state,
  - the last locution of the interlocutor (e.g. $\mathtt{ll}(\mathtt{concede})$),
  - the last deals proposed by the players (denoted $\mathtt{lo}(p,d)$),
  - the previous deals proposed by the players (denoted $\mathtt{po}(p,d)$),
  - the deals which have been already (and implicitly) rejected by the interlocutor (denoted $\mathtt{rejected}(d)$) ;
- a set of assumptions in $\mathcal{A}sm$ representing that some deals have not been yet rejected (denoted $\mathtt{notrejected}(d)$), that some deals have not been proposed in the previous moves (denoted $\mathtt{notpo}(p,d)$) and that a number of standstills has not been reached (e.g. $\mathtt{notnbss}(3)$).

The **preference** relation $\mathcal{P}$ on the goals in $\mathcal{G}$ has been extended in order to take into the new goals respond and optimal. By adopting the MC strategy, the agent tries to utter the "optimal" utterances, optimal. If the agent cannot reach this goal, then the agents responds with a legal move, $\mathtt{optimal}\mathcal{P}\mathtt{respond}$ and $\mathtt{respond} \in \mathtt{RV}$. Since this decision framework (in particular the rules) depends on the dialogue state of the history $h$, we denote it by $\mathtt{DF}_h = \langle \mathcal{L}, \mathcal{G}, \mathcal{D}, \mathcal{B}, \mathcal{R}_h, \mathcal{A}sm, \mathcal{C}on, \mathcal{P} \rangle$.

Some inference rules of the agent $\mathrm{ag}_1$ (which plays the role of the initiator) are depicted in Tab. 2. The additional rules are depicted in Tab. 3. These rules are related to the dialogue state after the move $\mathtt{mv}_2$ (1-6) or the negotiation strategy (7-18). While one of the players starts by asserting a first proposal (7), the other agent replies with a counter-proposal (8). An agent must adopt one of these attitudes : i) either it **stands still**, i.e. it repeats its previous proposal ; ii) or it **concedes**, i.e. it withdraws to put forward one of its previous proposal and it considers another one. In order to articulate these attitudes, the MC strategy consists of adhering the reciprocity principle during the negotiation. If the interlocutor stands still, then the agent will stand still (13). Whenever the interlocutor has made a concession, it will reciprocate by conceding as well (11). If the agent is not able to concede (e.g. there is no other deals which satisfy its constraints), the agent will standstill (12). It is worth noticing that the third step in the negotiation has a special status, in that the player has to concede (9). If the agent is not able to concede (e.g. there is no other deal which satisfies its constraints), the agent will standstill (10). If an acceptable offer has been put forward by the interlocutor, the player accepts it (16-18). When the player can no more concede, it stops the negotiation (15). It is worth noticing that, contrary to [7], our strategy does not stop the ne-

gotiation after 3 consecutive standstills but the strategy allows to concede after them (14). Moreover, any previous offer of the interlocutor can accepted. As we will see in the next section, this will allow a negotiation to succeed even if, contrary to [7], an agent does not know the preferences and the reservation value of the other agent. The inference rules of the part are similar.

Differently from [7], we do not assume that the agents know the preferences of their interlocutors. Therefore, we say that a decision is a **minimal** concession for a speaker since there is no other deal which has not been already (and implicitly) rejected by the interlocutor and which is preferred by the speaker.

**Definition 9 (Minimal concession)** *Let* $DF = \langle \mathcal{L}, \mathcal{G}, \mathcal{D}, \mathcal{B}, \mathcal{R}, \mathcal{A}sm, \mathcal{C}on, \mathcal{P} \rangle$ *be a decision framework as defined in Section 5. The decision* $d$ *is a* ***concession*** *wrt* $d'$ *iff there exists a set of decisions* $D$ *such that* $d \in D$ *and for all* $D' \subseteq \mathcal{D}$ *with* $d' \in D'$*, it is not the case that* $D \mathcal{P} D'$*.*
*The decision* $d$ *is a* ***minimal concession*** *wrt* $d'$ *iff it is a concession wrt* $d'$ *and there is no* $d'' \in \mathcal{D}$ *such that*
*–* $d''$ *is a concession wrt* $d'$*, and*
*–* *there is* $D'' \subseteq \mathcal{D}$ *with* $d'' \in D''$ *with* $D'' \mathcal{P} D$*.*

The minimal concessions are computed by the decision framework proposed in this section. In our example, the agent $\mathrm{ag}_1$ concedes not to grab the third resource after the move $\mathrm{mv}_1$, since its first proposal has been rejected.

The MC strategy has been implemented by means of MARGO[2] [16] (Multiattribute ARGumentation framework for Opinion explanation), an argumentation-based engine for decision-making adopting the assumption-based approach of argumentation [3]. It is written in Prolog and its distributed under the GNU GPL. MARGO is built on top of CaSAPI[3] [8] (Credulous and Sceptical Argumentation : Prolog Implementation), a general-purpose tool for (several types of) assumption-based argumentation which is also written in Prolog.

## 7 Properties

The negotiation protocol, as well as the MC strategy, has useful properties. The negotiations always terminate. Moreover, if both players adopt the MC strategy, the negotiation is successful, when it is possible. Finally, the outcome is optimal.

Due to the finiteness assumption of the language, and hence the finiteness of possible decisions, the set of histories is also finite. Hence it is immediate that the negotiations always terminate.

**Theorem 1 (Termination)** *The dialogues are finite.*

Due to the finiteness assumption and the definition of the MC strategy over the potential agreements, it is not difficult to see that such negotiations are successful, if a potential agreement exists. The final agreement of the negotiation is said to be a Pareto optimal if it is not possible to strictly improve the individual welfare of an agent without making the other worse off. This is the case of our realisation of the MC strategy in a bilateral bargaining.

**Claim 1 (Outcome)** *If both players adopt a MC strategy and a potential agreement exists, then the dialogue is a success and the outcome is Pareto optimal.*

Differently from [7], a player will concede at a certain point even if its interlocutor stands still since it can no more concede. Therefore, the negotiation between two players adopting the MC strategy go throw the whole sets of acceptable deals. In our example, $\mathrm{d}([0,1,0],[0,0,1])$, which is Pareto optimal, is the outcome of the successful dialogue.

Differently from [7], our realisation of the MC strategy allows to reach an agreement even if the agents do not know the preferences and the reservation value of the other agents. However, this realisation of the MC strategy is not in a pure symmetric Nash equilibrium.

## 8 Related works

Rahwan et al. [21] propose an analysis grid of strategies for agents engaged in negotiations. According to this grid, the factors which influence our strategy are : the goals (an optimal outcome here), the domain (represented in terms of multi-attribute choice here), the negotiation protocol, the abilities of agents (their resources

[2] http ://margo.sourceforge.net
[3] http ://casapi.sourceforge.net

$$
\begin{array}{rcll}
\texttt{ll}(\texttt{reply}) & \leftarrow & & (1)\\
\texttt{nbss}(0) & \leftarrow & & (2)\\
\texttt{po}(p,d) & \leftarrow & \texttt{lo}(p,d) & (3)\\
\texttt{lo}(\texttt{init},\texttt{d}([0,0,0],[0,0,1])) & \leftarrow & & (4)\\
\texttt{lo}(\texttt{part},\texttt{d}([1,1,0],[0,0,0])) & \leftarrow & & (5)\\
\texttt{rejected}(d) & \leftarrow & \texttt{po}(\texttt{init},d) & (6)\\
\texttt{optimal} & \leftarrow & \texttt{loc}(\texttt{assert}),\texttt{ll}(\texttt{none}) & (7)\\
\texttt{optimal} & \leftarrow & \texttt{loc}(\texttt{reply}),\texttt{ll}(\texttt{assert}) & (8)\\
\texttt{optimal} & \leftarrow & \texttt{loc}(\texttt{concede}),\texttt{d}(x,y),\texttt{ll}(\texttt{reply}), & \\
 & & \texttt{notrejected}(\texttt{d}(x,y)),\texttt{notpo}(\texttt{part},\texttt{d}(x,y)) & (9)\\
\texttt{respond} & \leftarrow & \texttt{loc}(\texttt{standstill}),\texttt{d}(x,y),\texttt{ll}(\texttt{reply}), & \\
 & & \texttt{po}(\texttt{init},\texttt{d}(x,y)) & (10)\\
\texttt{optimal} & \leftarrow & \texttt{loc}(\texttt{concede}),\texttt{d}(x,y),\texttt{ll}(\texttt{concede}), & \\
 & & \texttt{notrejected}(\texttt{d}(x,y)),\texttt{notpo}(\texttt{part},\texttt{d}(x,y)) & (11)\\
\texttt{respond} & \leftarrow & \texttt{loc}(\texttt{standstill}),\texttt{d}(x,y),\texttt{ll}(\texttt{concede}), & \\
 & & \texttt{lo}(\texttt{init},\texttt{d}(x,y)) & (12)\\
\texttt{optimal} & \leftarrow & \texttt{loc}(\texttt{standstill}),\texttt{ll}(\texttt{standstill}), & \\
 & & \texttt{notnbss}(3) & (13)\\
\texttt{optimal} & \leftarrow & \texttt{loc}(\texttt{concede}),\texttt{d}(x,y),\texttt{ll}(\texttt{standstill}), & \\
 & & \texttt{notrejected}(\texttt{d}(x,y)), & \\
 & & \texttt{notpo}(\texttt{part},\texttt{d}(x,y)),\texttt{nbss}(3) & (14)\\
\texttt{respond} & \leftarrow & \texttt{loc}(\texttt{reject}),\texttt{d}(x,y),\texttt{ll}(\texttt{standstill}), & \\
 & & \texttt{lo}(\texttt{part},\texttt{d}(x,y)), & \\
 & & \texttt{nbss}(3) & (15)\\
\texttt{optimal} & \leftarrow & \texttt{loc}(\texttt{accept}),\texttt{d}(x,y),\texttt{ll}(\texttt{reply}), & \\
 & & \texttt{po}(\texttt{part},d(x,y)) & (16)\\
\texttt{optimal} & \leftarrow & \texttt{loc}(\texttt{accept}),\texttt{d}(x,y),\texttt{ll}(\texttt{concede}), & \\
 & & \texttt{notrejected}(\texttt{d}(x,y)), & \\
 & & \texttt{po}(\texttt{part},\texttt{d}(x,y)) & (17)\\
\texttt{optimal} & \leftarrow & \texttt{loc}(\texttt{accept}),\texttt{d}(x,y),\texttt{ll}(\texttt{standstill}), & \\
 & & \texttt{notrejected}(\texttt{d}(x,y)), & \\
 & & \texttt{po}(\texttt{part},\texttt{d}(x,y)),\texttt{nbss}(3) & (18)
\end{array}
$$

TAB. 3 – The additional inference rules of the agent $\text{ag}_1$ (which plays the role of the initiator) after the move $\texttt{mv}_2$

here), the values (promoted by the reciprocity principle here). While the strategy of our agents is directly influenced by the behaviour of its interlocutor, it is not clear how to situate this factor in the analysis grid of [21].

Few concrete strategies of agents engaged in negotiations have been proposed. More works are concerned by dialogues with theoretical issues rather than practical issues. In particular, some works aim at formalizing and implementing communication strategies for argumentative agents, specifying how an agent selects a move according to the dialogue state and the arguments it has. For instance, Amgoud and Parsons [2] define different attitudes : an agent can be agreeable/disagreeable, open-minded/argumentative or an elephant's child, depending on the the legal moves and their rational conditions of utterance. Differently from [2], our strategy takes into account also the overt behaviour of the interlocutor, since this strategy is based on the reciprocity principle. More attitudes have been proposed in [19] (credulous, skeptical, cautious) based on the various degrees of justification captured by these different semantics of abstract argumentation. In this paper, we claim that, in negotiations, the different semantics allow us to distinguish risk-taking agents and risk-averse agents. In [2, 19], some properties of these strategies have been studied, such as the existence/determinism of the responds of these strategies, as well as the impact of these attitudes on the result and the termination and the complexity of the dialogue. In this paper, we have similar results expected for the complexity. The main difference between the work in [2, 19] and our work is the type of dialogues which are considered. While [19] focus on theoretical dialogues, i.e. with discursive purposes, we are interested on bilateral bargaining dialogues between parties which aim at reaching a practical agreement.

Alternatively, Kakas et al. [13, 14] consider the argumentation-based mechanism for decision-making [10] implemented in GORGIAS [4] to perform the communication strategy of agents which depends on the agent knowledge, roles, context and possibly on dynamic preferences. The work of Kakas, Maudet and Moraitis is guided by the requirements for communication strategies of an expressive and declarative language which is directly implementable. The Agent Argumentation Architecture model we have proposed in [18] shares with [11] (a) the vision of argumentative deliberation for internal agent modules and (b) the assumption that an agent can prioritize its needs. This paper focus on a simple strategy and the study of its properties in game-theoretical terms.

## 9 Conclusions

In this paper we have presented a realisation of the minimal concession strategy which applies argumentation for generating and evaluating proposals during negotiations. According to this strategy, agents start the negotiation with their best proposals. During the negotiation, an agent may concede or stand still. It concedes minimally if the other agent has conceded in the previous step, or after the optimal offers for the participants have been put forward. It stands still if the other agent has stood still in the previous step. A concession is minimal for a speaker since there is no other alternative which has not been already (and implicitly) rejected by the interlocutor, and which is preferred by the speaker. Our realisation of the minimal concession strategy has useful properties : the outcome of the negotiation, which is guaranteed to terminate, is optimal when it is possible, even if the agents ignore the preferences and the reservation values of the other agents.

## Acknowledgements

We would like to thank the anonymous reviewers for their comments on this paper.

## Références

[1] Leila Amgoud, Yannis Dimopoulos, and Pavlos Moraitis. A unified and general framework for argumentation-based negotiation. In *Proc. of AAMAS*, pages 963–970, 2007.

[2] Leila Amgoud and Simon Parsons. Agent dialogues with conflicting preferences. In *Proc. of ATAL*, volume 2333 of *LNCS*, pages 190–205. Springer, 2002.

[3] Andrei Bondarenko, Francesca Toni, and Robert Kowalski. An assumption-based framework for non-monotonic reasoning. In *Proc. of LPNMR*, pages 171–189. MIT Press, 1993.

[4] Neophytos Demetriou and Antonis C. Kakas. Argumentation with abduction. In *Proc. of the 4th Panhellenic Symposium on Logic*, 2003.

[5] Yannis Dimopoulos, Pavolis Moraitis, and Leila Amgoud. Characterizing the outcomes of argumentation-based integrative negotiation. In *Proc. of IAT*, pages 456–460, 2008.

[6] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2) :321–357, 1995.

[7] Phan Minh Dung, Phan Minh Thang, and Francesca Toni. Towards argumentation-based contract negotiation. In *Proc. of COMMA*, pages 134–146. IOS Press, 2008.

[8] Dorian Gartner and Francesca Toni. CaSAPI : a system for credulous and sceptical argumentation. In *Proc. of ArgNMR*, pages 80–95, 2007.

[9] Nick R. Jennings, Peyman Faratin, Alession R. Lomuscio, Simon Parsons, Carles Sierra, and Michael Wooldridge. Automated negotiation : prospects, methods and challenges. *Int. Journal of Group Decision and Negotiation*, 10(2) :199–215, 2001.

[10] Antonis Kakas and Pavlos Moraitis. Argumentative-based decision-making for autonomous agents. In *Proc. of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 883–890. ACM Press, 2003.

[11] Antonis Kakas and Pavlos Moraitis. Argumentative-based decision-making for autonomous agents. In *Proc. of AAMAS*, pages 883–890. ACM Press, 2003.

[12] Antonis Kakas and Pavlos Moraitis. Adaptive agent negotiation via argumentation. In *Proc. of AAMAS*, pages 384–391, 2006.

[13] Antonis C. Kakas, Nicolas Maudet, and Pavlos Moraitis. Flexible agent dialogue strategies and societal communication protocols. In *Proc. of AAMAS*, pages 1434–1435, 2004.

[14] Antonis C. Kakas, Nicolas Maudet, and Pavlos Moraitis. Layered strategies and protocols for argumentation-based agent interaction. In *Proc. of ArgMAS*, volume 3366 of *LNCS*, pages 64–77. Springer, 2004.

[15] Michael Luck and Peter McBurney. Computing as interaction : agent and agreement technologies. In *Proc. of the 2008 IEEE International Conference on Distributed Human-Machine Systems*, 2008.

[16] Maxime Morge and Paolo Mancarella. The hedgehog and the fox. An argumentation-based decision support system. In *Proc. of ArgMAS*, pages 55–68, 2007.

[17] Maxime Morge and Paolo Mancarella. Assumption-based argumentation for the minimal concession strategy. In *Proc. of ArgMAS*, page 18 pages, 2009.

[18] Maxime Morge and Kostas Stathis. The agent argumentation architecture revisited. In *Proc. of EUMAS*, 2007.

[19] Simon Parsons, Michael Wooldridge, and Leila Amgoud. Properties and complexity of some formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3) :347–376, 2003.

[20] Henry Prakken. Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, 21 :163–188, 2006.

[21] Iyad Rahwan, Peter McBurney, and Liz Sonenberg. Towards a theory of negotiation strategy (a preliminary report). In *Proc. of the AAMAS Workshop on Game Theoretic and Decision Theoretic Agents*, 2003.

# Négociation de Ressources et Maximisation de Bien-être Social

A. Nongaillard*† antoine.nongaillard@lifl.fr
P. Mathieu* philippe.mathieu@lifl.fr

* Univ Lille Nord de France, F-59000 Lille, France
USTL, LIFL, F-59650 Villeneuve d'Ascq, France

† Computer Science and Software Engineering
Concordia University
H3G1M8 Montreal, Québec – CANADA

**Résumé :**
Résoudre un problème d'allocation de ressources au sein d'une communauté d'agents par une approche centralisée possède des inconvénients importants pouvant rendre la solution trouvée inapplicable en pratique. Nous proposons dans ce papier une approche adaptative et "anytime" qui évite les inconvénients des approches centralisées. Cette approche distribuée peut s'appuyer sur n'importe quel genre de réseau d'accointances, utiliser n'importe quelle valeur d'utilité, et permet maximiser trois des notions les plus importantes de la théorie du bien-être social. Pour chacune de ces notions, nous montrons qu'il existe un comportement simple menant le processus de négociation à une allocation socialement optimale par un phénomène d'émergence, ou à une allocation socialement très proche le cas échéant.

**Mots-clés :** Allocation de ressources, émergence, algorithme distribué

**Abstract:**
Centralized approaches have several important drawbacks when employed to solve the resource allocation problem within an agent community. This paper seeks to provide scalable and distributed algorithms to solve this problem, avoiding centralized approach drawbacks. We propose adaptive and "anytime" algorithms which can be applied with any kind of contact network, any range for the utility values, and for the most important social welfare notions. In that purpose, we study various agent behaviors. In each case, we show that there exists a simple behavior which leads the negotiation process to a socially optimal allocation as an emergent phenomenon, or to a socially close allocation if the need arises.

**Keywords:** Resource allocation, emergence, distributed algorithm

## 1 Introduction

Le problème d'allocation de ressources est étudié depuis longtemps, aussi bien de manière centralisée que de manière distribuée. Les approches centralisées abordent le problème d'allocation de ressources comme un problème d'optimisation. Dans ces études, le but est de déterminer l'allocation qui maximise la fonction objectif considérée, pour ensuite allouer les ressources de manière adéquate. Les agents reportent leurs préférences sur les ressources à une entité extérieure, à la manière d'un commissaire-priseur, qui détermine alors l'allocation de ressources optimale [9]. Différents auteurs suggèrent des modèles de transactions pour des types d'enchères donnés [4, 22]. Ces approches centralisées ont cependant d'importants inconvénients. D'abord, une connaissance totale de l'ensemble des informations est requise, alors que de nos jours, les gens acceptent de moins en moins de révéler leurs informations personnelles. Ces approches centralisées déterminent la meilleure allocation et allouent les ressources en conséquence, sans jamais se préoccuper de la séquence de transactions nécessaire y amenant. Implicitement, ces études centralisées font l'hypothèse qu'un agent est capable de communiquer avec tous les autres agents du système. Cette hypothèse n'est pas réaliste dans la plupart des applications. De plus, l'allocation fournie peut ne pas être atteignable en pratique par des échanges bilatéraux. Les approches centralisées manquent également d'adaptabilité : une petite variation dans les données, comme par exemple un nouvel agent qui entre dans le système, mène à un redémarrage du processus de résolution. Enfin, de cette manière, seuls des problèmes de petite taille peuvent être résolus.

D'autres approches se basent sur la notion d'agent [14, 24], et résolvent le problème par des processus de négociations distribués. Dans de tels cas, l'allocation de ressources initiale évolue grâce à des négociations locales entre les agents, sans que le problème global d'optimisation ne soit jamais considéré. Des études mathématiques ont été réalisées, étudiant les différentes classes de transactions et les différents types d'optima locaux, et établissant des théorèmes sur l'existence ou non de séquences de transactions menant jusqu'à une allocation optimale [21]. Ces classes de transactions ont été évalué dans [2] mais seulement sur des systèmes de faible taille. Ces travaux ont ensuite été étendu dans [11] qui se focalise sur les longueurs des séquences de transactions. Cepen-

dant, les auteurs ont toujours considérés des possibilités de communication complètes. L'expression et la représentation des préférences des agents ont été étudiées [10], ainsi que les classes de fonctions d'utilité ou les fonctions de paiement ont été étudiées afin de concevoir des processus de négociation qui convergent [8], de même que les critères d'acceptabilité et les propriétés des transactions [12]. En revanche, ces études se focalisent sur une notions particulière du bien-être social, et ne fournissent pas explicitement de processus de négociation. Des comportements d'agents ont aussi été définis, des conditions de négociations favorisant les transactions équitables ont été identifiées [13] et l'absence d'envie durant les processus de négociation a été étudiée [6, 7]. Ces études identifient des conditions menant en théorie les processus de négociation à des allocations optimales, elles ne peuvent pas montrer un chemin de transactions acceptables. Des protocoles de négociations ont été conçus [20], mais toujours sous l'hypothèse de possibilités de communication totales. Dans chacune des ces études, les possibilités de communication des agents sont supposées totales.

Dans cette étude, nous introduisons donc la notion de réseau d'accointances. Un tel graphe représente les relations entre les agents et définit donc leurs possibilités de communication. Ce réseau d'accointances peut correspondre à n'importe quel genre de graphe connexe, allant des graphes complets aux graphes petit-monde [23, 1], incluant les graphes structurés comme les arbres, les anneaux ou les grilles. Nous cherchons à concevoir le comportement individuel le plus simple et le plus efficace selon le rasoir d'Occam afin de favoriser les performances de notre approche. Nous considérons trois des notions les plus importantes de la théorie du bien-être social : le bien-être utilitaire, le bien-être égalitaire et le produit de Nash. Pour chacune de ces notions, nous voulons définir le comportement des agents menant un processus de négociation à une allocation optimale par un phénomène d'émergence, et basé sur n'importe quel genre de réseau d'accointances. Les approches centralisées sont les seules avec qui nous pouvons nous comparer, mais elles n'ont aucun des avantages de notre approche, à savoir la dynamicité, l'évolution du réseau d'accointances, et l'identification d'une séquence de transactions menant à un optimum. Les solutions centralisées n'ont pas de réel intérêt dans la résolution de ce problème et ne sont utilisé qu'à titre de comparaison.

La section 2 décrit une solution centralisée pour chacun des bien-êtres sociaux considérés. Ensuite, la section 3 présente les caractéristiques de l'approche distribuée et discute également les problèmes relatifs à l'évaluation d'un processus de négociation. Enfin, les sections 4, 5 et 6 présentent successivement les résultats obtenus pour chacune des notions du bien-être social considérées ainsi qu'un comportement efficace.

### 1.1 Notations

Soit $\mathcal{P} = \{a_1, \ldots, a_n\}$ une population d'agents et $\mathcal{R} = \{r_1, \ldots, r_m\}$ un ensemble de ressources supposées indivisibles. Chaque agent $a$ de la population possède un ensemble de voisins $\mathcal{N}_a$, ainsi qu'un panier de $m_a$ ressources $\mathcal{R}_a$. Une allocation de ressources est un partitionnement de toutes les ressources de $\mathcal{R}$ parmi les agents de la population $\mathcal{P}$. Notons $\mathcal{A}$ l'ensemble de toutes les allocations de ressources possibles. Une allocation peut être décrite en utilisant l'ensemble des ressources possédé par chaque agent :

$$A = [\mathcal{R}_1, \ldots, \mathcal{R}_n].$$

Les préférences des agents sont exprimées par le biais d'une fonction d'utilité :

$$u_a : \mathcal{R} \to \mathbb{R}, \qquad a \in \mathcal{P}.$$

Cette fonction est supposée normalisée et additive [18]. Les propriétés suivantes doivent donc être satisfaites :

$$\text{normalisée} \Leftrightarrow \quad u_a(\emptyset) = 0, \qquad a \in \mathcal{P}$$

$$\text{1-additive} \Leftrightarrow \; u_a(\mathcal{R}_a) = \sum_{r \in \mathcal{R}_a} u_a(r), \quad a \in \mathcal{P}$$

Dans certains travaux, les paiements compensatoires sont parfois autorisés durant les négociations. Cependant, même si l'emploi de l'argent est toujours contraint (pas de création d'argent durant une transaction), le budget des agents n'est bien souvent pas limité, ce qui n'est pas une hypothèse réaliste. Sans cette limite de budget, il est relativement facile d'assurer artificiellement n'importe quel critère d'acceptabilité, mais avec celle-ci, il n'est plus possible d'assurer qu'une allocation socialement optimale est atteignable. Les questions relatives aux paiements compensatoires vont au-delà du sujet de cette étude et ne seront donc pas considérées par la suite.

### 1.2 Le bien-être social

La théorie du bien-être social (*social welfare*)

est utilisée pour évaluer un système multi-agent dans son ensemble [3, 19]. Dans cette étude, trois des notions principales de la théorie du bien-être social sont considérées.

Tout d'abord, considérons le bien-être social utilitaire. Celui-ci est facilement calculable et correspond à la notion la plus utilisée dans la littérature. Le bien-être de toute la société est considéré, sans prise en compte du bien-être individuel.

**Définition 1** (Bien-être utilitaire)**.** Le bien-être utilitaire d'une allocation de ressource $A$, noté $sw_u(A)$, est la somme de l'utilité de tous les agents :

$$sw_u(A) = \sum_{a \in \mathcal{P}} u_a(\mathcal{R}_a).$$

A l'opposé, le bien-être égalitaire se focalise uniquement sur le bien-être individuel. Il est souvent employé pour réduire les inégalités au sein d'une population.

**Définition 2** (Bien-être égalitaire)**.** Le bien-être égalitaire d'une allocation de ressources, $A$, noté $sw_e(A)$, correspond à l'utilité de l'agent le plus pauvre :

$$sw_e(A) = \min_{a \in \mathcal{P}} u_a(\mathcal{R}_a).$$

Enfin, le produit de Nash peut être vu comme un compromis entre les bien-êtres utilitaire et égalitaire. En dépit de ses qualités, cette notion est rarement employée du fait de sa non linéarité et du manque de méthode efficace.

**Définition 3** (Produit de Nash)**.** Le produit de Nash d'une allocation de ressources $A$, noté $sw_N(A)$, correspond au produit de l'utilité de tous les agents :

$$sw_N(A) = \prod_{a \in \mathcal{P}} u_a(\mathcal{R}_a).$$

**Exemple 1.** Afin d'illustrer la différence entre ces notions et leurs impacts sur la distribution des ressources, considérons une population de trois agents $\mathcal{P} = \{a_1, a_2, a_3\}$ ainsi qu'un ensemble de six ressources, $\mathcal{R} = \{r_1, r_2, r_3, r_4, r_5, r_6\}$. Les préférences de chaque agent sont exprimées au moyen d'une fonction d'utilité, décrite dans la table 1.

Pour chacune des notions de bien-être considérées, la valeur sociale optimale et une allocation de ressources correspondante sont présentées dans le tableau 2.

TAB. 1 – Préférences des agents

| Ressources | Agents | | |
|---|---|---|---|
| | $a_1$ | $a_2$ | $a_3$ |
| $r_1$ | 10 | 6 | 1 |
| $r_2$ | 7 | 10 | 2 |
| $r_3$ | 10 | 3 | 1 |
| $r_4$ | 9 | 4 | 2 |
| $r_5$ | 2 | 8 | 1 |
| $r_6$ | 1 | 6 | 3 |

TAB. 2 – Valeur sociale optimale

| Bien-être social | Valeur | Allocation de ressources |
|---|---|---|
| $sw_u$ | 53 | $[\{r_1, r_3, r_4\}\{r_2, r_5, r_6\}\{\}]$ |
| $sw_e$ | 6 | $[\{r_1\}\{r_5\}\{r_2, r_3, r_4, r_6\}]$ |
| $sw_N$ | 1800 | $[\{r_1, r_3\}\{r_2, r_5\}\{r_4, r_6\}]$ |

Ces notions de bien-être ont des conséquences différentes sur la distribution des ressources parmi les agents de la population. Par exemple, dans le cas du bien-être utilitaire, une allocation socialement optimale peut complètement négliger un agent, $a_3$ en l'occurrence, et ne pas lui fournir la moindre ressource. Une telle situation survient en particulier lorsque pour chacune des ressources, il existe un autre agent qui y associe une plus grande valeur d'utilité. Le bien-être égalitaire n'a pas cet inconvénient puisqu'un si un des agents n'a pas de ressources, la valeur du bien-être égalitaire sera nulle. Ainsi, si pour chaque agent il existe au moins une ressource distincte dont l'utilité associée n'est pas nulle, chaque agent peut obtenir une ressource. Cependant, un agent peut drainer les ressources si il a de "faibles préférences" comme l'agent $a_3$. La distribution des ressources au sein de l'allocation finale peut alors être très inégale. Le produit de Nash maximise l'utilité globale tout en favorisant une réduction des inégalités. Cela peut mener à des allocations de ressources plus équilibrées, tout en évitant de négliger un agent et le drainage des ressources. Cette dernière notion ne peut toutefois être utilisée que lorsque les valeurs d'utilité sont strictement positives.

## 2 Les approches centralisées

Le problème d'allocation de ressources peut être bien évidemment résolu à l'aide d'algorithmes centralisés, quelque soit la notion de bien-être social considérée. De telles approches allouent simplement les ressources tout en maximisant la fonction sociale, mais ne prennent pas en

considération la séquence de transactions nécessaire pour y arriver, font l'hypothèse d'un réseau d'accointances complet et ne sont pas dynamique. La valeur sociale donnée par ces approches correspond à un *optimum global*.

**Définition 4** (Optimum global)**.** Une allocation de ressources est un *optimum global* si il n'existe pas d'autre allocation de ressources associée à une plus grande valeur de bien-être social. Un optimum global est indépendant des transactions autorisées entre les agents. De plus, une telle valeur sociale est unique, mais peut correspondre à plusieurs allocations de ressources.

Ces approches centralisées sont décrites et utilisées pour que l'on puisse comparer nos négociations distribuées, mais elles n'ont aucun intérêt en elle-même pour la résolution du problème évoqué.

### 2.1 Cas utilitaire

La maximisation du bien-être utilitaire peut être modélisée par un système d'équations utilisant les variables $x_{ra}$ ($r \in \mathcal{R}, a \in \mathcal{P}$), qui traduisent la possession d'une ressource $r$ par un agent $a$.

$$sw_u^\star = \begin{cases} \max \sum_{a\in\mathcal{P}} \sum_{r\in\mathcal{R}} u_a(r) x_{ra} \\ \text{s.t. : } \sum_{a\in\mathcal{P}} x_{ra} = 1 \quad r \in \mathcal{R} \\ \quad\quad x_{ra} \in \{0,1\} \quad r \in \mathcal{R}, a \in \mathcal{P}. \end{cases}$$

Une autre méthode plus simple peut également être employée : elle se base sur une propriété que doit satisfaire une allocation utilitaire optimale.

**Proposition 1** (Optimum utilitaire et allocation de ressources)**.** Dans une allocation socialement optimale, chaque ressource est allouée à un des agents qui lui associe la plus grande valeur d'utilité lorsque le bien-être utilitaire est considéré,

*Démonstration.* Faisons une preuve par contradiction. Considérons une allocation $A$ que l'on suppose être un optimum global, et dans laquelle un agent $a$ possède une ressource $r$. Supposons maintenant qu'un agent $a' \in \mathcal{P} \setminus \{a\}$ associe à $r$ une plus grande valeur d'utilité. Plus formellement :

$$\exists r \in \mathcal{R} \text{ et } a, a' \in \mathcal{P} \quad \text{tels que} \quad u_{a'}(r) > u_a(r).$$

$A'$ correspond à une allocation dans laquelle $r$ est allouée à l'agent $a'$.

$$\begin{aligned} sw_u(A) &= \sum_{a\in\mathcal{P}} u_a(\mathcal{R}_a) \\ &= u_a(\mathcal{R}_a) + u_{a'}(\mathcal{R}_{a'}) + \dots \\ &< u_a(\mathcal{R}_a) - u_a(r) + u_{a'}(\mathcal{R}_{a'}) + u_{a'}(r) + \dots \\ &< sw_u(A'). \end{aligned}$$

Le bien-être utilitaire associé à $A'$ est plus grand que celui associé à $A$, donc l'allocation $A$ ne peut être un optimum global. Ainsi, toute allocation de ressources n'allouant pas chacune des ressources à l'un des agents lui attribuant la plus grande valeur d'utilité ne sera pas un optimum global. □

Une allocation maximisant le bien-être utilitaire peut être facilement trouvée grâce à l'algorithme 1 de manière triviale : chacune des ressources est allouée à l'agent qui lui associe la plus grande utilité.

**Algorithme 1** Algorithme centralisé pour le bien-être utilitaire

```
1: pour tout r ∈ R do
2:    a ← arg max_{a∈P} u_a(r)
3:    Allouer r à a
4: fin pour
```

### 2.2 Cas égalitaire

En utilisant les mêmes variables $x_{ra}$ ($r \in \mathcal{R}, a \in \mathcal{P}$) traduisant la possession d'une ressource $r$ par un agent $a$, la maximisation du bien-être égalitaire correspond au système suivant :

$$sw_e^\star = \begin{cases} \max \min_{a\in\mathcal{P}} \sum_{r\in\mathcal{R}} u_a(r) x_{ra} \\ \text{s.t. : } \sum_{a\in\mathcal{P}} x_{ra} = 1 \quad r \in \mathcal{R} \\ \quad\quad x_{ra} \in \{0,1\} \quad r \in \mathcal{R}, a \in \mathcal{P}. \end{cases}$$

Ce programme linéaire peut se résoudre de manière exacte en utilisant un logiciel tel que CPLEX[17] si les agents acceptent de divulguer leurs informations personnelles. En revanche, cette méthode n'est pas applicable pour les grands jeux de données.

### 2.3 Cas du produit de Nash

La maximisation du produit de Nash peut se formuler au moyen d'un système d'équations si-

milaires utilisant les mêmes variables $x_{ra}$ ($r \in \mathcal{R}, a \in \mathcal{P}$).

$$sw_N^\star = \begin{cases} \max \prod_{a\in\mathcal{P}} \sum_{r\in\mathcal{R}} u_a(r)x_{ra} \\ \text{s.t. : } \sum_{a\in\mathcal{P}} x_{ra} = 1 \quad r \in \mathcal{R} \\ \quad\quad x_{ra} \in \{0,1\} \quad r \in \mathcal{R}, a \in \mathcal{P}. \end{cases}$$

La fonction objectif d'un tel problème n'est ni linéaire, ni convexe, ni concave. Théoriquement, une estimation peut être faite grâce à une combinaison de méthodes mathématiques. Une relaxation *Lagrangienne* [15] combinée à un algorithme à départs multiples [16] peuvent être employés. Initialiser l'algorithme à partir de solutions initiales différentes évite parfois les optima locaux quand la fonction considérée n'est pas convexe. De plus, puisque les ressources ne sont pas divisibles, une solution entière doit encore être trouvée. En effet, la relaxation transforme l'intervalle discret $\{0,1\}$ des valeurs des variables, en un intervalle discret $[0,1]$. Dans ce but, un algorithme de *branch-and-bound* doit être utilisé, guidé par les valeurs fournies par la solution relaxée.

Une telle méthode ne peut garantir le caractère optimal de l'allocation obtenue tout en étant extrêmement gourmande en temps de calcul. Cela est du à la combinaison du caractère non-linéaire de la fonction objectif et de la taille exponentielle de l'espace des solutions. Par exemple, résoudre un simple système avec 25 agents est équivalent à optimiser une somme de produits ayant 25 facteurs chacun.

Puisque cette méthode n'est pas utilisable en pratique, nous avons développé des heuristiques afin d'estimer la valeur sociale optimale. Seule l'heuristique donnant les meilleurs résultats est décrite et utilisée par la suite. Décrite par l'algorithme 2, il se focalise sur la valeur des ressources. La première étape est donc d'allouer chacune des ressources à l'agent lui associant la plus grande utilité. Ensuite, la seconde partie de l'algorithme s'assure que tous les agents aient au moins une ressource. Dans le cas contraire, il recherche parmi les agents ayant au moins deux ressources, la ressource maximisant le produit de leurs utilités.

**Algorithme 2** Heuristique centralisée pour le produit de Nash

**pour tout** $r \in \mathcal{R}$ **do**
  $a \leftarrow \arg\max_{a\in\mathcal{P}} u_a(r)$
  Allouer $r$ à $a$
**fin pour**
**pour** $a \in \mathcal{P}$ tq $m_a = 0$ **do**
  $val \leftarrow 0$
  $r' \leftarrow \emptyset$
  $a'' \leftarrow \emptyset$
  **pour** $a' \in \mathcal{P}$ tq $m_a > 1$ **do**
    $r' \leftarrow \arg\max_{r\in\mathcal{R}_{a'}} u_a(r')u_{a'}(\mathcal{R}_{a'} \setminus \{r'\})$
    **si** $val < u_a(r')u_{a'}(\mathcal{R}_{a'} \setminus \{r\})$ **alors**
      $val \leftarrow u_a(r)u_{a'}(\mathcal{R}_{a'} \setminus \{r\})$
      $r \leftarrow r'$
      $a'' \leftarrow a'$
    **fin si**
    Allouer $r$ à $a''$
  **fin pour**
**fin pour**

## 3 Négociations distribuées

A l'inverse des approches centralisées, l'allocation de ressources initiale évolue petit à petit grâce à des négociations locales entre les agents jusqu'à ce qu'une allocation émerge, ce qui marque alors la fin du processus de négociation. De telles approches distribuées peuvent prendre en compte les différents aspects du problème comme la confidentialité des informations, ou l'applicabilité des solutions trouvées.

Une transaction $\delta_a(A, A')$, initiée par l'agent $a$, est habituellement définie par une paire d'allocations de ressources $(A, A')$, décrivant respectivement l'état du système avant et après cette transaction qui implique un sous-ensemble fini d'agents. En pratique, les agents n'ont pas une vue globale du système multi-agent. Nous supposons donc qu'ils ne connaissent initialement que leurs propres préférences ainsi que la liste de leurs voisins. Ainsi, une transaction doit être définie sur la base d'informations locales seulement. Notons $\mathcal{R}_{a\leftrightarrow a'}$ l'ensemble des ressources échangées entre les agents $a$ et $a'$ durant une transaction.

**Définition 5** (Transaction)**.** Une transaction, initiée par l'agent $a$ et impliquant les agents $a', a'', \ldots$, est une liste d'ensembles de ressources échangées entre l'agent initiateur et chacun des agents impliqués.

$$\delta_a = [\mathcal{R}_{a\leftrightarrow a'}, \mathcal{R}_{a\leftrightarrow a''}, \ldots].$$

### 3.1 Le réseau d'accointances

Contrairement aux approches centralisées où un

agent est supposé pouvoir parler à tous les autres agents, les approches centrés sur les agents peuvent prendre en compte les notions de voisinage et de réseau d'accointances. Le voisinage d'un agent $a$, noté $\mathcal{N}_a$, est la liste des voisins avec lesquels il lui est possible de parler. La plupart des études reposent sur l'hypothèse d'un réseau d'accointances symétrique et complet. Symétrique signifie que si l'agent $a$ connaît l'agent $a'$, alors l'agent $a'$ connaît également l'agent $a$. Un réseau d'accointances complet signifie qu'un agent peut communiquer avec n'importe quel autre agent du système. Cette hypothèse a des conséquences très importante sur la circulation des ressources et sur la qualité des solutions obtenues, même si elle n'est pas très réaliste au regard de la majorité des applications. Par exemple, dans le cas d'un réseau social, une personne ne connaît qu'un sous-ensemble réduit des membres de la communauté. Dans cette étude, nous considérons que le réseau d'accointances peut être n'importe quel genre de graphe connexe, allant des graphes complets aux graphes petit-monde [1, 5], incluant également les graphes structurés comme les anneaux, les arbres ou les grilles.

Selon le genre de transaction autorisé durant les négociations, un processus qui converge vers une allocation optimale dans le cas d'un réseau d'accointances complet, peut très bien s'arrêter sur une allocation sous-optimale dans le cas d'un réseau d'accointances restreint. La connexité moyenne d'un réseau d'accointances est définie ici comme le nombre moyen de voisins par agent.

### 3.2 Le critère d'acceptabilité

Dans [21], l'auteur a prouvé qu'il existe toujours une séquence de contrats simples (achat d'une ressource) non rationnels menant à une allocation optimale. Cependant l'existence d'une telle séquence ne signifie pas pour autant l'arrêt du processus de négociation sur un optimum. En effet, les agents doivent décider par eux-mêmes si une transaction est profitable ou non. Sans une conception appropriée, le processus de négociation ne peut s'arrêter. Dans cette étude, des stratégies individuellement rationnelles ont également été évaluées comme dans [21, 8, 13] par exemple, mais celles-ci menaient toujours à des allocations socialement plus faible. Seul le critère le plus efficace sera décrit et présenté ici.

**Définition 6** (Agent social)**.** Un agent social est un agent qui n'accepte que les transactions qui augmentent la valeur de la fonction sociale considérée.

Une transaction est sociale quand la valeur de la fonction sociale augmente grâce à elle :

**Définition 7** (Transaction sociale)**.** Une transaction, $\delta$, qui change une allocation de ressources initiale $A$ en une nouvelle $A'$, est sociale quand : $sw(A') > sw(A)$.

En pratique, déterminer la valeur de la fonction de bien-être social nécessite des informations sur tous les agents. Cependant, de telles informations ne sont pas disponibles localement dans les systèmes distribués. Néanmoins, il est possible de déterminer l'évolution de cette valeur sociale sur la base d'informations locales. Seul un nombre fini d'agents est impliqué lors d'une transaction donnée, nous pouvons donc considérer tout ceux qui ne sont pas impliqués comme des constantes. Par exemple, appliqué au bien-être utilitaire, une transaction est sociale lorsque la relation suivante est satisfaite :

$$u_a(\mathcal{R}'_a) + u_{a'}(\mathcal{R}'_{a'}) > u_a(\mathcal{R}_a) + u_{a'}(\mathcal{R}_{a'})$$

où $a, a' \in \mathcal{P}$ sont les deux agents impliqués, $\mathcal{R}_a, \mathcal{R}_{a'}$ les paniers de ressources des agents avant la transaction, et $\mathcal{R}'_a, \mathcal{R}'_{a'}$ les paniers après coup.

### 3.3 Transactions

Cherchons maintenant à définir le meilleur comportement individuel, en respectant le principe du rasoir d'Occam. Dans ce but, seules les transactions bilatérales seront considérées, i.e., les transactions n'impliquant que deux agents simultanément. Trois transactions bilatérales de base peuvent être définies, les autres transactions n'étant que des combinaisons de ces trois transactions de base. Dans chaque cas, la transaction est initiée par l'agent $a$, qui implique un de ses voisins $a'$. Chacun d'eux possède respectivement $m_a$ et $m_{a'}$ ressources dans leurs paniers.

Considérons tout d'abord, le *don*. L'agent initiateur $a$ peut donner une unique ressource à son voisin $a'$. Seulement $m_a$ dons sont possibles entre eux.

Ensuite, considérons l'*échange*. Chacun des agents fournit une ressource unique. C'est une transaction symétrique : le nombre de ressources par agent ne varie pas. Par conséquent,

une solution optimale ne peut être atteinte que si l'allocation initiale et au moins une des allocations optimales ont une distribution des ressources similaires. $m_a \times m_{a'}$ échanges sont possibles entre ces deux agents.

Enfin, considérons l'*échange groupé* (EG). Chacun des agents peut fournir un sous-ensemble de son panier de ressources. Contrairement à l'échange, l'échange groupé peut être asymétrique et contient les deux transactions précédentes. $2^{m_a+m_{a'}}$ échanges groupés sont possibles entre les agents $a$ et $a'$. Notons qu'il est possible d'implémenter cette transaction plus efficacement à l'aide d'un programme linéaire par exemple. Mais une telle implémentation requiert alors la centralisation d'information de tout le voisinage, alors que cela va à l'encontre de notre objectif premier.

Quatre stratégies de négociation sont comparées durant les expérimentations des sections 4, 5 et 6 : trois stratégies "pures" utilisant chacune uniquement une des transactions définies précédemment, et une stratégie "mixte", l'échange+don (E+D) dans laquelle l'agent initiateur essaie tout d'abord de trouver un échange acceptable, puis un don acceptable le cas échéant.

### 3.4 Comportement d'agent

Les négociations peuvent être réalisées de différentes manières. En effet, lors d'une négociation initiée par l'agent $a$, il arrive que le participant $a'$ rejette l'offre. Trois alternatives sont alors envisageables pour l'agent initiateur :

– il abandonne et met simplement fin à la négociation
– il change le voisin impliqué
– il change la ressource proposée.

Divers comportements ont été conçus, implémentés et évalués. Dans chacun des ces cas, nous avons néanmoins supposé que chaque agent essaie en premier lieu de donner sa ressource de plus faible utilité. Seul le comportement menant aux meilleurs résultats, décrit dans l'algorithme 3, est présenté dans la suite.

Ce comportement est fugace et volatile, i.e que l'agent initiateur peut aussi bien changer le voisin impliqué que la ressource qu'il propose. Un tel comportement est "complet" : selon les transactions autorisées, si une transaction acceptable existe dans son voisinage, l'agent initiateur l'identifiera. Cependant, cette complétude menant à de meilleurs résultats a un coût. L'évaluation d'un processus de négociation et des comportements ne sont pas des choses évidentes, et sont discutés dans la section suivante.

**Algorithme 3** Comportement fugace et volatile de l'agent initiateur $a$

```
Tri de son ensemble de ressources R_a
pour tout r ∈ R_a do
  pour tout a' ∈ N_a do
    si TEST alors // δ_a est acceptable ?
      APPLIQUER // effectuer δ_a
      Mettre fin à la négociation
    fin si
  fin pour
fin pour
```

### 3.5 Expérimentations et évaluation

L'évaluation d'un processus de négociation n'est pas une question évidente. En effet, il est toujours possible de présenter une métrique permettant de montrer qu'une stratégie est la meilleure selon cette mesure. De nombreuses mesures peuvent être utilisées : le nombre de transactions réalisées, le nombre de ressources échangées, le nombre de tours de parole, le nombre de transactions proposées, l'efficacité sociale ou encore l'écart type entre les valeurs sociales associées aux allocations obtenues à la fin des processus de négociation. Le nombre de transactions tentées et le nombre de tour de parole peuvent estimer le coût communicationnel des stratégies. Les valeurs sociales données par les approches centralisées sont utilisées comme référence pour évaluer la qualité des allocations de ressources fournies par chacune des stratégies testées. Un large écart type signifie que la stratégie de négociation est très sensible à l'allocation de ressources initiales, et que donc la qualité de la solution obtenue varie beaucoup en fonction de l'état initial du système.

Durant les expérimentations, différents réseaux d'accointances ont été générés, des graphes complets, des graphes Erdős-Rényi [5], des grilles, des graphes petit-monde générés par attachement préférentiel (AP) [1], ou encore selon le modèle de Watts et Strogatz [23]. Ces graphes représentent un panel assez large de réseaux d'accointances, avec des caractéristiques qui varient beaucoup, comme le degré moyen de connexité (le nombre moyen de voisins par agent), le coefficient de clustering (quantification de la connexité du voisinage d'un agent) ou

TAB. 3 – Efficacité utilitaire des négociations entre agents ($n = 100$, $m = 500$).

| Type de réseau | Don | Échange | E+D | EG |
|---|---|---|---|---|
| Complet | 100 | 97.6 | 100 | 100 |
| Erdős-Rényi | 98.8 | 95.9 | 99.1 | 98.9 |
| Grille | 71.7 | 68.5 | 76.3 | 66.1 |
| Petit-monde WS | 96.1 | 92.7 | 97.2 | 96.1 |
| Petit-monde PA | 88.8 | 87.8 | 91.0 | 89.2 |

encore la longueur moyenne du chemin le plus court entre deux agents.

Les processus de négociation basés sur des réseaux d'accointances complets sont uniquement utilisés à des fins de comparaisons avec l'approche centralisée, mais n'ont pas d'autre intérêt ici. Les ressources sont distribuées de manière aléatoire. Les préférences des agents sont également générées de manière aléatoire dans l'intervalle {1..100}. Durant les processus de négociations, le tour de parole est distribués de manière uniforme entre les agents. Enfin, pour chacun des paramétrages, un processus de négociations est lancé 100 fois à partir de solutions initiales différentes.

## 4 Négociations entre agents pour le bien-être utilitaire

Lorsque l'on considère le bien-être utilitaire, dans l'algorithme 3, le critère d'acceptabilité se traduit par la relation suivante : TEST := $u_{a'}(r) > u_a(r)$ dans le cas d'un don social, et par TEST := $u_a(r') + u_{a'}(r) > u_a(r) + u_{a'}(r')$ dans le cas d'un échange. Le cas de l'échange groupé est équivalent à celui de l'échange en remplaçant $r$ et $r'$ par des ensembles de ressources ;

Le tableau 3 décrit l'efficacité utilitaire obtenue par les différentes stratégies de négociation selon le genre du réseau d'accointances. Par exemple, un processus de négociation basé sur une grille et utilisant des échanges, mène à des allocations associées à 68.5% de la valeur sociale optimale.

Basé sur des réseaux d'accointances complets, plusieurs stratégies mènent à une allocation optimale. La stratégie la plus simple y parvenant est celle basée sur le don social. Celle-ci est contenue dans les stratégies basées sur l'échange groupé ou sur l'échange+don. Leurs utilisations engendrent donc des coûts additionnels en termes de tour de parole, de transactions proposées ainsi qu'en temps de calcul. Ainsi, la stratégie reposant sur le don social est suffisante pour garantir l'atteinte d'une allocation socialement optimale, lorsque le réseau d'accointances considéré est complet.

**Théorème 4.1.** *Considérons un problème d'allocation de ressources supposées indivisible, dans lequel les agents expriment leurs préférences au moyen d'une fonction d'utilité additive. Un processus de négociation reposant sur un réseau d'accointances complet et utilisant des dons sociaux converge vers un optimum global.*

*Démonstration*. Puisque le réseau d'accointances est complet et connexe, un agent $a$ peut parler à n'importe quel autre agent $a'$ de la population. Pour chacune des ressources $r$ de son panier $\mathcal{R}_a$, si un don social impliquant cette ressource est possible entre $a$ et $a'$, alors $u_{a'}(r) > u_a(r)$ par définition d'une transaction sociale. Il est donc toujours possible de créer une séquence de dons sociaux menant une ressource $r$ dans le panier d'un agent qui lui associe la plus grande valeur d'utilité. Appliqué à toutes les ressources, et selon la proposition 1, l'allocation résultante est un optimum global. □

A l'opposé des approches centralisées, notre approche fonctionne également lorsque le réseau d'accointances est restreint. Il n'est cependant plus possible de garantir l'atteinte d'une allocation optimale. En effet, cela dépend trop de la topologie en elle-même.

Dans le cas de processus de négociations basé sur des réseaux Erdős-Rényi, les valeurs sociales sur lesquelles s'arrêtent les processus de négociations sont très proches, autour de 99% pour le don, l'échange+don et l'échange groupé. Dans le cas de grilles, chaque agent n'a que quatre voisins ce qui réduit beaucoup la circulation des ressources. La stratégie basée sur l'échange+don mène jusqu'à 76.3% de la valeur optimale. Lorsque la connexité moyenne est faible, il est alors utile d'autoriser les transactions échanges afin d'éviter des optima locaux. Dans le cas des graphes petit-monde WS, la stratégie du don mène à 96.1% de la valeur optimale alors que l'échange+don mène à 97.2%. Un coefficient de clustering important aide à la circulation des ressources. Enfin, dans le cas de graphes petit-monde PA, l'échange+don obtient les meilleurs allocations, associées à 84%

TAB. 4 – Performances des négociations entre agents dans le cas utilitaire.

| $\frac{m}{n}$ | $n$=5 | | $n$=25 | | $n$=50 | |
|---|---|---|---|---|---|---|
| 5 | 30ms | 30 | 175ms | 350 | 0.5s | 800 |
| 25 | 90ms | 160 | 1.5s | 1650 | 9s | 4050 |
| 50 | 0.3s | 320 | 13s | 3400 | 1min10 | 8100 |

de l'optimum. Dans de tels graphes, un grand nombre d'agents ne possède qu'un seul voisin. Si ceux-ci ne peuvent pas donner une ressource à leur seul voisin, celle-ci ne peut alors pas circuler, et cela pénalise beaucoup les processus de négociation.

Dans tous les cas, l'écart type reste très faible, ce qui signifie que lorsque le bien-être utilitaire est considéré, l'allocation initiale n'a pas une influence très importante.

Les performances des processus de négociations utilisant le don social sont décrit dans le tableau 4. Ce tableau regroupe le temps de calcul nécessaire à l'arrêt des négociations (coté gauche des cases) ainsi que le nombre de transactions effectuées (coté droit des cases). Un processus impliquant 50 agents qui négocient un total de 2500 ressources disponibles s'arrêtent sur une allocation optimale après seulement une minute et 8000 dons sociaux.

**Théorème 4.2** (Complexité des négociations utilitaires utilisant des dons)**.** *Lors d'un processus de négociation basé sur des dons sociaux, les nombres de transactions réalisées et proposées sont tout deux polynomiaux.*

*Démonstration*. Lorsque le bien-être utilitaire est considéré, un don social $\delta_a^{a'}$ entre deux agents $a$ et $a'$ se donnant une ressource $r$, est caractérisé par la relation $u_{a'}(r) > u_a(r)$. Ainsi, durant une séquence de don sociaux, la valeur d'utilité associée à cette ressource $r$ augmente graduellement avec ses possesseurs successifs. Une ressource $r$ ne peut donc retourner dans les mains d'un détenteur précédent, et donc aucun cycle ne peut apparaître. Une ressource donnée $r$ peut être donnée $n-1$ fois dans le pire des cas. Ainsi le nombre de transactions réalisées est donc borné par $m(n-1) \sim O(nm)$. □

La démonstration de la complexité du point de vue du nombre de transactions tentées dépend de nombreuses caractéristiques comme l'implémentation ou la distribution du tour de parole.

TAB. 5 – Efficacité égalitaire des négociations entre agents ($n = 100$, $m = 500$).

| Type de réseau | Don | Échange | E+D | EG |
|---|---|---|---|---|
| Complet | 66.6 | 15.8 | 99.9 | 99.9 |
| Erdős-Rényi | 66.1 | 14.1 | 86.2 | 89.9 |
| Grille | 61.0 | 12.9 | 80.9 | 81.0 |
| Petit-monde WS | 65.3 | 18.7 | 84.4 | 86.1 |
| Petit-monde PA | 49.1 | 13.9 | 55.6 | 54.4 |

Cependant, si le tour de parole est distribué de manière uniforme, la démonstration peut être faite de la manière suivante :

*Démonstration*. Le nombre maximum de transactions tentées par un agent est de $m(n-1)$. En effet, dans le pire des cas, un agent peut ponctuellement posséder chacune des ressources et essayer de la donner à chacun de ses voisins. Ainsi, le nombre de transactions tentées est borné par $m(n-1)^2 \sim O(n^2 m)$. □

**Solution du cas utilitaire** : la stratégie basée sur le don social est la plus efficace lorsque la connexité moyenne du réseau est importante. Dans l'algorithme 3, on a alors APPLIQUER :="*exécute un don social*". En revanche, si elle diminue beaucoup comme dans le cas des grilles, la stratégie basée sur l'échange+don devient la meilleure alternative. Dans ce cas, dans l'algorithme 3, on a APPLIQUER :="*exécuter un échange+don social*".Toutes deux restent très performantes et peu gourmandes en ressources : elles peuvent être employées pour de larges jeux de données.

## 5 Négociations entre agents pour le bien-être égalitaire

Lorsque l'on considère le bien-être égalitaire, un agent ne peut accepter une transaction qui le rend plus pauvre que son voisin. Dans l'algorithme 3, le test d'acceptabilité est donc : TEST :=$\min[u_a(\mathcal{R}'_a), u_{a'}(\mathcal{R}'_{a'})] > \min[u_a(\mathcal{R}_a), u_{a'}(\mathcal{R}_{a'})]$, où $\mathcal{R}'_a, \mathcal{R}'_{a'}$ sont les paniers de ressources des agents après la transaction.

Le tableau 5 décrit l'efficacité des différentes stratégies selon le genre de réseau d'accointances utilisé.

D'une manière générale, la stratégie du don sociale qui était menait à des allocations proches

TAB. 6 – Performances des négociations entre agents dans le cas utilitaire.

| $\frac{m}{n}$ | $n$=5 | | $n$=25 | | $n$=50 | |
|---|---|---|---|---|---|---|
| 5 | 50ms | 50 | 2s | 1000 | 20s | 3200 |
| 25 | 3s | 350 | 8min | 5000 | 15min | 9000 |
| 50 | 8s | 450 | 10min | 12000 | 45min | 45000 |

de l'optimum dans le cas utilitaire n'obtient pas de bons résultats : après un nombre fini de transactions, les agents ne peuvent plus donner de ressource sans devenir plus pauvre que leur voisin. Une telle stratégie n'est donc plus suffisante pour obtenir des allocations socialement proche de l'optimum. La transaction échange mène quant à elle à des allocations très éloignés (moins de 20%). Puisque la distribution initiale des ressources ne peut être modifié, un agent pauvre qui ne possède que peut de ressources initialement pénalisera les négociations égalitaires. L'échange groupé est une stratégie très efficace, mais son coût exponentiel ne la rend pas attractive. La stratégie basée sur l'échange+don peut être considérée comme la meilleur alternative, et ce quelque soit le réseau d'accointances considéré.

Dans le cas du bien-être égalitaire, il n'est jamais possible de garantir qu'une allocation optimale peut être atteinte. Le tableau 5 montre que plus la connexité moyenne est faible, et plus la valeur sociale atteinte en fin de négociation est faible. Dans le cas de réseau Erdős-Rényi, les négociations basées sur l'échange+don s'arrêtent sur les allocations associées à 86% de la valeur optimale, alors que pour les grilles la valeur sociale atteinte ne représente que 80% de l'optimum, et autour de 84% quand les négociations sont basées sur des petit-monde WS. En revanche, pour ceux générés par attachement préférentiel, les valeurs atteintes varient autour de 55% seulement. Des voisinages trop restreints handicapent beaucoup les négociations égalitaires. L'écart type est d'autant plus élevé que la connexité moyenne est faible : l'allocation initiale à de l'importance.

Les performances de la stratégie de l'échange+don sont évaluées et regroupées dans le tableau 6 puisqu'elle est la plus efficace. Des négociations entre 25 agents impliquant 1250 ressources s'arrêtent après plusieurs minutes alors qu'il faut bien plus de temps pour l'approche centralisée, qui est par ailleurs plus gourmande en ressource mémoire. Les négociations égalitaires nécessitent plus de temps que les négociations utilitaires, mais il est intéressant de noter que la valeur égalitaire augmente fortement au début des négociations (20% du temps mène à plus de 80% de la valeur sociale finale), mais passe beaucoup de temps pour n'améliorer que faiblement cette valeur sociale, comme décrit sur la figure 1.

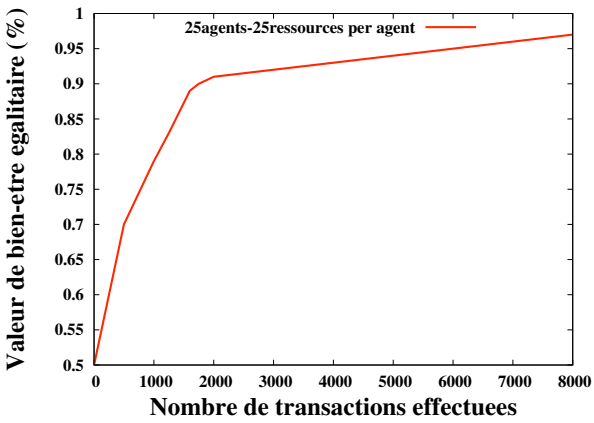



FIG. 1 – Évolution de la valeur du bien-être égalitaire durant le processus de négociation

Indépendemment du réseau d'accointances considérés, une unique situation ne peut être résolue par la stratégie échange+don : une telle situation est nommée "transaction multi-agent" dans la classification établie dans [21]. Considérons 3 agents possédant une ressource chacun : $a_1$ possède $r_1$, $a_2$ possède $r_2$ et $a_3$ possède $r_3$. Si $a_1$ n'est intéressé que par $r_2$, $a_2$ lui-même intéressé uniquement par $r_3$ et $a_3$ uniquement intéressé par $r_1$, alors aucune transaction n'est possible en suivant cette stratégie. Tout agent tentant de donner sa ressource ou de l'échanger, se retrouverait plus pauvre que son voisin, ce qui n'est pas socialement acceptable.

**Solution du cas égalitaire** : la stratégie de l'échange sur l'échange+don est la meilleure alternative et mène à des allocations plus intéressantes en des temps raisonnables. Dans l'algorithme 3, APPLIQUER := "*exécuter un échange+don social*". Alors que les négociations utilitaires peuvent se faire de manière efficace sur des réseaux faiblement connectés, les négociations égalitaires nécessitent une connexité beaucoup plus importante pour être efficace.

TAB. 7 – Efficacité des négociations entre agents pour le produit de Nash ($n = 100$, $m = 500$).

| Genre de réseau | Don | Échange | E+D | EG |
|---|---|---|---|---|
| Complet | 80.2 | 169.2 | 694.1 | 277.2 |
| Erdős-Rényi | 49.6 | 79.9 | 510.0 | 156.5 |
| Grille | 50.9 | 6.4 | 353.2 | 106.7 |
| Petit-monde WS | 178.8 | 22.8 | 549.1 | 253.6 |
| Petit-monde PA | 77.4 | 2.11 | 519.2 | 137.5 |

## 6 Négociations entre agents pour le produit de Nash

Lorsque l'on considère le produit de Nash, le test d'acceptabilité de l'algorithme 3 devient TEST := $u_a(\mathcal{R}'_a)u_{a'}(\mathcal{R}'_{a'}) > u_a(\mathcal{R}_a)u_{a'}(\mathcal{R}'_{a'})$.

Le tableau 7 regroupe les résultats obtenus selon les genres de réseaux d'accointances utilisés. Puisque la valeur servant de référence est obtenue grâce à une heuristique, les processus de négociations entre agents peuvent menés à des allocations dont l'efficacité est supérieure à 100% : la valeur sociale obtenue par notre approche est simplement supérieure à celle donnée par l'heuristique.

La stratégie de l'échange+don amène encore une fois aux meilleurs allocations de ressources, associées avec le plus faible écart type (toujours inférieur à 5% de la valeur totale), indépendemment de la nature du réseau d'accointances. Cette stratégie mène à de meilleures valeurs sociales que l'heuristique même lorsque le réseau d'accointances est très restreint. Les stratégies du don et de l'échange ne sont pas efficace, menant à des allocations très éloignées des optima. La stratégie de l'échange social est très sensible à la connexité moyenne : sur des grilles ou des petit-monde PA, les valeurs sociales obtenues n'excèdent pas 5% de l'estimation heuristique. L'échange groupé amène à de bons résultats mais reste toutefois trop cher pour être considère comme intéressante.

Certaines valeurs peuvent paraître grande (694.1% de l'estimation heuristique). Cependant, l'objectif n'est pas linéaire et un simple échange de deux ressources peut faire augmenter la valeur sociale d'un facteur supérieur à 100.

Les performances des processus de négociation maximisant le produit de Nash utilisant la stratégie échange+don sont décrites dans le tableau 8 en terme de temps de calcul nécessaire avant la fin des négociations ainsi qu'en terme de transactions réalisées. Un processus de négociation entre les 50 agents d'une population négociant 2500 ressources disponibles se termine dans un temps tout à fait raisonnable.

TAB. 8 – Performances des négociations entre agents dans le cas du produit de Nash.

| $\frac{m}{n}$ | $n$=5 | | $n$=25 | | $n$=50 | |
|---|---|---|---|---|---|---|
| 5 | 50ms | 35 | 250ms | 350 | 600ms | 900 |
| 25 | 150ms | 150 | 45s | 1800 | 4min | 4600 |
| 50 | 6s | 400 | 5min | 4000 | 25min | 9000 |

**Solution du cas produit de Nash** : la stratégie de l'échange+don est suffisamment flexible pour mener les processus de négociations à des allocations socialement efficaces quelque soit la nature du réseau d'accointances, et le tout en un temps raisonnable. Dans l'algorithme 3, on a donc APPLIQUER :="*exécuter un échange+don social*".

## 7 Conclusion

Cette étude fournit une méthode pour résoudre efficacement le problème d'allocation de ressources de manière décentralisée. Basée sur des négociations locales entre les agents, cette approche prend en compte les relations entre les agents grâce à la notion de réseau d'accointances, qui peut être n'importe quel type de graphe connexe. La méthode proposée est de plus un processus adaptatif : l'ajout de nouveaux agents est possible durant le processus de négociation, sans pour autant obtenir l'émergence d'une solution de qualité moindre. C'est également une approche "anytime" puisque la qualité de la solution augmente graduellement et que le processus de négociation peut être interrompu à n'importe quel moment tout en fournissant une solution. A l'opposé des approches centralisées, les algorithmes distribués que nous proposons donnent une séquence de transactions acceptables menant à l'allocation fournie, tout en respectant la topologie du réseau d'accointances. Les approches centralisées ne peuvent gérer cet aspect du problème en un temps raisonnable. Même si nous ne pouvons garantir l'atteinte d'un optimum que dans le cas utilitaire sur un réseau complet, notre approche fournie des comportements efficaces et pratiques menant à des allocations socialement proches dans tous les autres cas.

## Références


[1] R. Albert and A. Barabási. Statistical Mechanics of Complex Networks. *Reviews of Modern Physics*, 74(1) :47–97, 2002.

[2] M. Andersson and T. Sandholm. Contract types for satisficing task allocation : II experimental results. In *AAAI Spring Symposium Series : Satisficing Models*, volume 17, pages 1–7, USA, California, Stanford University, March 1998. AAAI Press.

[3] K. Arrow, A. Sen, and K. Suzumura. *Handbook of Social Choice and Welfare*, volume 1. Elsevier, 2002.

[4] M. Bellosta, S. Kornman, and D. Vanderpooten. An Agent-Based Mechanism for Autonomous Multiple Criteria Auctions. In *IAT'06 - Intelligent Agent Technology*, volume 0, pages 587–594, China, Hong-Kong, December 2006.

[5] B. Bollobás. *Random Graphs*. Cambridge University Press, 2001.

[6] S. Bouveret and J. Lang. Efficiency and Envy-Freeness in Fair Division of Indivisible Goods : Logical Representation and Complexity. In *IJCAI'05 - International Joint Conference on Artificial Intelligence*, pages 935–940, UK, Scotland, Edinburgh, Agust 2005.

[7] Y. Chevaleyre, E. Endriss, S. Estivie, and N. Maudet. Reaching Envy-free States in Distributed Negotiation Settings. In *IJCAI'07 - International Joint Conference on Artificial Intelligence*, pages 1239–1244, India, Hyderabad, January 2007. AAAI Press.

[8] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. Negotiating over Small Bundles of Resources. In *AAMAS'05 - Autonomous Agents and multiagent systems*, pages 296–302, EU, The Netherlands, Utrecht, July 2005. ACM Press.

[9] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial auctions*. MIT Press, 2006.

[10] J. Doyle. Prospects for preferences. *Computational Intelligence*, 20(2) :111–136, 2004.

[11] P. Dunne. Extremal behaviour in multiagent contract negotiation. *Journal of Artificial Intelligence Research*, 23 :41–78, 2005.

[12] U. Endriss, N. Maudet, F. Sadri, and F. Toni. Negotiating Socially Optimal Allocations of Resources. *Journal of Artificial Intelligence Research*, 25 :315–348, 2006.

[13] E. Estivie, Y. Chevaleyre, U. Endriss, and N. Maudet. How Equitable is Rational Negotiation ? In *AAMAS'06 - Autonomous Agents and Multiagent Systems*, pages 866–873, Japan, Hakodate, May 2006. ACM Press.

[14] J. Ferber. *Multi-Agent System : An Introduction to Distributed Artificial Intelligence*. Addison Wesley Longman, 1999.

[15] M. Fisher. The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management Science*, 50(13) :1861–1871, 2004.

[16] F. Hickernell and Y. Yuan. A Simple Multistart Algorithm for Global Optimization. *OR Transactions*, 1(2), December 1997.

[17] I. Inc. CPLEX User's Manual. *ILOG Inc.*, 1995.

[18] P. Miranda, M. Grabisch, and P. Gil. Axiomatic Structure of $k$-Additive Capacities. *Mathematical Social Sciences*, 49 :153–178, 2005.

[19] H. Moulin. Choosing from a Tournament. *Social Choice and Welfare*, 3(4) :271–291, 1986.

[20] S. Saha and S. Sen. An Efficient Protocol for Negotiation over Multiple Indivisible Resources. In *IJCAI'07 - International Joint Conference on Artificial Intelligence*, pages 1494–1499, India, Hyderabad, January 2007.

[21] T. Sandholm. Contract Types for Satisficing Task Allocation : I Theoretical Results. In *AAAI Spring Symposium : Satisficing Models*, volume 99, pages 68–75, USA, California, Stanford University, March 1998.

[22] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2) :1–54, 2002.

[23] D. Watts and S. Strogatz. Collective dynamics ofsmall-world'networks. *Nature*, 393(6684) :440–442, 1998.

[24] M. Woolridge. *Introduction to Multiagent Systems*. John Wiley & Sons, 2001.

# Représentation des schémas d'arguments dans un contexte d'aide à la décision multicritère

Wassila Ouerdane
wassila.ouerdane@lamsade.dauphine.fr

LAMSADE
Université Paris-dauphine
75016 Paris – FRANCE

**Résumé :**
Nous proposons dans cette article de construire deux nouveaux schémas d'argument pour la représentation de l'étape agrégation d'un processus d'évaluation dans le cadre d'un processus d'aide à la décision. Ce dernier représente un dialogue entre un client et un analyste dans le but d'arriver à un consensus et de résoudre un problème de décision. Ces deux schémas permettrons de raffiner la structure proposée dans [13]. De plus, nous envisageons de représenter graphiquement les différents échanges durant ce dialogue. Pour cela, nous adaptons à notre contexte la structure de graphe d'argument proposée dans le modèle du Carneades[8].

**Mots-clés :** schémas d'argument, procédure d'agrégation, caractérisation, graphe d'arguments

**Abstract:**
We propose in this article to build two argument schemes in order to represent the aggregation step of an evaluation process within a decision aiding process. The latter represents a dialogue between a client and an analyst in order to reach a consensus and to resolve a decision problem. Such schemes will allow us to refine the structure proposed in [13]. Moreover, we plan to construct a graphical representation of the exchanged arguments during the dialogue. To do that, we suggest to adapt the structure of argument graphs of the Carneades model [8].

**Keywords:** Argument schemes, aggregation procedure, characterisation, argument graphs

## 1 Introduction

Une étape importante dans un processus d'aide à la décision est l'agrégation des préférences du décideur dans le but de construire une recommandation. Dans un contexte avec des critères multiple le développement de cette recommendation nécessite une procédure d'agrégation multicritère. Cette dernière constitue un point crucial dans l'élaboration d'un modèle d'évaluation dans le cadre de ce processus. La description d'une méthode d'agrégation peut être abordée de deux points de vue. D'un coté, l'agrégation peut être considérée comme un opérateur ou un mécanisme qui transforme un ensemble d'informations sur les évaluations des alternatives sur plusieurs dimensions en une information globale et synthétique, souvent représentée par une relation. D'un autre coté la description d'une procédure d'agrégation suit le point de vue de la théorie du mesurage conjoint. Dans ce cas, l'attention n'est pas portée sur les propriétés du mécanisme lui même mais sur les relations de préférences. Si une relation de préférence globale remplit certaines conditions, alors elle admet une représentation numérique d'un modèle (méthode) particulier. En d'autres termes, le mesurage conjoint a pour but d'identifier les conditions nécessaires pour utiliser une procédure d'agrégation.

Parti de ce constat, nous proposons dans cet article de considérer ces deux points de vue dans la définition de schémas d'argument pour la procédure d'agrégation. Le premier portera sur son fonctionnement et le second sur les conditions de son utilisation. Ces deux schémas viennent enrichir l'ensemble des schémas proposés dans [13]. En effet, nous avons présenté dans ce dernier une structure hiérarchique de schémas d'argument dans le but de représenter explicitement les différentes étapes du raisonnement d'un processus d'évaluation basé sur des modèles multicritère. Notre but à travers les schémas d'argument était de répondre à des problèmes liés d'une part à la gestion de la révision et d'autre part à l'absence d'explications formelles durant le processus.

D'un autre coté, le processus d'évaluation est une sorte de dialogue entre deux participants. Nous envisageons de permettre la visualisation des arguments échangés au cours de ce dialogue, sous la forme d'un graphe. Ce dernier peut être utilisé, entre autres, comme un moyen formel pour montrer les conséquences des changements effectués par le décideur sur le modèle d'évaluation (par exemple, l'ajout d'une nouvelle information, la révision un paramètre, etc.). Pour construire ce graphe nous allons nous baser sur la structure proposée dans le modèle du Carneades [8], en introduisant certaines modifications afin de prendre en compte le contexte multicritère.

Le reste du papier est organisé comme suit.

Section 2 offre un bref rappel des éléments de base d'un modèle d'évaluation multicritère. En particulier, nous décrivons le fonctionnement et la caractérisation d'une procédure d'agrégation. Dans la Section 3 nous décrivons l'ensemble des schémas d'argument proposés pour représenter les étapes du processus d'évaluation. Notamment les nouveaux schémas de la phase agrégation. Section 4 porte sur la représentation graphique des arguments échangés dans un dialogue. Cette représentation est basée sur une structure de graphe d'arguments présenté dans cette même section. Dans la Section 5 nous présentons un exemple de dialogue et le résultat graphique. nous concluons par la discussion des perspectives de ce travail.

## 2 Processus d'évaluation multicritère

Un processus d'aide à la décision est un type de dialogue entre un décideur (ou client) et un analyste (ou expert). L'objectif de l'analyste, durant ce processus, est non seulement de guider le décideur pour réussir à résoudre son problème de décision mais aussi de le convaincre des solutions proposées. Ce processus peut être représenté par quatre artefacts : (i) la représentation d'une situation problématique, (ii) la formulation du problème, (ii) le modèle d'évaluation et (iv) la recommandation finale [5, 19].

Dans ce papier nous focalisons sur le modèle d'évaluation. Plus spécifiquement, notre attention portera sur le processus d'évaluation dans sa totalité, où l'analyste se base sur les différents paramètres du modèle pour construire une recommandation. Différentes étapes sont distinguées dans ce processus, par exemple : la comparaisons des actions sur la base d'un point de vue particulier, la selection de la procédure d'agrégation, la construction de la recommendation, etc. Chacune de ces étapes peut faire l'objet de modifications ou de mise à jour afin de prendre en compte les préférences et les besoins du décideur.

Dans ce qui suit, nous rappelons les éléments de base d'un modèle d'évaluation.

### 2.1 Un modèle d'évaluation

Pour une formulation d'un problème de décision donné, l'analyste construit un modèle d'évaluation qui permet d'organiser l'information recueillie auprès du décideur de telle sorte à obtenir par la suite une réponse formelle au problème de décision. Ce modèle est composé essentiellement des éléments suivants.

- **L'ensemble des actions :** la première étape dans la résolution d'un problème de décision consiste à identifier les différents objets soumis au processus d'aide à la décision. Ces objets peuvent être des decision potentielles, des projets, des alternatives, etc. Dans la suite de ce papier, l'ensemble des actions, noté $A$, est considéré fini.
- **Le concept de critère :** Un critère est un outil pour l'évaluation et la comparaison d'actions selon un point de vue bien déterminé. Cette évaluation doit prendre en compte, pour chaque action $a \in A$, l'ensemble des effets ou des attributs rattachés au point de vue considéré. Elle est notée $h(a)$ et appelée la performance de $a$ selon le critère $h$. Formellement,

**Definition 2.1.** ***(le concept de critère)***
*Un critère peut être défini comme étant toute dimension à laquelle il est possible d'associer un modèle de préférence, même partiel, de telle sorte qu'il soit possible pour le décideur de faire un choix entre les actions sur cette dimension.*

En d'autres termes, le critère permet de traduire les préférences du décideur sous la forme d'une relation de préférences sur l'ensemble des actions. Cette relation est souvent représentée par une relation binaire.

**Definition 2.2.** ***(Relation de préférence)***
*Soit $A$ un ensemble d'actions. Une relation de préférences sur l'ensemble $A$ est une relation binaire notée $\succeq$, tel que $\succeq \subset \mathcal{A} \times \mathcal{A}$.*

*Si $x \succeq y$ signifie" $x$ est au moins aussi bonne que $y$" alors : la partie asymétrique de $\succeq$ traduit une préférence stricte ($\succ$: $x \succeq y \wedge \neg(y \succeq x)$) et la partie symétrique traduit une indifférence ($\sim$: $x \succeq y \wedge y \succeq x$) ou une incomparabilité ($?$ : $\neg(x \succeq y) \wedge \neg(y \succeq x)$).*

Dans la plupart des cas, une étape du processus d'aide à la décision consiste à construire $n$ critères avec $n > 1$. Il constitue ce que nous appelons une famille de critères $H = \{h_1, h_2, \ldots, h_n\}$

- **Les méthodes d'agrégation :** est un ensemble d'opérateurs (ou fonction d'agréga-

tion) qui permettent d'obtenir des informations synthétiques sur les élément de $A$ ou de $A \times A$. Un exemple de procédure d'agrégation est présenté dans ce qui suit.

### 2.2 Agrégation et caractérisation

Un des objectifs de la méthodologie d'aide à la décision est d'établir une recommendation basée sur les préférences du décideur. Dans un contexte de critères multiples, la construction d'une telle recommendation nécessite l'utilisation d'une procédure d'agrégation multicritère [18, 5] pour synthétiser les préférences établies sur chaque critère. Cette agrégation est un point important dans la construction du modèle d'évaluation.

Par ailleurs, une tâche délicate pour l'analyste consiste à déterminer le modèle ou le mécanisme d'agrégation le plus adéquat à la situation problématique à résoudre. En effet, selon les préférences du décideur, il est possible d'utiliser certaines procédures et non pas d'autres. La difficulté majeure réside dans le fait qu'il n'est pas évident de savoir, a priori, si le modèle numérique de la procédure permet de représenter le plus fidèlement possible les préférences du décideur. En d'autres termes, faire la correspondance entre les préférences du décideur et un modèle numérique n'est pas une tâche simple. Le problème est dans le fait qu'il n'est pas facile d'avoir une connaissance globale et précise, dès le début du processus, de l'ensemble des informations préférentielles.

Donc, sur quelle base l'analyste peut choisir une procédure en particulier ? Une façon de faire est de s'appuyer sur la caractérisation des procédures d'agrégation établit par *la théorie du mesurage conjoint* [10]. En fait, cette théorie a pour but d'identifier des systèmes d'axiomes qui caractérisent les différentes procédures d'agrégation. Ces axiomes peuvent être utilisés pour spécifier les conditions (sur les structures de préférences) sous lesquelles il est judicieux d'utiliser une procédure donnée.

Dans ce qui suit nous présentons la procédure lexicographique comme exemple de méthode d'agrégation.

**La méthode lexicographique** cette procédure est assez simple d'utilisation et fonctionne comme suit : en premier, les critères doivent être ordonnés selon un ordre linéaire (un ordre sans ex aequo) noté $>_l$ et qu'ils soient considérés dans cette ordre lors de la comparaisons des actions. Considérant, alors, le premier critère selon cette ordre, si $a$ est strictement meilleure que $b$ sur ce critère, alors $a$ est déclarée globalement meilleure que $b$ sans considérer les autres critères. De façon similaire, si $b$ est strictement meilleure que $a$ sur ce critère, alors $b$ est considérée comme globalement meilleure que $a$ sans prendre en considération les autres critères. Maintenant, si $a$ et $b$ sont indifférentes sur ce critère, alors nous passons au deuxième plus important critère. Nous comparons à nouveau $a$ et $b$, si elles sont toujours indifférentes, nous considérons alors le critère suivant et ainsi de suite. Formellement,

$$a \succeq b \Leftrightarrow \begin{cases} a \sim_i b & \text{pour tout les critères} \\ or & \\ a \succ_i b & \text{pour le premier critère } i, \\ & \text{w.r.t l'ordre linéaire,} \\ & \text{pour lequel } a \nsim_i b \end{cases}$$

La particularité de cette procédure est l'existence de l'ordre linéaire $>_l$ et le fait que chaque critère est totalement ou infiniment plus important que tout les autres critères plus bas dans l'ordre. Il n'existe pas de compensation possible entre critères. Pour caractériser cette méthodes trois axiomes sont nécessaires. Ils sont décrits dans ce qui suit.

- Strong Pareto (SP). Si $a$ est strictement meilleure que $b$ sur un certain critère ($a \succ_i b$ pour un certain $i$) et $a$ est au moins aussi bonne que $b$ sur l'ensemble des critères ($a \succeq_i b$ pour tout $i$), alors globalement $a$ est strictement meilleure que $b$. i.e., $a \succ b$. De plus, si $a$ est indifférente à $b$ sur l'ensemble des critères ($a \sim_i b$ pour tout $i$), alors $a$ est indifférente à $b$ sur l'ensemble des critères.i.e. $a \sim b$.
- Independence of Irrelevant Alternatives (IIA). La relation de préférence globale entre $a$ et $b$ dépends uniquement de leur positions dans cette relation et non pas des autres actions.
- Weak Order (WO). La relation de préférence globale est toujours un weak order (a rangement avec possibilité d'ex aequo).

Dans le cas où les préférences du décideur répondent aux trois propriétés citées ci-dessus, alors la procédure qui devra être utilisée est celle de la lexicographie. Inversement, si nous utilisons cette procédure, ces propriétés sont nécessairement satisfaites.

En théorie, l'analyste construit la recommandation en se basant sur les différents paramètres du modèle d'évaluation, présentés cidessus. Ces paramètres représentent la formalisation de l'ensemble des préférences et des informations fournit par le décideur. Durant ce processus, le client peut intervenir pour effectuer des ajustement, des modifications ou ajouter de nouvelles informations, engendrant la reconstruction ou la mise à jour du modèle d'évaluation. En effet, au début du processus, le décideur peut se trouver dans l'incapacité de s'exprimer clairement et complètement sur ses préférences et/ou ses objectifs, ou sur la nature de son problème. Par conséquent, différentes versions du modèle d'évaluation peuvent être construites tout au long du dialogue. Cependant, ces versions sont étroitement liées puisqu'elles portent globalement sur les mêmes informations et la révision ne concerne qu'une petite partie du modèle. Le problème rencontré est qu'il n'existe pas de représentations formelles de comment ces changements interviennent et de comment elles peuvent être gérées dans la pratique par l'analyste. De plus, aucun dispositif n'est mis en place pour confronter le décideur aux conséquences de chaque modification ou mise à jour sur la solution finale. Ce type de moyen peut aider le client à mieux comprendre le processus et les solutions proposées à la fin de celui-ci.

Par ailleurs, un autre aspect qui semble absent dans ce processus est celui de la justification et de l'explication. En effet, le modèle actuel nous fournit aucune indication sur comment produire des explications ou des arguments pour répondre aux doutes du client. En d'autres termes, quel type d'information, un analyste peut utiliser pour construire une justification qui soit à la fois simple, naturelle et compréhensible pour le décideur.

Pour répondre à ces besoins, en termes de révision et d'explication, nous avons proposé dans [13] de construire une structure hiérarchique de schémas d'argument pour représenter les étapes du processus d'évaluation. Dans ce papier, nous proposons d'une part de raffiner cette structure au niveau de l'agrégation pour inclure deux nouveaux schémas d'argument. D'autre part, nous envisageons d'associer au dialogue une représentation graphique des différents échanges.

## 3 Des schémas d'argument pour une évaluation multicritère

Il existe différentes structures ou formes d'arguments. Dans cet article nous avons adopté la structure de schéma d'argument. Selon [22, 23], un schémas d'argument permet d'identifier les prémisses et la conclusion nécessaire à la représentation d'une étape d'un raisonnement donné. De plus, a ce schéma est associé un ensemble de questions critiques. Ces questions permettent d'identifier les différents moyens d'attaquer ou d'invalider ce schéma.

Différentes classes de schémas d'argument sont nécessaires pour représenter le processus d'évaluation. Ces schémas forment, comme le montre la Figure 1, une structure hiérarchique de trois niveaux : (i) *le niveau élémentaire* qui inclut des schémas pour l'évaluation d'une action sur un critère unique ; (ii) *le niveau agrégation* comportant des schémas pour l'agrégation des préférences sur différents critères et (iii) *le niveau multicritère* qui permettra de construire une relation globale entre deux[1] actions.

Par ailleurs, en plus de ces trois niveaux, la structure est partagée en deux cotés : un coté positif et un coté négatif. Cette distinction est justifiée tout simplement par le fait qu'il est très répandu et naturel, face à une situation problématique, de construire des raisons positives et des raisons négatives pour pouvoir prendre une décision. Les informations positives permettent de soutenir la conclusion et de la confirmer. Les informations négatives, quant à elles, sont établies pour annuler ou contredire la conclusion [20, 17, 14]. En outre, en théorie de l'argumentation, plusieurs études et approches ont souligné la possibilité d'avoir cette «bipolarité» (positif vs négatif) dans la construction des arguments [2, 21]. Donc, nous pouvons avoir des arguments en faveur et contre une conclusion.

L'idée générale est qu'au niveau supérieure de la hiérarchie la proposition «$a$ est au moins aussi bonne que $b$» est soutenue lorsqu'il existe assez de raisons positives en sa faveur et qu'aucune raison négative ne vient la contredire. Ce niveau se base sur les conclusions du niveau agrégation qui se base à son tour sur les conclusions du niveau élémentaire. Le détail de ces différents niveaux est présenté dans ce qui suit.

[1] Par souci de simplicité nous considérons dans ce papier un ensembel de deux actions uniquement.

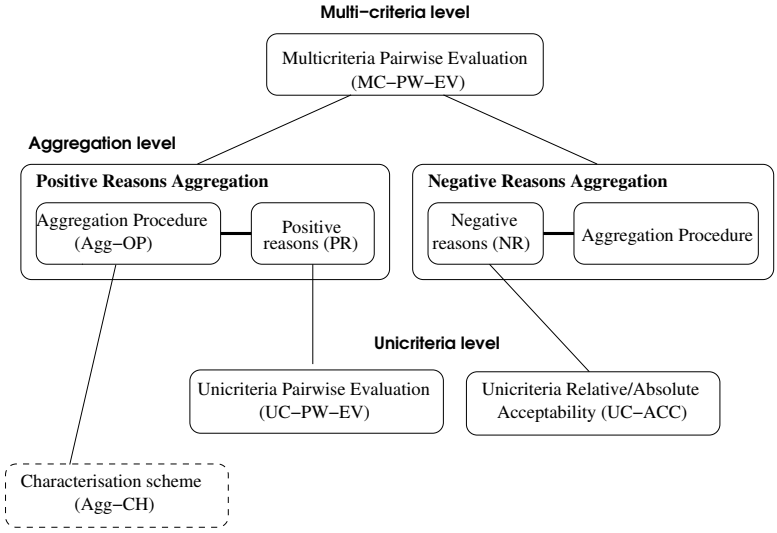



FIG. 1 – Une hiérarchie de schémas d'argument

### 3.1 Le niveau élémentaire

La brique de base dans un processus d'évaluation est la comparaison de deux actions sur la base d'un point de vue unique. Le premier schéma, illustré dans la Table1, présente l'évaluation relative, i.e. la comparaison d'une action à une autre action. Ce schéma traduit le fait qu'une action $a$ est au moins aussi bonne que l'action $b$ si la performance de $a$ sur le critère $i$ est aussi bonne que la performance de $b$ sur ce même critère suivant un modèle (ou une relation) de préférence donné.

TAB. 1 – Scheme for Unicriteria Pairwise evaluation (UC-PW-EV

| | | |
|---|---|---|
| **Premises** | a criterion | $h_i$ |
| | an action | $a$ |
| | whose performance is | $h_i(a)$ |
| | an action | $b$ |
| | whose performance is | $h_i(b)$ |
| | a preference relation | $\succeq_i$ |

**Conclusion** : $a$ is at least as good as $b$ ($a \succeq_i b$)

Au même niveau mais du coté négatif de la hiérarchie, nous définissons un argument pour l'acceptabilité d'une action. Par acceptabilité, nous faisons références au fait qu'une action $a$ ne peut être meilleure par exemple qu'une autre action $b$ s'il existe une raison négative contre elle. Cette raison peut se traduire, par exemple, par l'existence d'une différence de performances entre les deux actions qui favorise l'action $b$. Ce type d'acceptabilité est appelé «relatif». Un autre type d'acceptabilité est celui de la Table 2 représentant l'acceptabilité «intrinsèque». Ce schéma exprime le fait qu'il est possible de s'opposer à la conclusion «$a$ est au moins aussi bonne que $b$» sur la base que l'action $a$ n'a pas atteint un certain seuil, appelé seuil de veto, sur un critère donné (quelques soient ces performances sur les autres critères).

TAB. 2 – Scheme for Unicriteria Intrisic Acceptability (UC-IN-ACC)

| | | |
|---|---|---|
| **Premises** | a criterion | $h_i$ |
| | an action | $a$ |
| | whose performance is | $h_i(a)$ |
| | a veto threshold | $\mu$ |

**Conclusion** : $a$ is unacceptable according to $h_i$ ($g_i(a) < \mu$)

### 3.2 Le niveau agrégation

Ce niveau permet de synthétiser l'ensemble des résultats sur les différents critères. La particularité de ce niveau, comme nous l'avons expliqué au début de cette section, est qu'il est divisé en deux parties. Cependant, contrairement à ce qui a été proposé dans [13], où nous avons considéré pour chaque coté un schéma d'argument unique, dans ce qui suit nous proposons de définir un ensemble de schémas d'argument reliés entre eux comme le suggère la Figure 2.

Nous faisons remarquer que la discussion durant cette section va porter uniquement sur le coté positif de la hiérarchie. Un raisonnement simi-

laire peut être appliqué au coté négatif qui ne sera pas abordé dans cet article.

L'analyse de la phase d'agrégation du coté positif montre que la construction de l'ensemble des raisons positives est fortement liée à la procédure d'agrégation utilisée. En effet, une fois les différents arguments pour et contre la proposition $a \succeq b$, au niveau de chaque critère, sont construits, il convient de choisir une règle (ici une méthode d'agrégation) qui nous permettra de les départager et de les synthétiser. Différentes procédures produisent nécessairement différents ensembles de raisons positives.

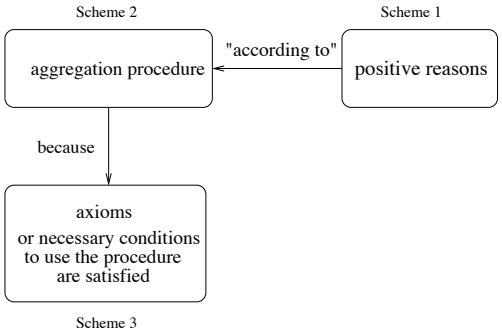



FIG. 2 – Le niveau agrégation

Afin de représenter ce lien entre les raisons positives et la procédure d'agrégation utilisée pour les construire, nous proposons d'enrichir l'ancienne structure hiérarchique en reliant deux schémas d'arguments distincts au niveau agrégation. La table 3 présente le détail du premier schéma, tel que $R_i$ est la méthode d'agrégation en cours d'utilisation.

TAB. 3 – Scheme for Positive reasons (RP)

**Premises** { a set of pairwise evaluation of actions $a$ and $b$

**Conclusion** : there are good reasons to support $a$ is at least as good as $b$ *according To* the selected "$R_i$"

Le deuxième schéma, quant à lui, formalise le mode de fonctionnement de la procédure d'agrégation. Il permet, entre autres, de fournir une explication sur la façon dont les raisons positives ont été construites. Un exemple est celui de la méthode lexicographique présenté dans la Table 4. Avec cette procédure, les raisons positives correspondent au fait qu'il existe un argument, en faveur de la conclusion finale, rattaché au critère le plus important selon l'ordre linéaire (voir l'exemple présenté dans la Section5).

TAB. 4 – Scheme for Argument from the lexicographic method

**Premises** $\begin{cases} \text{a set of criteria} & \{h_1, \ldots, h_n\} \\ \text{a linear order} & >_l \\ a \sim_i b & \forall j > k \\ a \succeq_k b \end{cases}$

**Conclusion** : Lexicographic is the selected procedure

D'un autre coté, il existe une variété de procédures d'agrégation des préférences, et faire un choix n'est pas souvent évident. Pour y remédier, l'analyste peut se baser sur les résultats de la théorie du Mesurage Conjoint. Comme nous l'avons mentionné précédemment, cette théorie propose d'associer à chaque procédure une caractérisation sous le forme d'un ensemble d'axiomes. Nous suggérons, alors, d'utiliser ces axiomes comme les prémisses d'un schéma d'argument nous permettons ainsi de soutenir l'utilisation d'une procédure en particulier. Le principal objectif à travers ce type de schéma est de fournir un moyen pratique d'évaluer, durant le dialogue, la possibilité d'utiliser ou non une procédure d'agrégation. Si par exemple au cours du processus d'évaluation, un axiome vient a être remis en cause, l'utilisation de la procédure à ce moment n'est plus justifiée.

En résumé, au niveau agrégation nous proposons de lier trois schémas d'argument qui permettent d'expliciter les étapes du raisonnement à ce niveau. En particulier, nous associons à chaque procédure d'agrégation deux schémas, le premier représente son mode de fonctionnement, offrant la possibilité de justifier la construction des raisons positives. Le second porte sur les conditions nécessaires aidant à justifier son utilisation au cours du processus.

### 3.3 Le niveau multicritère

Le dernier schéma que nous présentons est celui du niveau multicritère (voir la Table 5). Ce schéma repose sur les conclusions obtenues au niveau agrégation, tel que $\mathcal{R}_P$ représente la procédure qui doit être utilisée pour agréger les raisons positives en faveur de la conclusion ; tandis que $\mathcal{R}_N$ représente la procédure qui concerne l'agrégation des raisons négatives. Par ailleurs cette distinction fait à penser qu'il y aura (au moins) deux façons d'attaquer cet argument : soit sur la base d'un manque de soutien positif, ou sur la base de la présence d'une forte oppo-

sition(par exemple un veto).

TAB. 5 – Scheme for pairwise evaluation multicriteria (MC-PW-EV)

| | | |
|---|---|---|
| **Premises** | an action | $a$ |
| | an action | $b$ |
| | a set of criteria | |
| | there are enough supportive reasons | $\mathcal{R}_P$ |
| | there are no sufficiently strong reasons | $\mathcal{R}_N$ |

**Conclusion** : $a$ is at least as good as $b$ ($a \succeq b$)

# 4 Le représentation des arguments

Notre approche a pour ambition d'associer au dialogue durant le processus d'évaluation la construction d'un graphe sous la forme d'un arbre. Cet arbre permettra de garder trace et de visualiser l'ensemble des arguments et informations échangés durant ce dialogue. Notre objectif est de permettre au décideur d'avoir une vue assez générale mais précise des informations échangés au cours de l'interaction. De plus, cet outil servira à informer le décideur des effets, sur le modèle et/ou sur la conclusion finale, des différents changements et/ou mise à jours effectués au cours du processus.

Pour la construction de ce graphe nous avons été inspiré par le modèles du Carneades [8]. Ce modèle est un outil pour la visualisation et l'évaluation des arguments dans un graphe composé de différents nœuds et liens. De plus, il se base sur le concept de proof standards pour determiner l'acceptabilité des propositions. Avant d'adapter le modèle à notre contexte, nous décrivons dans ce qui suit les différents concepts liés au Carneades. Nous tenons à préciser que nous allons décrire uniquement les concepts abordés dans ce papier.

## 4.1 Le modèle du Carneades

Le graphe d'argument, dans le Carneades, est composé de deux types de nœuds, *les nœuds proposition* et *les nœuds arguments*. Les propositions sont des phrases déclaratives dans un certain langage. Un argument est une instantiation d'un schéma d'argument, liant un ensemble de prémisses à une conclusion. les prémisses peuvent être de trois types : (i) les *prémisses ordinaires* : elles sont établies par celui qui propose l'argument (ii) les *hypothèses* (assumptions) : elles sont supposées vraies par défaut jusqu'à ce qu'elles soient questionnées, et (iii) les *exception* : elles sont supposées fausses jusqu'à ce que le contraire soit établi. Cette distinction permet de distribuer la charge de la preuve durant le dialogue. En effet, a chaque schéma d'argument est associé un ensemble de questions critiques. Ces questions permettent d'identifier les différentes possibilités ou moyens d'invalider ou d'attaquer cet argument. Par conséquent, si une question critique est posée, alors la charge de la preuve est à celui qui a établit l'argument si la preuve qui doit être fournit concerne les prémisses ordinaires et les hypothèses. Par contre la charge est à celui qui a posé la question si la preuve concerne les prémisses de type exception. En outre, le graphe d'arguments inclut différents types de liens. Ces liens relient les premisses et les conclusion des arguments. Un exemple de graphe d'arguments est présenté dans la figure 3. Les nœuds arguments et propositions sont représentés, respectivement, par des cercles et des rectangles. Ce graphe illustre le celebre exemple de [16].

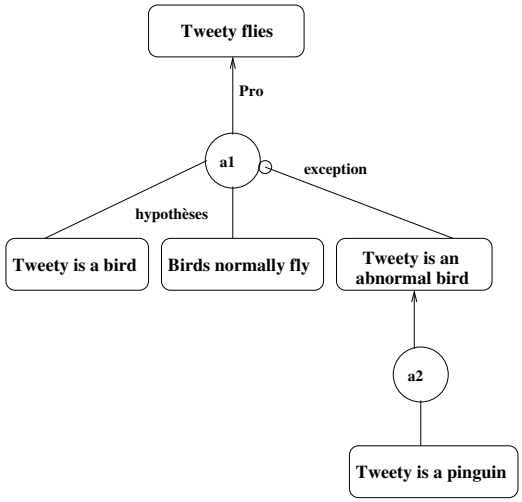



FIG. 3 – Un exemple de graphe d'arguments

Une fois les différents arguments construits, le Carneades offre un moyen de les évaluer. Contrairement au cadre abstrait d'argumentation (e.g.[6]), où l'évaluation consiste à determiner l'ensemble des arguments acceptables, la structure d'évaluation dans le Carneades est utilisée pour déterminer l'acceptabilité des propositions (statements) dans le graphe d'arguments. La détermination de cette acceptabilité dépends essentiellement de deux éléments : *la proof standard* rattachée à la proposition, et *le statut dialectique* de cette dernière dans le dialogue.

En effet, le Carneades a été conçu pour être

utilisé dans un contexte de dialogue. Ainsi, au cours du dialogue, le statut d'une proposition peut être : déclaré (stated), interrogé (questioned), accepté (accepted) ou rejeté (rejected). Le statut est fixé par les actes du langage du dialogue (par exemple, poser une question critiques, avancer un argument, etc). D'un autre coté, la *proof standard* permet de départager un ensemble d'arguments pour et contre une proposition suivant une règle précise. La proof standard est largement utilisée dans le domaine légale et elle est fixée par le contexte.Un exemple est celui de la *scintilla of evidence*. Avec cette proof, même le plus petit élément de preuve est suffisant pour soutenir la proposition. Selon la nature du problème, la proof standard peut être différente.

### 4.2 Graphe d'argument dans un contexte multicritère

Dans le cadre de notre processus, nous adoptons la représentation graphique proposée par le Carneades, en lui apportant quelques modifications. En premier, dans la version originale du Carneades, la syntaxe des propositions (statements) n'est pas spécifiée. Pour notre contexte, la définition de cette syntaxe est importante car elle représente différents types d'informations pour différents niveaux. Formellement,

**Definition 4.1.** *(*Proposition*)*
*Une proposition est un ensemble de phrases déclaratives construites sur la base des informations disponibles via le modèle d'évaluation (relation de préférence, performances pour les alternatives, etc.).*

Si $p$ est une proposition alors elle peut être une :

- **Information sur la procédure :** nous associons à chaque procédure d'agrégation un label qui nous permettra de reconnaître la procédure en cours d'utilisation. Par exemple, *Lexico*, pour la procédure lexicographique.
- **Information sur la polarité :** les deux cotés de la structures hiérarchiques sont distinguées par les deux labels : *positive* et *negative*.
- **Information préférentielle :** Les préférences du décideur sont représentés par des relations binaires qui reflètent des situations différentes face à deux actions, par exemple une préférence pour une actions (notée par *Pref*), une indifférence entre deux actions (notée *Indif*). Ces informations sont soit sur un critère : $Pref_{h_i}(x, y)$, $Indif_{h_i}(x, y)$, ou sur un ensemble de critères : $Pref(x, y)$, $Indif(x, y)$. Nous pouvons également avoir des informations préférentielles relatives aux critères : $Pref(h_i, h_j)$ ;
- **Information sur les performances :** à chaque action, nous pouvons lui associer une performance qui représente son évaluation sur un critère donné. Cette information est représentée par : $h_i(x)$ ;
- **Comparaison entre les performances :** Cette comparaison nous permet de construire une relation binaire entre deux actions. Un exemple : $h_i(x) > h_i(y)$ ;
- **Definitional information :** cette information est de type : $X$ équivalent à $Y$, tel que $X$ représente une information préférentielle et $Y$ est une comparaison entre les performances (e.g., $Pref_i(a, b) \equiv h_i(x) > h_i(y)$).

Par ailleurs, nous avons remarqué que le rôle jouer par la procédure d'agrégation ressemble fortement à celui du concept de la proof standard utilisé dans le Carneades. En effet, toutes deux ont pour principal objectif d'agréger un ensemble d'arguments selon une règle bien précise. Donc, une procédure d'agrégation peut très bien être assimilée à la proof standard dans le graphe d'argument. Cependant, contrairement au Carneades où la proof est fixée par le contexte et ne change pas durant le dialogue. Dans notre cas, la procédure d'agrégation peut être choisie et discutée à tout moment du processus de dialogue. De ce fait nous avons opté pour une représentation explicit de la proof sur le graphe. De plus comme la proof est rattaché à une proposition donnée. Dans notre cas, elle est rattaché au niveau agrégation par un nouveau lien, appelé «according To» que nous proposons d'ajouter aux différents liens existants dans le graphe.

**Definition 4.2.** *("according To")*
*Chaque proposition de type positif ou négatif est rattachée à une procédure d'agrégation par le lien "according to". Ce lien exprime le fait que les arguments construit en faveur ou contre cette proposition doivent être agréger selon cette procédure.*

Sur la figure 4 un exemple de graphe d'arguments, basé sur les différents concepts définis dans ce travail, est illustré. La construction de ce graphe est le résultat du dialogue présenté dans la section 5. D'une manière générale, le système construit la recommendation en emboîtant les différents schémas des trois niveaux. Ces schémas sont construits sur la base des informations fournies par le décideur. Si le client

contest ou à des doutes, le système fournit les différentes étapes du raisonnement en révélant avec parcimonie les niveaux inférieurs qui composent la recommandation. Chaque fois qu'un schéma est présenté, l'ensemble des questions critiques est mis à disposition de l'utilisateur pour lui permettre de discuter la conclusion en cours (des exemples de questions critiques sont présentées dans [13]). Il y'a différentes raisons qui poussent à faire des modifications ou des révisions dans un tel processus. Dans certains cas, l'utilisateur veut simplement corriger ou améliorer une de ces précédentes déclarations, introduire de nouvelles informations. Dans d'autres cas, il peut souhaiter contredire l'une des hypothèses avancées par le système.

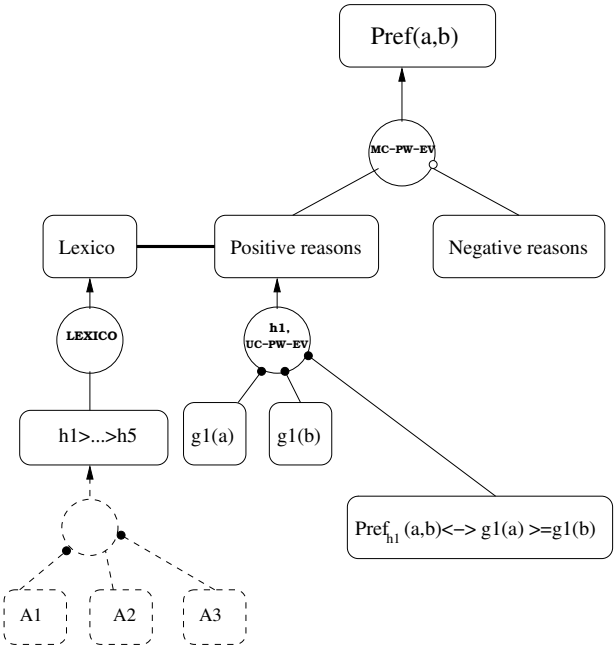



FIG. 4 – Exemple de la nouvelle structure du graphe d'argument

## 5 Exemple

Dans cette section, nous donnons un aperçu du système dialectique, dont les informations échangées sont représentées sur la figure 4.

User : [renseigne son modèle d'évaluation : deux actions $a$ et $b$, six critères $h_0, h_1, h_2, h_3, h_4, h_5$, tel que $h_0$ est considéré le plus important, suivi de $h_1$, et ainsi de suite. Les différentes évaluations des actions sont regroupées dans le tableau suivant :]

| | $h_0$ | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $h_5$ |
|---|---|---|---|---|---|---|
| $a$ | 3 | 7.5 | 6 | 2 | 3 | 5 |
| $b$ | 3 | 7 | 4 | 8 | 4 | 7 |

1. System : Given your informations, $a$ is at least as good as $b$. [Recommendation]
2. User : Why ? [Challenge]
3. System : The second most important criteria according to you defend this claim, so by comparing actions on the basis of criteria of decreasing importance, $a$ should be prefered to $b$ [Justified Recommendation]
4. User : however, I see that the difference between $a$ and $b$ is not significant enough...[update information]
5. System : My recommendation remains the same because $a$ is prefered to $b$ on the third most important criterion [Recommendation]
6. User : I agree but still $a$ cannot be acceptable. [reject the recommendation]
7. System : Why ? [Burden of proof shift]
8. User : $b$ is way better than $a$ on $h_3$, I could not justify this choice... [Argument Scheme for Relative acceptability (negative reason)]
9. System : Ok. [Revise the recommendation] But beware that applying this scheme for acceptability may have the following consequences... [Inform of consequences of this revision]

Nous décrivons brièvement, dans ce qui suit, le dialogue. Au tour 1, la recommendation est présentée, qui est aussitôt contestée par l'utilisateur au tour 2. L'idée est que l'utilisateur demande au système de fournir des informations plus explicites en ce qui concerne les raisons positives qui soutiennent cette recommendation. Le système au tour 3, donne une explication sur la base du schéma RP de la table 3. Au tour 4, l'utilisateur attaque l'argument élémentaire UC-PW-EV (construit à partir de la table1), sur lequel la recommandation est basée. La question critique porte sur la pertinence de la relation de préférence utilisée. Le système accepte et modifie la relation de préférence sur la base de la nouvelle information de l'utilisateur. Mais, en même temps, le système réaffirme que la recommandation reste inchangée : cela est dû au fait que sur le troisième plus important critère, $a$ est toujours meilleure que $b$. L'utilisateur accepte mais en même temps attaque sur la base de l'existence d'une raison négative et justifie que $a$ ne peut être acceptée en faisant appel au schéma relatif à l'acceptabilité. Finalement, le système révise la recommandation et peut par la suite expliciter les conséquences de cette révision.

## 6 Travaux antérieurs

Différentes approches ont proposé d'utiliser l'argumentation dans un context de prise de décision. Une étude critique a été présentée récemment dans [12]. D'un point de vue aide à la décision certains éléments restes largement inexplorées. En effet, les modèles actuels d'argumentation ne sont pas totalement satisfaisant pour notre contexte pour différentes raisons, par exemple : (i) la plupart des approches [9, 11] supposent un seul type de problème de décision dont le but est de sélectionner la meilleure solution, tandis que nous pouvons distinguer une variété de problèmes tel que : le choix, le rangement, le tri, etc. ; (ii) la plupart des modèles [7, 3] se basent sur l'évaluation intrinsèque des actions (argument en faveur ou contre une action), tandis que beaucoup de problèmes de décision font appel à l'évaluation relative (les actions sont comparées les unes aux autres) et (iii) beaucoup d'approches [1, 4] se basent sur des procédures d'agrégation portant sur le nombre ou la force des arguments, or l'analyse multicritère propose des procédures plus variées et plus complexe, prenant en compte les préférences du décideur.

Par ailleurs, récemment, il y a eu un grand intérêt pour les outils de représentation des arguments. Ces outils permettent à leurs utilisateurs de construire et de visualiser des arguments sous la forme de graphes composés de nœuds et de liens. Ils sont conçus pour rendre cette tâche la plus simple possible en guidant les utilisateurs lors de la constructions. Dans ce papier nous avons présenté un exemple de ces outils, à savoir le Carneades. D'autres outils existent (e.g. Araucaria [15]) mais qui semblent atteindre leurs limites quand il s'agit de représenter des schémas d'argument et leurs questions critiques (pour plus d'informations sur ce sujet voir [24]).

## 7 Conclusion et travaux futurs

Nous avons présenté dans ce papier une nouvelle structure de schémas d'argument visant à représenter les différentes étapes d'un processus d'évaluation multicritère. D'un coté, nous avons distingué deux nouveaux schémas rattachés à une procédure d'agrégation. Le premier basé sur les conditions de fonctionnement de la procédure et le second sur les conditions nécessaires pour son utilisation. D'un autre coté, nous avons adapté une structure de graphe d'arguments pour qu'elle puisse s'intégrer au contexte d'aide à la décision. L'objectif principal de cette structure est de permettre la visualisation et par la suite l'évaluation des différents arguments échangés au cours du dialogue décideur-analyste.

Plusieurs pistes sont envisagées pour enrichir ce travail. Le premier point concern l'évaluation des schémas d'argument basé sur les axiomes de la caractérisation d'une procédure. Par évaluation, nous faisons référence à un moyen qui nous permettra de vérifier la validité d'un axiome et de définir sur cette base si l'utilisation d'une procédure est toujours justifiée ou pas. Le second point concerne la détermination de l'acceptabilité des différentes proposition du graphe d'argument. En effet, le Carneades propose une fonction d'acceptabilité, qui prend en compte plusieurs éléments, tel que la proof standard, le statut dialectique et le type des prémisses. Intuitivement, une proposition est acceptable si la décision d'accepter cette proposition peut être expliquée ou justifiée en utilisant les arguments qui ont été mis en avant dans le dialogue. Etant donné les modifications que nous avons effectué sur le graphe, dans quelle mesure cette fonction peut être adaptée à la nouvelle structure ?

## Références

[1] L. Amgoud, J.-F. Bonnefon, and H. Prade. An argumentation-based approach to multiple criteria decision. In *Proc. of the 8th European Conf. on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 269–280, 2005.

[2] L. Amgoud, C. Cayrol, and M.-C. Lagasquie Schiex. On the bipolarity in argumentation frameworks. In J.P. Delgrande and T. Schaub, editors, *Proc. of 10th Inter. Workshop on Non-Monotonic Reasoning*, pages 1–9, Canada, 2004.

[3] K. Atkinson. Value-based argumentation for democratic support. In P.E Dunne and T.J.M. Bench-Capon, editors, *Proc. of the 1st Inter. Conf. on Computational Models of Natural Argument*, volume 144, pages 47–58, Amesterdam, The Netherlands, 2006.

[4] J.-F. Bonnefon and H. Fargier. Comparing sets of positive and negative arguments : Empirical assessment of seven qualitative rules. In G. Brewka, editor, *Proc. of 17th European Conf. on AI*, Riva di Gard, Italy, 2006.

[5] D. Bouyssou, Th. Marchant, M. Pirlot, A. Tsoukiàs, and Ph. Vincke. *Evaluation and decision models with multiple criteria : Stepping stones for the analyst*. Springer, Boston, 2006.

[6] P. M. Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-person games. *Artificial Intelligence*, 77(2) :321–358, 1995.

[7] J. Fox and S. Parsons. On Using Arguments for Reasoning about Actions and Values. In *Proc. of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, pages 55–63, 1997.

[8] T. Gordon, H. Prakken, and D. Walton. The carneades model of argument and burden of proof. *Artificial Intelligence*, 171(4) :875–896, 2007.

[9] N. I. Karacapilidis and D. Papadias. A group decision and negotiation support system for argumentation based reasoning. In *Selected Papers from the Workshop on Reasoning with Incomplete and Changing Information and on Inducing Complex Representations*, pages 188–205, London, UK, 1998.

[10] D.H. Krantz, R.D. Luce, P. Suppes, and A. Tversky. *Foundations of measurement*, volume 1 : Additive and polynomial representations. Academic Press, New York, 1971.

[11] M. Morge and P. Mancarella. The hedgehog and the fox. an argumentation-based decision support system. In *Proc. of the 4th Intern. Workshop on Argumentation in Multi-Agent Systems*, 2007.

[12] W. Ouerdane, N. Maudet, and A. Tsoukiàs. Arguing over actions that involve multiple criteria : A critical review. In *Proc. of the 9th European Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 308–319, 2007.

[13] W. Ouerdane, N. Maudet, and A. Tsoukiàs. Argument schemes and critical questions for decision aiding process. In Ph. Besnard, S. Doutre, and A. Hunter, editors, *Proc. of the 2nd Inter. Conf. on Computational Models of Argument*, pages 285–296, 2008.

[14] M. Oztürk, A. Tsoukiàs, and Ph. Vincke. Preference modelling. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis : State of the Art Surveys*, pages 27–72. Springer Verlag, Boston, Dordrecht, London, 2005.

[15] C. Reed and G. Rowe. Araucaria : Software for argument analysis, diagramming and representation. *International Journal of AI Tools*, 14 :2004, 2004.

[16] R. Reiter. Nonmonotonic reasoning. *Annual Review of Coputer Science*, 2 :147–186, 1987.

[17] B. Roy. The outranking approach and the foundations of ELECTRE methods. *Theory and Decision*, 31 :49–73, 1991.

[18] B. Roy and D. Bouyssou. *Aide Multicritèr à la Décision : Méthode et Cas*. Economica, Paris, 1993.

[19] A. Tsoukiàs. On the concept of decision aiding process. *Annals of Operations Research*, 154(1) :3 – 27, 2007.

[20] A. Tsoukiàs, P. Perny, and Ph. Vincke. From concordance/discordance to the modelling of positive and negative reasons in decision aiding. In D. Bouyssou, E. Jacquet-Lagrèze, P. Perny, R. Slowinski, D. Vanderpooten, and Ph. Vincke, editors, *Aiding Decisions with Multiple Criteria : Essays in Honour of Bernard Roy*, pages 147–174. Kluwer Academic, Dordrecht, 2002.

[21] B. Verheij. On the existence and multiplicity of extensions in dialectical argumentation. In S. Benferhat and E. Giunchiglia, editors, *Porc. of the 9th Inter. Workshop on Non-Monotonic Reasoning*, pages 416–425, Toulouse, France, 2002.

[22] D.N. Walton. *Argumentation schemes for Presumptive Reasoning*. Mahwah, N. J.,Erlbaum, 1996.

[23] D.N. Walton. Justification of argument schemes. *Australasian journal of logic*, 3 :1–13, 2005.

[24] D.N. Walton. Visualization tools, argumentation schemes and expert opitnion evidence in law. *Law, Probability and Risk Advance Access*, pages 1–22, 2007.

# Formal ex-post rationalization – A complete conclusion-based procedure for judgment aggregation

Gabriella Pigozzi
gabriella.pigozzi@uni.lu

Marija Slavkovik
marija.slavkovik@uni.lu

Leendert van der Torre
leendert@vandertorre.com

Individual and Collective Reasoning
Computer Science and Communication
University of Luxembourg
6, Rue Richard Coudenhove Kalergi, L-1359 – LUXEMBOURG

**Abstract:**
Judgment aggregation is a formal theory reasoning about how a group of agents can aggregate their individual judgments on connected propositions into a collective set of judgments on these propositions. There are three procedures for aggregating judgments sets : premise-based procedure, conclusion-based procedure and distance-based merging. The conclusion-based procedure has been little investigated because it provides a way to aggregate the conclusions, but not the premises, thus it outputs an incomplete judgment set. The goal of this paper is to present a conclusion-based procedure outputting complete judgment sets. Our procedure is composed of two steps. First, majority voting is used to aggregate the conclusion as it has been previously done by the incomplete conclusion-based procedure. Once a collective conclusion is reached, distance-based merging is used to aggregate the premises using the chosen conclusion as an integrity constraint to select only those premise sets which support it. An additional integrity constraint can be added in the merging phase to ensure that unanimity is heeded when selecting the premises. Additionally we discuss the issue of manipulability of the proposed procedure by drawing parallels with existing work on manipulability of belief merging operators.

**Keywords:** judgment aggregation, conclusion-based procedure, distance-based merging

## 1 Introduction

Judgment aggregation [13] is an emerging research area in economics. It is a formal theory reasoning about the aggregation of judgments of agents in small groups such as expert panels, legal courts, boards and councils. Judgment aggregation has recently attracted attention in multiagent systems and computer science, which often deal with the combination of information coming from equally reliable sources.

We talk about judgment aggregation whenever a group of individuals needs to make a collective decision on a finite set of issues, and these propositions are logically connected. The propositions are of two kinds : *premises* and a *conclusion*. The first serve as supporting reasons to derive a certain judgment on the conclusion. If, for example [2], your department has to hire a new lecturer and the decision rule is such that a candidate $X$ will be hired only if the candidate is good at teaching and good at research, we will say that "hiring $X$" is the conclusion while "good at teaching" and "good at research" are the premises.

How shall we derive a group decision given the individuals' opinions on premises and conclusion ? It is assumed that each individual expresses yes/no opinions on the propositions while respecting the logical relations. If we now define the group opinion as the majority view on the issues, it turns out that the collectivity may have to endorse an inconsistent position. This means that your department may have to face a situation in which the majority does not deem $X$ a good candidate. However, it will not be possible to provide reasons for this as a majority of people agrees that $X$ is actually good at teaching and (another) majority deems $X$ to be good at research. An example of such situation is presented in Table 1.

| | prof. A | prof. B | prof. C | Majority |
|---|---|---|---|---|
| $X$ is good at teaching | *yes* | *yes* | *no* | *yes* |
| $X$ is good at research | *no* | *yes* | *yes* | *yes* |
| hire $X$ | *no* | *yes* | *no* | *no* |

TAB. 1 – Hiring committee example. The candidate $X$ is hired only if $X$ is good at teaching and $X$ is good at research.

The problem is avoided if we decide to let the majority vote on the premises to dictate the final decision on the hiring process or if we decide just on the conclusion and ignore the reasons given by the group members. Unlike the aggregation procedure on the premises only [6], the aggregation on the individual judgments on the conclusion has been too quickly dismissed.

We claim that in many decision problems, the final conclusion is arguably more relevant than the reasons for it. For example, when deciding which candidate to hire in your department, you

may be more concerned of which new colleague you will have in your department than of the reasons for choosing her. Considering only the individual judgments on the conclusions has also the advantage that it is a strategy-proof procedure : if you disagree with the conclusion, you have no incentive to be insincere. The best you can do to increase the possibility that the group will agree with you is to truthfully state your view about the conclusion. The same does not hold when you aggregate on the premises.

The problem this paper addresses is how a group can make decisions on the conclusion while providing reasons in support of the collective conclusion. Our procedure prioritizes the individual judgments on the conclusion and outputs sets of premises that support the collective decision.

The paper is structured as follows : in Section 2 we present the problem of judgment aggregation. Section 3 is devoted to our formal framework, and in Section 4 we prove some results about our procedure. Section 5 relates our approach with existing work and, finally, Section 6 concludes the paper and outlines directions for future work.

## 2 Judgment aggregation

In this section we first present the original aggregation paradox from which judgment aggregation originated. Agents are required to express judgments (in the form of yes/no or, equivalently, 1/0) over propositions that have different status. More specifically, some propositions (called *premises*) provide the reasons for some other propositions (the *conclusions*). As in [21], to represent the distinction between premise and conclusion in our language, and in contrast to the existing literature on judgment aggregation, we distinguish between premise variables $a, b, c, p, q \ldots$, and conclusion variable $x$.

The problem of judgment aggregation was discussed by Kornhauser and Sager [10, 11]. In their example, a court has to make a decision on whether a person is liable of breaching a contract (proposition $x$, or *conclusion*). The judges have to reach a verdict following the legal doctrine. This states that a person is liable if and only if there was a contract (first *premise* $a$) and there was a conduct constituting breach of such a contract (second *premise* $b$). The legal doctrine can be formally expressed by the rule $(a \wedge b) \leftrightarrow x$. Each member of the court expresses her judgment on the propositions $a$, $b$ and $x$ such that the rule $(a \wedge b) \leftrightarrow x$ is satisfied.

Suppose now that the three members of the court make their judgments according to Table 2.

| Agenda | A | B | C | Majority |
|---|---|---|---|---|
| $a$ | 1 | 0 | 1 | 1 |
| $b$ | 0 | 1 | 1 | 1 |
| $x = (a \wedge b)$ | 0 | 0 | 1 | 0 |

TAB. 2 – Doctrinal paradox. Premises : $a$ = There was a contract, $b$ = There was conduct constituting breach of such a contract. Conclusion : $x = (a \wedge b)$ = There was a breach of contract.

Each judge expresses a consistent opinion, i.e. she says yes to $x$ if and only if she says yes to both $a$ and $b$. However, propositionwise majority voting (consisting in the separate aggregation of the votes for each proposition $a$, $b$ and $x$ via majority rule) results in a majority for $a$ and $b$ and yet a majority for $\neg x$. This is an inconsistent collective result, in the sense that $\{a, b, \neg x, (a \wedge b) \leftrightarrow x\}$ is inconsistent in propositional logic. The paradox lies in the fact that majority voting can lead a group of rational agents to endorse an irrational collective judgment. The literature on judgment aggregation refers to such problems as the *doctrinal paradox*.

The relevance of such aggregation problems goes beyond the specific court example, because it applies to all situations in which individual binary evaluations need to be combined into a group decision. Furthermore, the problem of aggregating individual judgments is not restricted to majority voting, but it applies to all aggregation procedures satisfying some seemingly desirable conditions. For an overview, the reader is referred to [15].

The paradox originates from the fact that it is assumed that the aggregation on the premises should be logically equivalent to the aggregation on the conclusion, i.e. the group of agents should say yes to $x$ if and only if the group says yes both to $a$ and $b$. However, by applying the majority rule on each proposition separately, the logical relations between premises and conclusion are disregarded. The aggregation of logically related propositions into a consistent outcome cannot be achieved by imposing that all propositions should be treated independently of each other. The independence condition that is

imposed on the aggregation rules in the literature is a reminiscence of the independence of irrelevant alternatives in social choice theory. In the aggregation of judgments, where the propositions are connected, the independence condition is the source of the inconsistent group outcomes.

### 2.1 Premise vs conclusion-based procedure

Two ways to avoid the inconsistency are the *premise-based procedure* and the *conclusion-based procedure* [19, 4]. According to the premise-based procedure, each agent votes on each premise. The conclusion is then inferred from the rule $(a \wedge b) \leftrightarrow x$ and from the judgment of the majority of the group on $a$ and $b$. If the judges of the example followed the premise-based procedure, the defendant would be declared liable of breaching the contract.

In the premise-based procedure, the aggregation of a premise proposition is independent from the other premise propositions and the aggregation of the conclusion depends only on the aggregation results of all the premises. For example, if judge A and judge B both would argue that there was no contract and there was no conduct constituting breach of such a contract, then the conclusions of the individual judges would be like in Table 2. Yet, unlike Table 2, there would be no inconsistent collective judgment on $a$, $b$ and $x$, and the defendant would be declared not liable.

Because the aggregation of the conclusion does not depend on the individual votes on the conclusion, it is possible for the premise-based procedure to violate the majority, or even the unanimity, of the individual votes on the conclusion.

In [18] Nehring presents a variation on the discursive dilemma, which he calls the Paretian dilemma for short. In his example, a three-judges court has

> to decide whether a defendant has to pay damages to the plaintiff. Legal doctrine requires that damages are due if and only if the following three premises are established : 1) the defendant had a duty to take care, 2) the defendant behaved negligently, 3) his negligence caused damage to the plaintiff. [p.1]

Suppose that the judges vote as in Table 3.

| Agenda | A | B | C | Majority |
|---|---|---|---|---|
| $a$ | 1 | 0 | 1 | 1 |
| $b$ | 1 | 1 | 0 | 1 |
| $c$ | 0 | 1 | 1 | 1 |
| $x = (a \wedge b \wedge c)$ | 0 | 0 | 0 | 0 |

TAB. 3 – Paretian dilemma. Premises : $a$ = duty, $b$ = negligence., $c$ = causation. Conclusion : $x = (a \wedge b \wedge c)$ = damages.

The Paretian dilemma is disturbing because, if the judges would follow premise-based procedure, they would condemn the defendant to pay damages contradicting the *unanimous* belief of the court that the defendant is *not* liable.

A conclusion-based procedure would not lead to such a unanimity violation. According to the conclusion-based procedure, the judges decide privately on $a$ and $b$ and only express their opinions on $x$ publicly. The judgement of the group is then inferred from applying the majority rule to the agent judgments on $x$. The defendant will be declared liable if and only if a majority of the judges actually believes that she is liable. In the example, contrary to the premise-based procedure, the application of the conclusion-based procedure would free the defendant. However, no reasons for the court decision could be supplied.

Unlike the premise-based procedure [16, 6], conclusion-based procedure did not receive much attention in the literature. In this paper we aim at filling this gap. We propose a procedure that attempts to overcome the major limit of conclusion-based procedure, that is the lack of reasons supporting the decision.

## 3 Framework

In this section we introduce our formal framework to represent judgment aggregation problems. A set of agents $N = \{1, 2, \ldots, n\}$, with $n \geq 3$ and odd, makes judgments on logically interconnected propositions. The set $\mathcal{P}$ of atomic propositions is defined as the union of two disjoint sets : $\mathcal{P}_p$ containing variables $a, b, c, \ldots, p, q$ for the premises, and $\mathcal{P}_c$ being a singleton $\{x\}$, where $x$ is the variable for the conclusion. We assume that the conclusion is an atomic formula. $\mathcal{L}$ is a language built from $\mathcal{P}$, including complex formulas as $\neg a, (a \wedge b), (a \vee b), (p \rightarrow q), (a \leftrightarrow p)$.

The set of issues on which the judgments have

to be made is called *agenda* and is denoted by $\Phi \subseteq \mathcal{L}$. The agenda is closed under negation : if $a \in \Phi$, then $\neg a \in \Phi$.[1] We split the agenda in two parts : one containing the premises ($\Phi_p$), and one containing the conclusion ($\Phi_c$). We exclude agenda items such as $a \rightarrow x$, i.e. formulas containing premises and conclusion. Our procedure consists of two different aggregations : one on the individual judgments on $\Phi_p$ and one on the individual judgments on $\Phi_c$.

A subset $J \subseteq \Phi$ is the *collective judgment set* and contains the set of propositions believed by the group. Similarly, we define individual $i$'s judgment set $J_i \subseteq \Phi$. A collective judgment set is *consistent* if it is a consistent set in $\mathcal{L}$, and is *complete* if, for any $a \in \mathcal{L}$, $a \in J$ or $\neg a \in J$ (consistent and complete individual judgment sets are defined in the same way). We only consider consistent complete judgment sets.

A *decision rule* $\mathcal{R}$ is a formula of $\mathcal{L}$ that represents the logical connections between premises and conclusion. More precisely, $\mathcal{R}$ has the form $\Psi \leftrightarrow x$, where $\Psi \in \mathcal{L}/\{x\}$. The decision rule is not an item of the agenda. This means that the group members do not vote on $\mathcal{R}$, but each individual is required to give judgments that satisfy the given rule (see [1] for an approach in which agents can disagree with the rule).

Like the agenda, each judgment set is split in two disjoint subsets : $J_{i,p}$ and $J_{i,c}$. The first is the individual $i$'s judgment set on the premises, and $J_{i,c}$ is the individual $i$'s judgment set on the conclusion. The collective judgment sets on premises and conclusion will be denoted respectively by $J_p$ and $J_c$.

We say that a premise $a$ (resp. a conclusion $x$) is *unanimously supported* if $a \in J_{i,p}$ for all $J_{i,p} \subseteq \Phi$ (resp. $x \in J_{i,c}$ for all $J_{i,c} \subseteq \Phi$).

A *profile* $\underline{J}$ is an $n$-tuple $(J_1, J_2, \ldots, J_n)$ of agents' judgment sets. An *aggregation rule* $F$ assigns a set of collective judgment sets $J$ to each profile $\underline{J}$. To define our procedure, we define two aggregation rules : one for the aggregation of the individual premises and one for the conclusion. To relate the two aggregation rules, we have set of integrity constraints $IC$ to govern the premise-aggregation rule. Also, to allow for situations in which the aggregated judgment set is not unique, i.e. there are ties, we aggregate the profiles into *sets* of aggregated judgment sets.

A *premise profile* $\underline{J_p}$ is an $n$-tuple $(J_{1,p}, J_{2,p}, \ldots, J_{n,p})$ of agent judgment sets on premises. A *premise aggregation rule* $F_{IC}$ assigns a set of collective judgment sets $J_p$ to each premise profile $(J_{1,p}, J_{2,p}, \ldots, J_{n,p})$ and set of integrity constraints $IC$. Conclusion profiles $(J_{1,c}, J_{2,c}, \ldots, J_{n,c})$ and conclusion aggregation rules $F_c$ are defined similarly.

### 3.1 Complete conclusion-based procedure

Each individual provides, simultaneously, the set of premises and conclusion that she believes. Our two-step procedure first performs a standard conclusion-based procedure, i.e. it aggregates the individual judgments on the conclusion by majority rule. This means that $x$ (resp. $\neg x$) is the collective conclusion iff there are at least $\frac{n+1}{2}$ agents voting for $x$ (resp. $\neg x$). The second step consists in determining the set of reasons which support the collective conclusion. This is done by applying a *distance-based merging operator* to $J_{i,p}$.

Distance minimization merging procedures have been already applied to judgment aggregation problems [20]. In this section we briefly present a majority merging operator with integrity constraints following [9, 8]. Whereas merging operators in general merge sets of propositional sentences called knowledge bases, we are interested in the particular case of merging maxi-consistent sets of literals called complete knowledge bases. When there are ties, a belief merging operator returns the disjunction of the alternatives.

An *interpretation* is a function $v : \mathcal{P} \rightarrow \{0, 1\}$ and it is represented as the list of the binary evaluations. For example, given three propositional variables $a$, $b$ and $c$, the vector (0,1,0) stands for the interpretation in which $a$ and $c$ are false and $b$ is true. Let $\mathcal{W} = \{0, 1\}^{\mathcal{P}}$ be the set of all interpretations. An interpretation is a *model* of a propositional formula if and only if it makes the formula true in the usual truth functional way.

Let us suppose that $\Phi_p = \{a, \neg a, b, \neg b, c, \neg c\}$, and that agent 1 believes that $a, \neg b$ and $\neg c$, i.e. $J_{1,p} = \{a, \neg b, \neg c\}$. We represent $J_{1,p}$ as a 0-1 vector of length equal to the number of propositions in $J_{1,p}$, i.e. $(1, 0, 0)$. Suppose also that $\mathcal{R} = ((a \vee b) \wedge c) \leftrightarrow x$ and that, unlike agent

[1] To increase readability, in the tables we list only the positive issues, and assume that, for any issue in the agenda, an individual deems that issue to be true if and only if she deems its negation to be false.

1, the majority of the individuals voted in favor of $x$. Hence, the first step of our procedure sets $v(x) = 1$. We now want to define an aggregation on $J_{i,p}$ such that the collective judgment set on the premises is one of the models of $((a \vee b) \wedge c) \leftrightarrow x$ where $v(x) = 1$. This means that $J_p$ must be one of the following interpretations : (1,1,1), (0,1,1), (1,0,1). The set of premises supporting the collective conclusion will constrain the aggregation procedure on $J_{i,p}$. We indicate the set of admissible interpretations as $IC$ (*integrity constraints*).

Given a premise profile $\underline{J_p}$ and $IC$, $F_{IC}(\underline{J_p})$ denotes a set of collective judgment sets on the premises resulting from the $IC$ merging on $\underline{J_p}$. The idea of a distance minimization merging operator is that $F_{IC}(\underline{J_p})$ will select those interpretations in $IC$, which are at minimal distance from $\underline{J_p}$. A distance $d(\omega, \underline{J_p})$ between an interpretation $\omega$ and the premise profile $\underline{J_p}$ induces a total pre-order ($\leq$) on the interpretations.

In order to obtain the total pre-order on the interpretations, we first need to determine a pseudo-distance between each admissible interpretation and each $J_{i,p}$. Then, we need to aggregate all these values in order to obtain a pseudo-distance value between an interpretation and $\underline{J_p}$. Let us see this in detail (we follow [9, 8]).

A pseudo-distance between interpretations is a function $d : \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}_+$ such that for all $\omega, \omega' \in \mathcal{W}$ :

1. $d(\omega, \omega') = d(\omega', \omega)$
2. $d(\omega, \omega') = 0$ iff $\omega = \omega'$.

A pseudo-distance between an interpretation $\omega$ and $\underline{J_p}$ is defined with the help of an aggregation function $D : \mathbb{R}_+^n \rightarrow \mathbb{R}_+$ as $D^d(\omega, \underline{J_p}) = D\left(d(\omega, J_{1,p}), d(\omega, J_{2,p}), \ldots, d(\omega, J_{n,p})\right)$ [8]. Any such aggregation function induces a total pre-order $\leqslant_{\underline{J_p}}$ on the set $\mathcal{W}$ with respect to the pseudo-distances to a given $\underline{J_p}$. Thus, an $IC$ *majority merging operator* for a profile $\underline{J_p}$ can be defined as $\Delta_{IC}(\underline{J_p}) = \min([IC], \leqslant_{\underline{J_p}})$, i.e., the set of all models of $IC$ (denoted by $[IC]$) with minimal pseudo-distance $D^d$ to $\underline{J_p}$. The minimal pseudo-distance identifies the final collective outcome on the premises, i.e. the set of premises that support the conclusion voted by the majority of the agents and with the minimal distance among all possible models satisfying $IC$.

A majority merging operator, often mentioned in the literature, is the operator $\Delta_{IC}^{d,\Sigma}$ defined as follows :

1. $d$ is the Hamming distance — the number of propositional letters on which two interpretations differ, i.e., $d(\omega, \omega') = |\{\pi \in \mathcal{P} | \omega(\pi) \neq \omega'(\pi)\}|$ and
2. $D^d(\omega, \underline{J_p}) = \sum_i d(\omega, J_{i,p})$ is the sum of componentwise distances $d$ defined before.

For example, the Hamming distance between $\omega = (1, 0, 0)$ and $\omega' = (0, 1, 0)$ is $d(\omega, \omega') = 2$. In the following we use the Hamming distance because it is a well known and intuitive distance. But the Hamming distance is only one among many possible distance functions that we may use.

The premise aggregation rule $F_{IC}$ outputs the disjunction of the sets selected by $\Delta_{IC}^{d,\Sigma}$ as the reasons in support of the conclusion voted by the majority of the agents. Given a premise profile $\underline{J_p}$, $F_{IC}$ is defined as :

$$F_{IC}(\underline{J_p}) = \bigvee \Delta_{IC}^{d,\Sigma}(\underline{J_p})$$

The constraint $IC$ is defined as $IC = \mathcal{R} \wedge \hat{x}$, where $\hat{x}$ is the conclusion chosen by the majority.

The best way to illustrate our procedure is with an example. For simplicity, we take the original doctrinal paradox.

**Example 1** *Individuals vote as in Table 2. Their judgment sets are then :* $J_1 = \{(1, 0, 0)\}$, $J_2 = \{(0, 1, 0)\}$ *and* $J_3 = \{(1, 1, 1)\}$. *A majority is in favor of* $\neg x$, *so the first step of our procedure sets that the group decided* $v(x) = 0$.

*We now want to provide a set of reasons in support of* $v(x) = 0$. *In order to select them, we calculate the minimal distance between* $IC = \{(1, 0), (0, 1), (0, 0)\}$ *(all the admissible sets of reasons for* $v(x) = 0$*) and each* $J_{i,p}$, $1 \leq i \leq 3$. *This is calculated in the table below.*

*The final group decision will support* $v(x) = 0$ *and the reasons will be* $(0, 1) \vee (1, 0)$, *representing a tie between* $(0, 1)$ *and* $(1, 0)$.

| $\omega$ | $J_{1,p}$ | $J_{2,p}$ | $J_{3,p}$ | $\Sigma_i\ d(\omega, J_{i,p})$ |
|---|---|---|---|---|
| (0,1) | 2 | 0 | 1 | 3 |
| (1,0) | 0 | 2 | 1 | 3 |
| (0,0) | 1 | 1 | 2 | 4 |

TAB. 4 – Selection of premises for the doctrinal paradox with $v(x) = 0$.

The example illustrates that, when aggregating the premises, we do not only take into account the judgment sets of the agents that support the aggregated conclusion, but also the judgment sets of agents that do not support the conclusion. Consider for example the selection of premises for the doctrinal paradox with $v(x) = 0$ in Table 2. We take also judgment set $J_{3,p}$ into account, although the third judge voted for $v(x) = 1$.

The justification for taking all individual judgments on the premises into account is twofolded. On the one hand, from the perspective of probability theory, if all judgments are independent, then more judgment sets mean a higher chance to get a better judgment. On the other hand, from the perspective of democracy, involving agents whose conclusion is not supported will give broader basis for the decision. However, we do not exclude the possibility that there are situations in which only the individuals' judgments that actually supported the aggregated conclusion should be taken into account when determining the reasons for that conclusion.

We now want to show that our procedure does not always provide a set of reasons, but that can also select a unique justification for the collective conclusion.

**Example 2** *Consider a* collegium medicum *that wishes to eliminate the possibility of a patient suffering from condition X before administering a treatment. We take $v(x) = 0$ if the patient is free of X. The doctors consider the three relevant alternative medical conditions $a$, $b$ and $c$ the patient may suffer from. The patient is free of X if medical conditions $a$, $b$ and $c$ are present ($v(a) = 1$, $v(b) = 1$ and $v(c) = 1$), if all three medical conditions are absent ($v(a) = 0$, $v(b) = 0$ and $v(c) = 0$) or if the last condition is present while the previous two are absent ($v(a) = 0$, $v(b) = 0$ and $v(c) = 1$). In all other cases the patient is likely to suffer from X. Table 5 gives the truth table of $\mathcal{R}$. Three equally qualified members of the* collegium medicum *give their opinions shown in Table 6. As Table 6 shows, the group is facing a dilemma. The majority of the conclusions from the doctors opinions indicates that the patient does not suffer from X though the majority on the premises supports the opposite conclusion.*

*Our procedure (see Table 7) selects the reasons that are most compatible with the doctors' different opinions, i.e. the judgment set (1,1,1).*

| $a$ | $b$ | $c$ | $x$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

TAB. 5 – The truth table of $\mathcal{R}$ for the doctor example.

| Condition | Dr. A | Dr. B | Dr. C | Majority |
|---|---|---|---|---|
| $a$ | 1 | 0 | 1 | 1 |
| $b$ | 1 | 0 | 1 | 1 |
| $c$ | 1 | 0 | 0 | 0 |
| $x$ | 0 | 0 | 1 | 0 |

TAB. 6 – The dilemma faced by the doctors.

| | $J_{1,p}$ | $J_{2,p}$ | $J_{3,p}$ | $\Sigma_i\ d(\omega, J_{i,p})$ |
|---|---|---|---|---|
| (1,1,1) | 0 | 3 | 1 | 4 |
| (0,0,0) | 3 | 0 | 2 | 5 |
| (0,0,1) | 2 | 1 | 3 | 6 |

TAB. 7 – Selection of the premise set from the lab results under the constraint $v(x) = 0$.

# 4 Results

We now show some properties which hold for the premise aggregation rule $F_{IC}$ we had defined in the previous section.

We start by noticing that, in the case of the aggregation of binary evaluations, there is an obvious correspondence between proposition-wise majority voting and distance minimization. This has been already observed in several contexts (see, e.g., [3]), and can be generalized to the following folk theorem.

**Theorem 4.1** *Let $\underline{J} = (J_1, \ldots, J_n)$ be a profile over the agenda $\Phi$. Let $J^{maj} \subset \Phi$ be a complete and consistent set. Let it hold that for every premise $a \in J^{maj}$, $a \in J_{i,p}$ for at least $\frac{n+1}{2}$ premise sets in the profile $J_p$. Also, for the conclusion $x \in J^{maj}$, let it holds that $x \in J_{i,c}$ for at least $\frac{n+1}{2}$ conclusion sets in the profile $J_c$. The sum of Hamming distances from $J^{maj}$ to the judgment sets in $\underline{J}$ is minimal.*

This means that, in the absence of a Paretian dilemma (i.e. when $J^{maj}$ satisfies the decision rule $\mathcal{R}$), proposition-wise majority voting, distance-based merging and our procedure coincide.

### 4.1 Unanimity preservation

One of the desirable properties for a judgment aggregation procedure is the heeding of unanimity. If all the agents unanimously support an agenda item, then it is natural to expect the unanimously supported item will be adopted as the collective judgment. However, the premise-based procedure does not necessarily preserve unanimity on the conclusion (as it was the case with the Paretian dilemma shown in Table 3).

The premise-based procedure aggregates each premise independently from the other premises, but the aggregation on the conclusion depends on the collective judgments on the premises. Therefore the unanimity on the premises will be preserved, but the unanimity on the conclusion may be violated.

When aggregating according to the conclusion-based procedure, unanimity on the conclusion will always be maintained, but unanimity on the premises may be violated. However, our procedure offers the option to heed unanimity on the premises as well, by constraining the models which do not support unanimity.

We begin by giving a formal definition on when a premise aggregation rule $F_{IC}$ preserves unanimity. Whether or not the unanimity on the premises is preserved by our $F_{IC}$ depends on the rule $\mathcal{R}$ as well as the agenda $\Phi$. We show two decision rules for which the unanimity is preserved and than we use an example to show that in the case of an arbitrary rule and agenda, the unanimity of the premises is not guaranteed.

**Definition 1** *Let $\underline{J_p} = (J_{1,p}, \ldots, J_{1,n})$ be a premise profile on the agenda $\Phi$ and $p$ a premise from the agenda. A premise aggregation rule $F_{IC}$ heeds unanimity on the premises if and only if the following holds :*
*If $p \in J_{i,p}$ for all $i = \{1, \ldots, n\}$ then $p \in F_{IC}(\underline{J_p})$.*

Note that, since $F_{IC}$ can select more than one premise judgment set, $p$ needs to be in all of them for unanimity to be preserved.

The following theorem indicates two decision rules $\mathcal{R}$, and an agenda, in the presence of which unanimity is preserved on the premises by $F_{IC}$.

**Theorem 4.2** *Let $\Phi$ be an agenda in which all the elements are atoms or negations of atoms. Let $\mathcal{R}$ be a decision rule of the form $(a_1 \wedge \ldots \wedge a_n) \leftrightarrow x$ or of the form $(a_1 \vee \ldots \vee a_n) \leftrightarrow x$. $\{a_1, \ldots, a_n\} \subseteq \Phi$ are premises and $x \subseteq \Phi$ is a conclusion. $F_{IC}$ preserves unanimity on the premises for any profile $\underline{J}$ over $\Phi$ and $\mathcal{R}$.*

***Proof*** *We give the proof for the decision rule $(a_1 \wedge \ldots \wedge a_n) \leftrightarrow x$. The proof for the decision rule $(a_1 \vee \ldots \vee a_n) \leftrightarrow x$ can be constructed symmetrically.*

*The aggregated conclusion for $x$ can be either $v(x) = 1$ or $v(x) = 0$. We construct a proof by cases :*
`Case` $v(x) = 1$
*There is exactly one set of evaluations on premises that can be selected which is consistent with $v(x) = 1$, i.e. the premise set $J_p^+$ in which every premise is evaluated to be true. Any unanimously supported premise in this case, can only be unanimously supported to have the value true and is necessarily included in $J_p^+$. Since $x = 1$, the majority of the premise judgment sets of the profile $\underline{J}$ are necessarily the premise set $J_p^+$. Consequently, the premise judgment set selected by the merging operator will select precisely the set $J_p^+$.*
`Case` $v(x) = 0$
*We construct a proof by contradiction.*

`Assumption` *: there is profile $\underline{J}$ and a premise $p$ such that $p \in J_{i,p}$ for every judgment set $i$ in the profile, and $\neg p \in F_{IC}(\underline{J_p})$. $F_{IC}(\underline{J_p})$ has the minimal distance to the judgment sets of the agents.*

*As a consequence of the assumption, $J_p$ must be the model setting $v(x) = 0$,*

$v(a_j) = 1$ *for all* $a_j \neq p$.

`Contradiction` *: Since there is a majority for* $v(x) = 0$*, there is an agent whose judgment set contains* $v(x) = 0$*, and thus, in this agents judgment set, there is a premise* $q$ *such that* $v(q) = 0$*. Now consider the judgment set* $J'$ *with* $v(a_k) = 1$ *for all* $a_k \neq q$*, and* $v(q) = 0$ *while* $v(x) = 0$ *.* $J'$ *is necessarily closer to the profile* $\underline{J}$ *than* $J$*. This is a contradiction with the assumption.*

Given an arbitrary agenda, a decision rule $\mathcal{R}$ corresponding to that agenda and an arbitrary profile $\underline{J}$, the merging operator does not necessary preserves unanimity. We show this through an example.

Consider the profile presented in Table 8. The rule $\mathcal{R}$ is such that the value of $x$ is 1 if and only if the premises are assigned as in the first column of Table 9. For all other evaluations of premises, $x$ is evaluated to 0.

| | Agent A | Agent B | Agent C | Majority |
|---|---|---|---|---|
| $p_1$ | 1 | 0 | 0 | 0 |
| $p_2$ | 0 | 1 | 0 | 0 |
| $p_3$ | 0 | 0 | 1 | 0 |
| $p_4$ | 1 | 0 | 0 | 0 |
| $p_5$ | 0 | 1 | 0 | 0 |
| $p_6$ | 0 | 0 | 1 | 0 |
| $p_7$ | 1 | 0 | 0 | 0 |
| $p_8$ | 0 | 1 | 0 | 0 |
| $p_9$ | 0 | 0 | 1 | 0 |
| $p_{10}$ | 1 | 0 | 0 | 0 |
| $p_{11}$ | 0 | 1 | 0 | 0 |
| $p_{12}$ | 0 | 0 | 1 | 0 |
| $p_{13}$ | 1 | 1 | 1 | 1 |
| $x$ | 1 | 1 | 1 | 1 |

TAB. 8 – A case in which unanimity on premises will be violated by the complete CBP.

Our procedure preserves the unanimity on the conclusion and selects $v(x) = 1$, but gives an aggregation for the premises which violates the unanimity on premise $p_{13}$ (Table 9).

The preservation of unanimously held premises can be imposed by $IC$. This is done by making $IC = \mathcal{R} \wedge \hat{x} \wedge a^*$ (see Section 3.1). Admissible outcomes for $J_p$ then are those supporting the conclusion voted by the majority and containing the premise(s) unanimously chosen.

| | $J_{1,p}$ | $J_{2,p}$ | $J_{3,p}$ | $\Sigma_i d()$ |
|---|---|---|---|---|
| (1,0,0,1,0,0,1,0,0,1,0,0,1) | 0 | 8 | 8 | 16 |
| (0,1,0,0,1,0,0,1,0,0,1,0,1) | 8 | 0 | 8 | 16 |
| (0,0,1,0,0,1,0,0,1,0,0,1,1) | 8 | 8 | 0 | 16 |
| (0,0,0,0,0,0,0,0,0,0,0,0,0) | 5 | 5 | 5 | 15 |

TAB. 9 – Selection of premises for the counterexample.

## 4.2 Manipulability

Another property which is of interest when dealing with aggregation procedures is that of manipulability. A judgment aggregation procedure is called manipulable if an agent, who would not obtain a desired outcome by submitting her "honest" premise set, can obtain a desired outcome by choosing to submit a set of premises different than her "honest" premise set. Under the context of complete-conclusion based procedures, we will distinguish between full and preferred manipulability.

*Full manipulability* means that we distinguish only whether the aggregated premise set entirely corresponds to an agent's judgments on premises, or not. A procedure is fully manipulable if an agent can obtain her complete "honest" premise set as an output from the procedure by submitting another (insincere) premise set that supports the same conclusion. Formally, let $\underline{J_p} = (J_{1,p}, J_{2,p}, \ldots, J_{i,p} \ldots, J_{n,p})$ be a premise profile. Let $F_{IC}(\underline{J_p}) = \{J^\circ_{1,p}, \ldots, J^\circ_{m,p}\}$, i.e. the merging operator selects the premise sets $J^\circ_{1,p}, \ldots, J^\circ_{m,p}$. Let $J_{i,p}$ be the "honest" premise set of an agent $i$.

**Definition 2** *Assume that a premise set* $J^*_{i,p} \neq J_{i,p}$ *exists, such that* $J^*_{i,p}$ *supports the same conclusion as the premise set* $J_{i,p}$*. The operator* $F_{IC}$ *is fully manipulable if* $J_{i,p} \in F_{IC}(J_{1,p}, J_{2,p}, \ldots, J^*_{i,p} \ldots, J_{n,p})$ *but* $J_{i,p} \notin F_{IC}(J_{1,p}, J_{2,p}, \ldots, J_{i,p} \ldots, J_{n,p})$.

**Theorem 4.3** $F_{IC}$ *is not fully manipulable.*

***Proof*** *The proof of this theorem is given in [7], under strategy proofness for complete bases when the merging operator is* $\Delta^{d,\Sigma}_{\mu}$.

Let us now assume that an agent has a premise $p$ which she holds most important (has a

strong preference on the evaluation of this premise). We say that a procedure is *preferred manipulable* if an agent can ensure that the preferred projection $w(p)$ is included in the output by submitting another premise set that supports the same conclusion. Since we do not represent the preferred premise explicitly in our framework, any premise can be the preferred one, and preferred manipulability therefore means that the agent is able to change her premise set in a way such that one premise which is not a member of the aggregated set becomes member of it.

**Definition 3** *Assume that a premise set $J^*_{i,p} \neq J_{i,p}$ exists, such that $J^*_{i,p}$ supports the same conclusion as the premise set $J_{i,p}$ and premise $p_{pref}$ is in both of the premise sets. The operator $F_{IC}$ is preferred manipulable if $p_{pref}$ is in at least one premise set $J^\circ_{j,p} \in F_{IC}(J_{1,p}, J_{2,p}, \ldots, J^*_{i,p} \ldots, J_{n,p})$, but $\neg p_{pref}$ is in all of the premise sets selected by $F_{IC}(J_{1,p}, J_{2,p}, \ldots, J_{i,p} \ldots, J_{n,p})$.*

**Theorem 4.4** *$F_{IC}$ is preferred manipulable.*

***Proof*** *To show that $F_{IC}$ is preferred manipulable it is sufficient to show that there exists a case in which an agent can ensure that the preferred premise appears in the aggregated result by misrepresenting her premise judgment set.*

*Assume the rule $\mathcal{R}$ is $x \leftrightarrow (p_1 \wedge p_2 \wedge p_3 \wedge p_4) \vee (p_1 \wedge p_2 \wedge \neg p_3 \wedge \neg p_4) \vee (\neg p_1 \wedge p_2 \wedge p_3 \wedge \neg p_4)$. The profile in which every agent submits the "honest" judgment sets is given in Table 10.*

| | Agent A | Agent B | Agent C | Majority |
|---|---|---|---|---|
| $p_1$ | 0 | 1 | 1 | 1 |
| $p_2$ | 0 | 1 | 1 | 1 |
| $p_3$ | 0 | 1 | 0 | 0 |
| $p_4$ | 0 | 1 | 0 | 0 |
| $x$ | 0 | 1 | 1 | 1 |

TAB. 10 – Preferred manipulability on premises when all agents vote "honestly".

*The full conclusion-based procedure will select $x = 1$ and the premise set $\{p_1 \wedge p_2 \wedge \neg p_3 \wedge \neg p_4\}$ (see Table 11).*

*Assume that Agent B holds premise $p_3$ as specially important and has the incentive to see it in the selected premise judgment set. If this agent submits (0,1,1,0) instead of (1,1,1,1) as in Table 12, then the selected premise judgment sets is*

| | $J_{1,p}$ | $J_{2,p}$ | $J_{3,p}$ | $\Sigma_i\ d(\omega, J_{i,p})$ |
|---|---|---|---|---|
| (0,0,0,0) | 0 | 4 | 2 | 6 |
| (1,1,1,1) | 4 | 0 | 2 | 6 |
| (1,1,0,0) | 2 | 2 | 0 | 4 |
| (0,1,1,0) | 2 | 2 | 2 | 6 |

TAB. 11 – Selection of premises for the "honest" profile.

*$\{0,0,0,0\}$ , $\{0,1,1,0\}$ $\{1,1,0,0\}$ (Table 13). Premise $p_3$ will be included in one of the selected premise judgment sets.*

| | Agent A | Agent B | Agent C | Majority |
|---|---|---|---|---|
| $p_1$ | 0 | 0 | 1 | 1 |
| $p_2$ | 0 | 1 | 1 | 1 |
| $p_3$ | 0 | 1 | 0 | 0 |
| $p_4$ | 0 | 0 | 0 | 0 |
| $x$ | 0 | 1 | 1 | 1 |

TAB. 12 – Preferred manipulability on premises when Agent B manipulates.

| | $J_{1,p}$ | $J_{2,p}$ | $J_{3,p}$ | $\Sigma_i\ d(\omega, J_{i,p})$ |
|---|---|---|---|---|
| (0,0,0,0) | 0 | 2 | 2 | 4 |
| (0,1,1,0) | 2 | 0 | 2 | 4 |
| (1,1,0,0) | 2 | 2 | 0 | 4 |
| (1,1,1,1) | 4 | 2 | 2 | 8 |

TAB. 13 – The result from the merging of the "manipulated" profile.

Full manipulability is a relatively weak condition, in the sense that it is relatively easy to satisfy. Everaere *et al.* [7] mention that their satisfaction relations corresponding to complete manipulability are the "most meaningful" ones, but in our setting, it seems that stronger notions of manipulability may be called for. Moreover, this definition of manipulability seems to conflict with the intuition of the distance measure used to aggregate the premises, which does take such distinctions into account. However, preferred manipulability is a very strong condition, since it means in practice that an agent should not be able to improve any premise (since this premise may happen to be the preferred one). Other notions of manipulability could be studied, such as the improvement of a preferred premise by changing the judgment on this premise only.

## 5 Related work

One of the noted shortcomings of the conclusion-based procedure is that it is suscep-

tible to path-dependence [17]. Path-dependent decisions are decisions whose outcome depends on the order in which propositions are considered. An order of priority over the propositions is assumed. For any proposition, the collective judgment on it is decided by majority rule (or by any other suitable aggregation rule) unless this conflicts with the collective judgments of previously aggregated propositions. In the latter case, the collective value of that proposition is deduced by logical implication from the previously aggregated propositions. List [12] provided necessary and sufficient conditions for path-dependence. Furthermore, in [5] it has been shown that the absence of path-dependence is equivalent to strategy-proofness.

Path-dependent procedures may be employed to define complete conclusion-based procedures. If a priority order is assumed on the premises, we can ensure a complete and consistent collective set of premises supporting the collective conclusion. However, here we propose a complete conclusion-based procedure without assuming any order over the premises. It is indeed arguable where such order comes from and, since different orders lead to different outcomes, path-dependend aggregation procedure are prone to strategic manipulation. In the present work we aimed at a procedure that treats all premises in an even-handed way. The conclusion-based procedure we presented in Section 3.1 avoids the problem of path-dependance by selecting the premises as a complete set instead of one premise after another. The absence of full manipulability is coherent with the results of [5].

The non-manipulability (of the outcome for the conclusion) of the conclusion-based procedure is one of its advantages over the premise-based procedure. The question of manipulability under operators used for merging of propositions has been treated extensively in [7]. There, Everaere *et al.* explore a broad spectrum of manipulability for various merging operators over complete and incomplete sets of beliefs (propositions, or in our case judgments). Our work uses results from [7] on complete sets of beliefs under model-based merging operators that use the sum of the distances between belief bases i.e. “the aggregation function $\Sigma$".

# 6 Conclusions and future work

By presenting a complete conclusion-based procedure for judgment aggregation, we show that conclusion-based procedures deserve more attention than they have received in the past. Our complete conclusion-based procedure keeps the desirable properties of non-manipulability (over the conclusion) and it can be modified to heed unanimity on the premises.

What can be considered a shortcoming of the procedure is that the procedure may select more than one premise judgment set to support the collective conclusion. Such “ties" in the output from aggregation are known to be resolved with an additional approval vote [3] or by random selection. A random selection is not a desirable tie-breaking solution in cases when the decisions on premises can influence some future decision making process. The approval voting requires more information to be injected in the framework and opens the questions of what incentives an agent may have to prefer one premise judgment set over another. In the case when there is a small number of agents, it may happen that each agent has her own judgment set in the tie and will not cast her vote of approval to any of the other premise judgment sets.

The issue which we plan to explore in future work is the relevance that current group decisions can have on future decisions when their agendas have a nonempty intersection. This “evolutionary" impact over the decision making process has been an important issue in the work that gave rise to the interest in judgment aggregation [10, 11], but it has fallen out of scope in the more formal study of judgment aggregation [16, 14]. It would be of interest, especially from the aspect of manipulability, to reintroduce the issue of “evolutionary" impact. This would require an extension of the current formal framework to allow for the “evolutionary" considerations to be part of the aggregation procedure.

# Acknowledgment

Thanks to Franz Dietrich and Christian List for discussions on the issues raised in this paper.

# Références

[1] F. Benamara, S. Kaci, and G. Pigozzi. Group decision procedures reactive to individual opinions. *Working paper*, 2009.

[2] L. Bovens and W. Rabinowicz. Democratic answers to complex questions. an epistemic perspective. *Synthese*, 150 :131–153, 2006.

[3] S. Brams, D. Kilgour, and M. Sanver. A minimax procedure for negotiating multilateral treaties. In M. Wiberg, editor, *Reasoned Choices : Essays in Honor of Hannu Nurmi*, pages 265–282. Finnish Political Science Association, 2004.

[4] B. Chapman. Rational aggregation. *Politics, Philosophy and Economics*, 1(3) :337–354, 2002.

[5] F. Dietrich and C. List. Judgment aggregation by quota rules : majority voting generalized. *Journal of Theoretical Politics*, 19(4) :391–424, 2007.

[6] F. Dietrich and P. Mongin. The premiss-based approach to judgment aggregation. Research memoranda, Maastricht : METEOR, Maastricht Research School of Economics of Technology and Organization, 2008.

[7] P. Everaere, S. Konieczny, and P. Marquis. The strategy-proofness landscape of merging. *J. Artif. Intell. Res. (JAIR)*, 28 :49–105, 2007.

[8] S. Konieczny, J. Lang, and P. Marquis. $Da^2$ merging operators. *Artificial Intelligence*, 157 :49–79, 2004.

[9] S. Konieczny and R. Pino-Pérez. Merging information under constraints : a logical framework. *Journal of Logic and Computation*, 12(5) :773–808, 2002.

[10] L. Kornhauser and L. Sager. Unpacking the court. *Yale Law Journal*, 96 :82–117, 1986.

[11] L. Kornhauser and L. Sager. The one and the many : Adjudication in collegial courts. *California Law Review*, 81 :1–51, 1993.

[12] C. List. A model of path-dependence in decisions over multiple propositions. *American Political Science Review*, 98(3) :495–513, 2004.

[13] C. List. Judgment aggregation - a bibliography on the discursive dilemma, the doctrinal paradox and decisions on multiple propositions. *http ://personal.lse.ac.uk/LIST/doctrinalparadox.htm*, 2007.

[14] C. List and P. Pettit. Aggregating sets of judgments : An impossibility result. *Economics and Philosophy*, 18 :89–110, 2002.

[15] C. List and C. Puppe. Judgment aggregation : A survey. In P. Anand, C. Puppe, and P. Pattanaik, editors, *Oxford Handbook of Rational and Social Choice*. Oxford University Press, forthcoming.

[16] P. Mongin. Factoring out the impossibility of logical aggregation. *Journal of Economic Theory*, 141(1) :100–113, July 2008.

[17] J. R. Nash. A context-sensitive voting protocol paradigm for multimember courts. *Stanford Law Review*, 56 :75–159, 2003.

[18] K. Nehring. The (im)possibility of a Paretian rational. Economics working papers, Institute for Advanced Study, School of Social Science, Nov 2005.

[19] P. Pettit. Deliberative democracy and the discursive dilemma. *Philosophical Issues*, 11 :268–299, 2001.

[20] G. Pigozzi. Belief merging and the discursive dilemma : an argument-based account to paradoxes of judgment aggregation. *Synthese*, 152(2) :285–298, 2006.

[21] G. Pigozzi, M. Slavkovik, and L. van der Torre. Conclusion-based procedure for judgment aggregation satisfying premise independence. In W. v. d. H. G. Bonanno, B. Lowe, editor, *Proceedings of the Eighth International Conference on Logic and the Foundations of Game and Decision Theory, LOFT 2008*, page 35, Amsterdam, The Netherlands, 2008.

# Une définition en logique modale de la pertinence orientée agent

S. Roussel
stephanie.roussel@onera.fr

L. Cholvy
laurence.cholvy@onera.fr

ONERA - Toulouse DTIM
BP 54025
31055 Toulouse Cedex 4 - FRANCE

**Abstract:**
Ce papier étudie la notion de la pertinence dans le contexte de système multi-agents dans lesquels les agents échangent des informations de façon coopérative. Plus précisément, on donne la définition de ce que sont les informations pertinentes pour un agent qui a un besoin en information. On analyse ensuite cette définition en mettant en avant ses limites et ses extensions.

**Mots-clés :** pertinence d'une information, systèmes multi-agents, besoin en information

**Keywords:** relevant piece of information, multiagent system, information need

## 1 Introduction

Le contexte général de travail est la modélisation de systèmes multi-agents. Dans de tels systèmes, les agents doivent agir de façon coopérative afin que la réalisation de leurs buts individuels permette la réalisation du but global du système. En particulier, les agents doivent communiquer de façon coopérative, ce qui signifie dans un premier temps que l'information échangée est facilement comprise par l'agent qui la reçoit (le récepteur), c'est à dire que son interprétation par le récepteur ne requiert ni un temps trop long ni un effort trop important. Mais plus important, cela implique que l'information échangée est exactement celle utile au récepteur pour la réalisation de son but individuel. Plus précisément, la réalisation du but individuel génère pour le récepteur des besoins en information, c'est à dire requiert l'acquisition d'informations que l'on appelle informations pertinentes.

Le concept de pertinence est un concept clé dans plusieurs domaines. Dès lors, de nombreuses approches et définitions de ce concept existent. D'après Borlund [3], les définitions de la pertinence peuvent être classées en deux catégories selon qu'elles visent à définir la pertinence "orientée système" ou la pertinence "orientée agent".

Les premières analysent la pertinence en terme de mots-clés, de sujets d'intérêt ou thèmes, de degrés de ressemblance entre une information et une requête, ou en terme de dépendance. La plupart des travaux sur cette approche peuvent être trouvés dans :

- le domaine de la recherche d'information [6, 7, 11] ; étant donnée une requête, un système de recherche d'information retourne les documents qui sont pertinents pour la requête.
- le domaine de l'intelligence artificielle [10, 12] ; dans le contexte de la logique propositionnelle, la notion de pertinence a été introduite pour caractériser la relation qui peut exister entre une formule (représentant un propos) et un ensemble de variables (représentant un sujet d'intérêt ou un thème), ou entre deux formules. Dans cette optique, l'inférence dans des bases de connaissances ou de données peut être accélérée.
- le domaine des logiques de la relevance [2, 18] ; les logiques de la relevance résolvent le problème posé par le concept de conséquence "classique". En effet, celle ci, parce ce qu'elle ne connecte pas des propositions uniquement par leurs contenus, entraîne un certain nombre de paradoxes.

D'un autre côté, les approches dites orientées agent de la pertinence tentent de définir une relation entre un agent et une information. La pertinence est analysée en terme d'utilité ou informativité pour l'agent. Ainsi, les informations pertinentes sont définies en fonction d'un besoin en information de l'agent. La plupart des travaux sur cette approche sont relativement informels.

- Dans le domaine de la recherche d'information, Borlund [3] et Mizzaro [15] classifient les différentes pertinences orientées agent en fonction du niveau de l'utilisateur considéré.
- Dans le domaine de la psychologie, Sperber et Wilson [17] définissent une théorie psychologique cognitive, la théorie de la pertinence, basée sur la définition informelle suivante : *An input (a sight, a sound, an utterance, a memory) is relevant to an individual when it connects with background information he has available to yied conclusions that matters to him.*

– Dans le domaine de la philosophie, Floridi [8] a développé une interprétation subjectiviste de la pertinence épistémique. Dans la théorie de Floridi, *the degree of relevance of a piece of information I towards an agent A is defined as a function of the accuracy of I understood by A as an answer to a query Q, given the probability that Q might be asked by A.*

Dans ce papier, notre but est de contribuer à l'étude de la pertinence orientée agent en donnant une définition formelle.

Pour cela, nous utilisons le modèle d'agents BDI (*Belief-Desire-Intention*) [19] dans lequel on considère qu'un agent est caractérisé par ses attitudes mentales, ces attitudes étant principalement la croyance, le désir et l'intention. La majorité des modèles formels basés sur BDI sont des logiques modales dont les opérateurs modaux sont utilisés pour représenter les différentes attitudes mentales. La sémantique de ces opérateurs est généralement donnée par la sémantique des mondes possibles [5].

Ce papier est organisé de la façon suivante : la section 2 présente le cadre de logique multimodale sur lequel nous basons notre travail. La section 3 traite de la pertinence concernant un besoin en information d'un agent. Dans la section 4, nous définissons une hiérarchie qui caractérise les informations les plus pertinentes. Dans la section 5, nous comparons la pertinence définie dans ce papier avec un travail du domaine de l'intelligence artificielle. Des extensions de la pertinence définie sont présentées dans les sections 6 et 7. Finalement, la section 8 conclut ce papier.

## 2 Cadre logique

On base notre travail sur le cadre logique défini dans [9]. On utilise donc une logique multimodale propositionnelle dont les opérateurs modaux sont la croyance et l'intention.

La sémantique de cette logique a été étudiée dans [9]. Dans ce papier, on ne considère que l'aspect axiomatique.

Soit $\mathcal{A}$ l'ensemble des agents.

– **Croyance.** Un opérateur modal de croyance $B_a$ est associé à chaque agent $a$ de $\mathcal{A}$. La formule $B_a\varphi$ se lit "l'agent $a$ croit que $\varphi$". On écrira $Bif_a\varphi$ au lieu de $B_a\varphi \vee B_a\neg\varphi$. $Bif_a\varphi$ se lit "l'agent $a$ sait[1] si $\varphi$". On utilise les axiomes KD45 pour la croyance [5]. Les règles d'inférence sont :
  – **Modus Ponens** : $\frac{\varphi\ \varphi\rightarrow\psi}{\psi}$
  – **Nécessitation** : $\frac{\varphi}{B_a\varphi}$
  Les axiomes sont :
  – **K** : $B_a(\varphi \rightarrow \psi) \rightarrow (B_a(\varphi) \rightarrow B_a(\psi))$
  – **D** : $B_a(\varphi) \rightarrow \neg B_a(\neg\varphi)$
  – **4** : $B_a(\varphi) \rightarrow B_a(B_a(\varphi))$
  – **5** : $\neg B_a(\varphi) \rightarrow B_a(\neg B_a(\varphi))$
  On suppose donc que les agents n'ont pas de croyances inconsistantes (D) et qu'ils sont conscients de leurs croyances (4) et de leur non-croyances (5).
– **Intention.** Un opérateur modal $I_a$ est associé à chaque agent $a$ de $\mathcal{A}$. $I_a\varphi$ se lit "l'agent $a$ a l'intention que $\varphi$". Pour cet opérateur, on suppose seulement la règle d'inférence :

$$\frac{A \leftrightarrow B}{I_aA \leftrightarrow I_aB}$$

– **Relation entre croyance et intention.** Comme dans [9], on suppose un **réalisme fort**. On considère donc que si un agent a l'intention qu'une proposition soit vraie alors il croit que cette proposition est fausse. Cela s'exprime par

$$I_aA \rightarrow B_a\neg A$$

De ce postulat, on prouve que :
  – $I_aA \rightarrow \neg B_aA$ : si un agent a l'intention qu'une proposition soit vraie alors il ne croit pas cette proposition.
  – $I_aA \rightarrow \neg I_a\neg A$ ;
  – $\neg B_aA \rightarrow \neg I_a\neg B_aA$
  De plus, on suppose des introspections négative et positive. C'est à dire que $I_aA \rightarrow B_aI_aA$ et $\neg I_aA \rightarrow B_a\neg I_aA$.

## 3 Pertinence

### 3.1 Notations préliminaires

– une formule *objective* est une formule qui ne contient pas d'opérateur modal. Une *requête* est une formule objective.
– Dans ce qui suit, l'opérateur $\otimes$ représente la disjonction exclusive généralisée à $n$ formules, c'est à dire que si $A_1$, ... $A_n$ sont $n$ formules alors $A_1 \otimes ... \otimes A_n$ est vraie si et seulement si $A_1 \vee ... \vee A_n$ est vrai et si pour tout $i$

[1] Dans le langage naturel, on utilise le verbe savoir au lieu de croire

et $j$ tels que $i \neq j$, $\neg(A_i \wedge A_j)$ est vraie. On définit l'opérateur $\bigotimes$ de la façon suivante :

$$\bigotimes_{i=1}^{n} A_i \equiv A_1 \otimes A_2 \otimes ... \otimes A_n$$

Pour $n = 1$, on a $\bigotimes_{i=1}^{1} Q_i \equiv Q_1$

### 3.2 Définition

Dans cette section, on introduit une définition formelle de la pertinence de la façon suivante :

**Définition 1** *Soit $a$ un agent de $\mathcal{A}$, $\varphi$ une formule et $Q$ une requête. On dit que $\varphi$ est pertinente pour l'agent $a$ par rapport à sa requête $Q$ si et seulement si*

$$I_a Bif_a Q \wedge \varphi \wedge (B_a(\varphi \rightarrow Q) \otimes B_a(\varphi \rightarrow \neg Q))$$

*Cette formule sera notée $R_a^Q \varphi$.*

Cette définition comprend trois principaux éléments :

- **Le besoin en information de l'agent** $I_a Bif_a Q$ : On suppose que les agents qui échangent des informations ont des besoins en information. Dans un premier temps, on suppose que ces besoins en information sont relativement simples et peuvent être modélisés de la façon suivante : "l'agent $a$ veut savoir si $Q$ ou si $\neg Q$", $Q$ étant une requête. [2] Formellement, le besoin en information s'écrit $I_a Bif_a Q$, qui signifie que l'agent $a$ a l'intention de savoir si $Q$.
- **Les croyances de l'agent** $B_a(\varphi \rightarrow Q) \otimes B_a(\varphi \rightarrow \neg Q)$ : C'est en utilisant ses croyances et l'information pertinente que l'agent doit être capable de répondre à sa requête $Q$, c'est à dire de déduire $Q$ ou déduire $\neg Q$. Pour représenter cette déduction, on choisit l'implication logique.
  Si une information permet à un agent de déduire à la fois $Q$ et $\neg Q$, alors cette information ne répond pas vraiment au besoin en information. L'utilisation de $\otimes$ empêche ce cas d'arriver [3].
- **La valeur de vérité de l'information** $\varphi$ : on considère qu'une information fausse ne peut pas être pertinente. En effet, une information fausse, bien qu'elle ait un sens, est fausse. Si on analyse la pertinence en terme d'efforts cognitifs, la désinformation est délétère. Par exemple, considérons un agent qui doit prendre le train. Ce train part à 13h05. Dans ce contexte, dire à l'agent que son train part à 13h15 est dommageable (car l'agent peut manquer son train). Ainsi, on ne peut pas considérer que l'information "Le train part à 13h15" est pertinente pour l'agent [4].

L'exemple suivant permet d'illustrer la définition de la pertinence.

**Exemple 1** *Soient $a$ et $b$ deux agents qui doivent prendre le train. Malheureusement pour eux, des incidents dans les gares peuvent bloquer les trains et entraîner des retards. Les agents veulent donc savoir si leur train est en retard (modélisé par $retard$). Considérons l'information "Il y a un incident", modélisé par $inc$. Dans notre contexte, on suppose que cette information est vraie.*

*L'agent $a$ veut savoir si sont train est en retard. Ainsi, il a le besoin en information $I_a Bif_a(retard)$. L'agent $a$ croit que s'il y a un incident, alors son train est en retard. Cette croyance est modélisée par $B_a(inc \rightarrow retard)$. Ainsi, dans ce contexte, on a :*
- $I_a Bif_a(retard)$
- $B_a(inc \rightarrow (retard)$
- $inc$
- *On peut donc déduire $R_a^{retard}(inc)$*

*Cela signifie que l'information $inc$ est pertinente pour l'agent $a$ par rapport à sa requête $retard$.*

*L'agent $b$ veut également savoir si son train est en retard ou non. Ses croyances sont différentes de celles de $a$. En effet, l'agent $b$ croit que s'il n'y a pas d'incident, alors son train n'est pas en retard. Ceci peut être modélisé par $B_b(\neg inc \rightarrow \neg retard)$ [5]. Dans ce contexte, on a donc :*
- $I_b Bif_b(retard)$
- $B_b(\neg inc \rightarrow \neg retard)$
- $inc$

*Cette fois, l'information $inc$ n'est pas pertinente pour l'agent $b$ car il ne peut déduire ni $retard$,*

[2] Dans ce papier, on ne s'intéresse ni au passage des buts individuels vers les besoins en informations, ni des besoins en information (tels qu'ils sont perçus par l'agent) vers une requête formelle.

[3] L'utilisation de $\otimes$ permet d'empêcher le cas où l'agent croit $\neg\varphi$

[4] Dans quelques cas particuliers, la désinformation peut être pertinente. Par exemple, il est pertinent pour un professeur d'apprendre qu'un de ses élèves ne sait pas sa leçon. Cependant, dans ce cas, ce n'est pas la leçon fausse qui est pertinente mais le fait que l'élève se trompe.

[5] Il faudrait également supposer que $b$ n'a pas d'autre croyance sur incident et retard, en particulier, $\neg B_b(inc \rightarrow retard)$ et $\neg B_b(inc \rightarrow \neg retard)$

*ni* $\neg retard$.
*L'information* $\neg inc$*, qui est fausse dans notre contexte, ne peut pas être pertinente pour* $b$ *car certes elle lui permettrait de répondre à son besoin en information mais sa conclusion serait erronée.*

### 3.3 Propriétés

Dans cette partie, on étudie quelques propriétés de la pertinence définie dans la partie précédente. Soient $a$ un agent de $\mathcal{A}$, $Q$, $Q_1$ et $Q_2$ des requêtes, $\varphi$, $\varphi_1$, $\varphi_2$ des formules. Les propositions qui suivent sont des théorèmes de la logique définie. [6]

**Proposition 1** $R_a^Q\varphi \rightarrow \neg B_a\varphi \wedge \neg B_a\neg\varphi$

Si une information $\varphi$ est pertinente pour un agent $a$, alors cet agent ne croit ni $\varphi$ (sinon, il pourrait déjà répondre à son besoin en information), ni $\neg\varphi$ (sinon il pourrait à la fois déduire $Q$ et $\neg Q$).

**Proposition 2** *Soit* $*$ *un opérateur de révision de croyances satisfaisant les postulats AGM [1].* $Bel_a$ *représente l'ensemble des croyances de l'agent* $a$ *et* $Bel_a * \varphi$ *l'ensemble des croyances de* $a$ *après révision par* $\varphi$ *en utilisant l'opérateur* $*$*. Alors, on a* $R_a^Q\varphi \rightarrow ((Bel_a * \varphi) \rightarrow Q) \otimes ((Bel_a * \varphi) \rightarrow \neg Q)$

Cette proposition montre que l'opérateur de déduction que l'on a choisi, l'implication logique, correspond à un opérateur "basique" de révision de croyances. En effet, si un agent révise ses croyances avec l'information pertinente, sa nouvelle base de croyances contiendra la réponse à son besoin en information.

**Proposition 3**
- $I_aBif_aQ \rightarrow R_a^QQ \otimes R_a^Q\neg Q$ *: une et une seule des informations* $Q$ *ou* $\neg Q$ *est pertinente pour l'agent* $a$ *par rapport à sa requête* $Q$*.*
- $(Q_1 \leftrightarrow Q_2) \rightarrow (R_a^{Q_1}\varphi \leftrightarrow R_a^{Q_2}\varphi)$ *: si deux requêtes sont équivalentes, toute information pertinente par rapport à la première est pertinente par rapport à la deuxième.*
- $R_a^Q\varphi \leftrightarrow R_a^{\neg Q}\varphi$ *: une information pertinente pour la requête* $Q$ *est également pertinente pour la requête* $\neg Q$*.*
- $\neg(\varphi_1 \wedge \varphi_2) \rightarrow \neg(R_a^{Q_1}\varphi_1 \wedge R_a^{Q_2}\varphi_2)$ *: deux informations contradictoires ne peuvent pas être toutes deux pertinentes.*

**Proposition 4** $R_a^Q\varphi \rightarrow \neg B_aR_a^Q\varphi$

Si une information $\varphi$ est pertinente pour un agent $a$, alors $a$ ne le sait pas. Ceci est du à la valeur de vérité de l'information présente dans la définition de la pertinence. Si un agent croit qu'une information est pertinente pour lui, alors c'est qu'il croit que cette information est vraie. S'il croit que cette information est vraie, alors il peut déduire de sa base de croyances la réponse à son besoin en information. C'est en contradiction avec le fait que l'agent ait un besoin en information.

**Notation.** Dans ce qui suit, on écrira $B_a(\varphi_1, \varphi_2/Q)$ au lieu de $\neg(B_a(\varphi_1 \rightarrow Q) \wedge B_a(\varphi_2 \rightarrow \neg Q)) \wedge \neg(B_a(\varphi_1 \rightarrow \neg Q) \wedge B_a(\varphi_2 \rightarrow Q))$. Cette formule signifie que l'agent $a$ croit que $\varphi_1$ et $\varphi_2$ ne permettent pas de déduire une contradiction par rapport à $Q$.

**Proposition 5** $B_a(\varphi_1, \varphi_2/Q) \rightarrow (\varphi_2 \wedge R_a^Q\varphi_1 \rightarrow R_a^Q(\varphi_1 \wedge \varphi_2))$

**Proposition 6** $B_a(\varphi_1, \varphi_2/Q) \rightarrow (R_a^Q\varphi_1 \wedge R_a^Q\varphi_2 \rightarrow R_a^Q(\varphi_1 \vee \varphi_2))$

Ces deux propositions montrent que la définition que l'on donne de la pertinence caractérise trop d'informations pertinentes. L'exemple suivant illustre ce propos.

**Exemple 2** *Considérons l'exemple du train qui peut être en retard à cause d'incidents. L'agent* $a$ *a besoin de savoir si train est en retard ou non et on suppose que l'information* $inc$ *est pertinente pour lui concernant cette requête.*
*Supposons que l'information "il pleut", modélisée par* $pluie$ *soit vraie dans notre contexte. Alors, l'information* $inc \wedge pluie$ *est pertinente pour l'agent* $a$*. En effet, elle contient tous les éléments nécessaires à* $a$ *pour qu'il puisse répondre à son besoin en information. Néanmoins, intuitivement, l'information* $inc$ *est plus pertinente que* $inc \wedge pluie$ *car cette dernière contient l'élément* $pluie$ *qui n'est pas nécessaire à* $a$ *par rapport à* $retard$*.*

[6] Pour alléger, on n'écrira pas le symbole $\vdash$ devant les théorèmes.

Toutes les informations caractérisées comme pertinentes sont "suffisamment" pertinentes. En effet, chacune permet de répondre au besoin en information. D'un autre côté, on pourrait considérer les informations qui sont "nécessairement" pertinentes, c'est à dire celles sans lesquelles l'agent $a$ ne peut pas répondre à son besoin en information. En combinant les deux concepts, on peut trouver, parmi les informations "suffisamment" pertinentes, celles qui le sont le plus "nécessairement". Ainsi, les informations les plus nécessaires sont celles qui sont le plus pertinentes.

# 4 Une hiérarchie pour la pertinence

Dans cette section, on caractérise la notion de "nécessairement pertinent" décrite précédemment.

## 4.1 Explication minimale

Pour cela, on introduit tout d'abord la notion d'explication minimale. Cette notion a été utilisée dans [10] par Lakemeyer pour définir la pertinence. Cependant, la définition d'explication minimale de [10] est très syntaxique [7] Afin d'avoir une définition "plus sémantique", on adapte la définition de Lakemeyer en utilisant des notions d'indépendance sémantique définies dans [12].

**Définition 2** *Soit $\varphi$ une formule objective. On dit que $\varphi$ est en Forme Négation Normal (NNF) si et seulement si seuls des symboles propositionnels apparaissent derrière un $\neg$ dans $\varphi$.*
*On note $Lit(\varphi)$ l'ensemble des littéraux apparaissant dans la forme NNF de $\varphi$.*

Par exemple, $\varphi = \neg((\neg a \wedge b) \vee c)$ n'est pas dans sa forme NNF car après le premier $\neg$, on trouve autre chose qu'un symbole propositionnel (en l'occurrence, un autre $\neg$). La forme NNF de $\varphi$ est $(a \vee \neg b) \wedge \neg c$. On a donc $Lit(\varphi) = \{a, \neg b, \neg c\}$.

**Définition 3** *Soit $\varphi$ une formule objective et $l$ un littéral.*
*On dit que $\varphi$ est* syntaxiquement Lit-dépendante *de $l$ (resp.* syntaxiquement Lit-indépendante *de $l$) si et seulement si $l \in Lit(\varphi)$ (resp. $l \notin Lit(\varphi)$).*

[7] En effet, il utilise la forme CNF d'une formule. Mais pour une formule donnée, cette forme n'est pas unique.

**Définition 4** *Soit $\varphi$ une formule objective et $l$ un littéral.*
*On dit que $\varphi$ est* Lit-indépendante *de $l$ si et seulement si il existe une formule $\Sigma$ telle que $\Sigma \equiv \varphi$ et $\Sigma$ est syntaxiquement Lit-indépendante de $l$. Dans le cas contraire, on dit que $\varphi$ est* Lit-dépendante *de $l$. Etant donné un langage, l'ensemble des littéraux $l$ de ce langage tels que $\varphi$ est Lit-indépendante de $l$ est noté $DepLit(\varphi)$.*

**Exemple 3** *Soit $\varphi = (a \wedge \neg b \wedge (a \vee b))$. On a $DepLit(\varphi) = \{a, \neg b\}$. Notons que $\varphi$ est Lit-independante de $b$ car elle est équivalente à $\Sigma = (a \wedge \neg b)$ dans laquelle $b$ n'apparaît pas positivement.*

On peut maintenant donner la définition de l'explication minimale.

**Définition 5** *Soit $\Delta$ un ensemble fini de formules objectives et $\alpha$ et $\beta$ deux formules objectives.*
*$\beta$ est une* explication *de $\alpha$ si et seulement si $\vdash B\Delta \rightarrow B(\beta \rightarrow \alpha)$ et $\nvdash B\Delta \rightarrow B(\neg\beta)$* [8].
*$\beta$ est une* explication minimale *de $\alpha$ si et seulement si $\beta$ est une explication minimale de $\alpha$ et il n'existe pas d'explication $\beta'$ de $\alpha$ telle que $DepLit(\beta') \subset DepLit(\beta)$.*

## 4.2 Informations les plus pertinentes

A partir de cette notion d'explication minimale, on peut définir quelles sont les informations les plus pertinentes.

Soit $\mathcal{R}_a^Q$ l'ensemble des formules pertinentes pour l'agent $a$ par rapport à sa requête $Q$. Pour tout $\varphi$ dans $\mathcal{R}_a^Q$, on a $B_a(\varphi \rightarrow Q)$ ou $B_a(\varphi \rightarrow \neg Q)$ et on a également $\neg B_a(\neg\varphi)$. Cela signifie que $\varphi$ est une explication de $Q$ ou de $\neg Q$.

**Définition 6** *On appelle $\mathcal{R}m_a^Q$ le sous-ensemble de $\mathcal{R}_a^Q$ qui contient les explications minimales de $Q$ et les explications minimales de $\neg Q$.*

**Exemple 4** *Supposons que l'ensemble des informations pertinentes pour un agent $a$ par rapport à sa requête $Q$ soit $\mathcal{R}_a^Q = \{inc \wedge pluie, inc \vee greve, greve\}$. Alors, $\mathcal{R}m_a^Q = \{greve, inc \wedge pluie\}$.*

[8] L'opérateur B utilisé ici est l'opérateur de croyance non associé à un agent

Ainsi, on peut caractériser les informations suffisamment et nécessairement (pour l'explication minimale) pertinentes. On peut remarquer que :

- si $\mathcal{R}_a^Q$ ne contient que des clauses, alors les clauses les plus pertinentes sont les maximales pour la subsumption. Cela correspond aux formules les plus précises.
- si $\mathcal{R}_a^Q$ ne contient que des cubes (conjonction de littéraux), les cubes les plus pertinents sont les impliquants premiers de $Q$ et impliquants premiers de $\neg Q$. Cela correspond aux formules les plus concises.

Ce compromis entre "être concis" et "être précis" s'accorde avec de nombreux travaux réalisés dans le domaine de de l'informativité ou la théorie de la pertinence ([16, 4, 17]).

Bien sûr, d'autres définitions de ce "nécessaire" peuvent être utilisées. Pour une autre définition, l'ensemble des formules nécessairement pertinentes pourrait donc être différent.

# 5 La pertinence de Lakemeyer

L'utilisation de la notion d'explication minimale permet de comparer la pertinence définie dans ce papier avec celle définie par Lakemeyer [10].

## 5.1 Définition

Dans [10], Lakemeyer définit plusieurs pertinences et compare sa notion de pertinence avec plusieurs concepts de la littérature ([14], [13]). Néanmoins, dans ce papier, on ne s'intéressera qu'à la pertinence qui est la plus proche de celle que l'on a définie ici. Pour éviter toute confusion entre la pertinence de Lakemeyer et celle définie dans ce papier, on écrira "L-pertinence" pour parler de la pertinence de Lakemeyer.

Tout d'abord, rappelons quelques définitions et notations utilisées dans [10].

**Définition 7** – *Une formule ou un ensemble de formules* $\Delta$ mentionne *un atome* $p$ *si* $p$ *ou si* $\neg p$ *apparaît dans* $\Delta$.
– *un* sujet d'intérêt $\pi$ *est un ensemble d'atomes.*
– $\pi_\Delta = \{p \mid \Delta$ *mentionne* $p\}$
– *Une formule est* triviale *si elle est équivalente à une tautologie.*

Lakemeyer définit alors la L-pertinence de la façon suivante :

**Définition 8** *Soient* $\pi$ *un sujet d'intérêt,* $\alpha$ *une formule objective et* $\Delta$ *un ensemble de formules objectives.* $\pi$ *est* L-pertinent *pour* $\alpha$ *par rapport à* $\Delta$*, noté* $RX_\Delta(\pi, \alpha)$*, si et seulement si il existe une explication minimale de* $\alpha$ *non triviale qui mentionne un atome* $p$ *de* $\pi$

Intuitivement, un sujet d'intérêt est pertinent pour une formule s'il contient des atomes qui sont nécessaires pour l'explication de cette formule.

**Exemple 5** *Considérons l'ensemble* $\Delta = \{inc \wedge greve \rightarrow retard, inc \wedge greve \wedge pluie \rightarrow retard\}$*. Dans cet ensemble, l'explication minimale de* $retard$ *est* $inc \wedge greve$*. Ainsi, tout sujet d'intérêt contenant soit* $inc$ *soit* $greve$ *est L-pertinent pour* $retard$ *par rapport à* $\Delta$*.*

En fait, la L-pertinence caractérise les formules "nécessairement" pertinentes. L'aspect "suffisamment" pertinent n'est pas considéré ici alors qu'il est à la base de la pertinence définie ici.

## 5.2 Comparaison

Afin de pouvoir étudier le rapport entre la L-pertinence et la pertinence de ce papier, il faut considérer un agent et l'ensemble de croyances qui lui est associé. Cet ensemble correspond à l'ensemble de formules $\Delta$ de Lakemeyer. Les formules pour lesquelles on cherche une explication sont la requête de l'agent et sa négation. Le sujet d'intérêt potentiellement pertinent pour ces formules est l'ensemble des atomes composant l'information potentiellement pertinente $\varphi$.

**Proposition 7** *Soient* $\varphi$ *et* $Q$ *deux formules objectives. Soit* $Bel_a$ *l'ensemble des croyances de l'agent* $a$*. On suppose que cet ensemble est fini. Supposons que l'agent* $a$ *ait besoin de savoir si* $Q$ *ou si* $\neg Q$*.*
*On note* $\mathcal{R}_a^Q$ *l'ensemble des formules qui sont pertinentes pour lui par rapport à sa requête* $Q$*. Si* $\varphi \in \mathcal{R}_a^Q$*, alors il existe une explication minimale de* $Q$ *ou de* $\neg Q$ *qui mentionne au moins un atome de* $\varphi$*, c'est à dire que l'on a la L-pertinence (*$RX_{\Delta_a}(\pi_\varphi, Q) \otimes RX_{\Delta_a}(\pi_\varphi, \neg Q)$*).*

Si une information est pertinente par rapport à une requête, alors elle contient des éléments nécessaires pour expliquer la requête ou la négation de la requête.

La relation entre les deux pertinences est encore plus étroite lorsque l'on considère l'ensemble des formules les plus pertinentes $\mathcal{R}m_a^Q$.

**Proposition 8** *Soit $\varphi$ une formule de $\mathcal{R}m_a^Q$. Alors, non seulement, il existe une explication minimale de $Q$ ou de $\neg Q$ qui mentionne au moins un atome de $\varphi$ ($RX_{\Delta_a}(\pi_\varphi, Q) \otimes RX_{\Delta_a}(\pi_\varphi, \neg Q)$) mais $\varphi$ est une telle explication minimale.*

Cela montre que si une information est suffisamment et nécessairement pertinente, alors non seulement elle est L-pertinente mais on peut caractériser cette L-pertinence plus précisément que Lakemeyer car l'explication minimale à laquelle il fait référence est l'information pertinente.

## 6 Retour aux systèmes multi-agents

Comme mentionné en introduction, nous considérons des agents qui communiquentq de façon coopérative. Cela signifie que pour chaque échange d'information, l'émetteur d'information $b$ sait quelles sont les informations pertinentes pour le récepteur d'information $a$ par rapport à ses besoins en information.

Ceci est représenté par la formule suivante :

$$B_b I_a B if_a(Q) \wedge B_b\varphi$$
$$\wedge B_b(B_a(\varphi \rightarrow Q) \otimes B_a(\varphi \rightarrow \neg Q))$$

Cette formule signifie que $b$ pense que la formule $\varphi$ est pertinente pour l'agent $a$ par rapport à un besoin en information de celui-ci si :
– l'agent $b$ croit que l'agent $a$ a ce besoin en information.
– l'agent $b$ croit que l'agent $a$ peut répondre à son besoin en information avec ses croyances et l'information pertinente.
– l'agent $b$ croit que l'information $\varphi$ est vraie.
A ce stade, on peut noter que l'agent $b$ peut se tromper sur beaucoup de points quant à la pertinence d'une information pour un autre agent : il peut se tromper sur le besoin en information de l'autre, sur les croyances de l'autre ou sur la valeur de vérité de l'information qu'il croit pertinente.

Le cas où une information pertinente n'est pas possédée par un mais plusieurs agents doit encore être étudié. Par exemple, l'information pertinente pour $a$ pourrait être $p \wedge q$. L'agent $b$ ne croit que $p$ et l'agent $c$ ne croit que $q$. Dans ce cas, une pertinence partielle doit être définie pour caractériser les différents éléments de l'information pertinente.

## 7 Généralisation du besoin en information

Jusqu'à maintenant, les besoins en information considérés étaient de la forme "l'agent $a$ veut savoir si $Q$ ou si $\neg Q$". Ce besoin peut être étendu de la façon suivante : "l'agent $a$ veut savoir si $Q_1$ ou si $Q_2$ ... ou si $Q_n$", $Q_1, ...Q_n$ étant des formules objectives mutuellement exclusives.

Ainsi, on peut étendre la pertinence de la façon suivante :

**Définition 9** *Soit $\mathbf{Q}$ un ensemble de $n$ ($n \geq 2$) formules objectives $Q_i$ mutuellement exclusives. On dit que $\varphi$ est pertinente pour $a$ par rapport à $\mathbf{Q}$ si et seulement si*

$$I_a \bigotimes_{i=1}^{n} B_a Q_i \wedge \bigotimes_{i=1}^{n} B_a(\varphi \rightarrow Q_i) \wedge \varphi$$

*Cette formule sera notée $R_a^{\mathbf{Q}}\varphi$.*

Ainsi, une information pertinente répond au besoin en information en permettant à l'agent de déduire un et un seul des $Q_i$.

**Exemple 6** *Considérons un agent $a$ qui a besoin de savoir sur quel quai de gare aller pour prendre son train. On suppose qu'il y a trois quais dans la gare. Le train part du quai 1 (resp 2 and 3) est modélisé par $q_1$ (resp. $q_2$ and $q_3$). On note $\mathbf{Q}$ l'ensemble $\{q_1, q_2, q_3\}$.*

*Supposons que l'agent $a$ croit que le train qu'il doit prendre est un TGV (modélisé par $TGV$). Considérons l'information $TGV \rightarrow q_1$ (signifiant que si le train est un TGV, alors il part du quai 1) et supposons qu'elle soit vraie dans notre contexte. On a donc :*

– $I_a(B_a q_1 \otimes B_a q_2 \otimes B_a q_3)$
– $TGV \rightarrow q_1$
– $B_a(TGV)$

*On peut donc déduire que $R_a^{\mathbf{Q}}(TGV \rightarrow q_1)$. En effet, l'information $TGV \rightarrow q_1$ permet à l'agent $a$ de déduire sur quel quai se rendre pour prendre son train.*

La plupart des propriétés vues plus tôt pour la pertinence sont également vraie pour la pertinence étendue. Soient $a$ un agent, $\mathbf{Q} = \{Q_i\}_{i=1...n}$ un ensemble de formules objectives mutuellement exclusives tel que $n \geq 2$ et $\varphi$, $\varphi_1$ and $\varphi_2$ trois formules. Les propositions qui suivent sont des théorèmes de notre logique.

**Proposition 9**

$$R_a^{\mathbf{Q}}\varphi \rightarrow \neg B_a \neg\varphi \wedge \neg B_a \varphi$$

**Proposition 10** *Soit $*$ un opérateur de révision de croyances satisfaisant les postulats AGM [1]. $Bel_a$ représente l'ensemble des croyances de l'agent $a$ et $Bel_a * \varphi$ l'ensemble des croyances de $a$ après révision par $\varphi$ en utilisant l'opérateur $*$. Alors, on a*

$$R_a^{\mathbf{Q}}\varphi \rightarrow \bigotimes_{i=1}^{n}((Bel_a * \varphi) \rightarrow Q_i)$$

**Proposition 11**

$$\bigotimes_{i=1}^{n} Q_i \wedge I_a \bigotimes_{i=1}^{n} B_a Q_i \rightarrow \bigotimes_{i=1}^{n} R_a^{\mathbf{Q}} Q_i$$

L'hypothèse $Q_1 \otimes ... \otimes Q_n$ est ici nécessaire pour s'assurer qu'une requête $Q_i$ est vraie.

**Proposition 12**

$$R_a^{\mathbf{Q}}\varphi \rightarrow \neg B_a R_a^{\mathbf{Q}}\varphi$$

Comme pour la pertinence définie au début de ce papier, la pertinence étendue caractérise trop d'informations comme étant pertinentes. Comme dans la section 4, on pourrait donc définir une hiérarchie parmi les informations pertinentes.

Avec cette définition étendue, une information pertinente répond au besoin en information car elle permet à l'agent de déduire une des $Q_i$. Mais il existe des informations qui peuvent permettre à l'agent d'éliminer des possibles $Q_i$. Dans ce cas, même si l'information n'est pas suffisamment pertinente (car elle ne permet pas de répondre au besoin en information), elle est partiellement pertinente. Ainsi, il est possible de définir des degrés de pertinence qui dépendant du nombre de possibilités $Q_i$ éliminées par l'information.

# 8 Conclusion

La principale contribution de ce travail est la définition et la formalisation d'une pertinence orientée agent. Étant donné un agent qui a un besoin en information, on a introduit une définitions des informations pertinentes pour lui. Ainsi, une information est pertinente pour un agent si l'agent a un besoin en information, si à partir de l'information et de ses croyances, l'agent peut répondre à son besoin en information et si l'information est vraie.

Cependant, cet définition caractérise trop d'informations comme étant pertinentes. Dans ce cas, il est alors possible de définir une hiérarchie pour déterminer quelles sont les informations les plus pertinentes. Dans cette hiérarchie, un compromis entre concision et précision doit être trouvé. L'explication minimale est un tel compromis.

Avec ce compromis qu'est l'explication minimale, on peut comparer la pertinence définie ici avec celle de Lakemeyer. Cette dernière peut être vue comme une caractérisation des formules nécessairement pertinentes alors que celle définie ici caractérise les informations suffisamment pertinentes.

En plus des extensions mentionnées à la fin des sections 6 et 7, ce travail peut être étendu de plusieurs manières.

Par exemple, dans ce papier on ne considère pas le passage des buts individuels vers les besoins en information. Néanmoins, si le but individuel est formalisable, il serait possible d'exprimer directement la pertinence pour ce but.

Il serait également intéressant de considérer d'autres besoins que le besoin en information. Par exemple, un agent pourrait avoir un besoin de vérification. Dans ce cas, toute information confirmant ou contredisant ses croyances serait pertinente. De la même façon, il pourrait avoir un besoin de complétion : il aurait besoin de savoir toutes les informations dans un domaine donné. Toute information dans ce domaine non connue par l'agent serait alors pertinente pour lui.

# Remerciements

Ce travail a été réalisé pendant une thèse financée par la DGA (Direction Générale de l'Armement).

## Références


[1] Carlos E. Alchourròn, Peter Gärdenfors, and David Makinson. On the logic of theory change : partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50 :510–530, 1985.

[2] Alan R. Anderson and Nuel D. Belnap. *Entailment : The Logic of Relevance and Necessity*, volume 1. Princeton University Press, Princeton, 1975.

[3] Pia Borlund. The Concept of Relevance in IR. *Journal of the American Society for Information Science and Technology*, 54(10) :913–925, 2003.

[4] Robyn Carston. Informativeness, relevance and scalar implicature. pages 179–236. Uchida,S., 1998.

[5] Brian F. Chellas. Modal logic : An introduction. Cambridge, MA, 1980. Cambridge University Press.

[6] Jean-Pierre Chevallet. Modélisation logique pour la recherche d'information. *Les systèmes de recherche d'information*, pages 105–138, 2004.

[7] Fabio Crestani and Mounia Lalmas. Logic and uncertainty in information retrieval. *Lecture Notes in Computer Science*, pages 179–206, 2001.

[8] Luciano Floridi. Understanding epistemic relevance. *Erkenntniss*, 2007.

[9] Andreas Herzig and Dominique Longin. A logic of intention with cooperation principles and with assertive speech acts as communication primitives . In C. Castelfranchi and W. L. Johnson, editors, *Proc. 1st Int. Joint Conf. on Autonomous Agent and Multi-Agent System (AAMAS 2002) , Bologna*, pages 920–927. ACM Press, 2002.

[10] Gerhard Lakemeyer. Relevance from an epistemic perspective. *Artif. Intell.*, 97(1-2) :137–167, 1997.

[11] Mounia Lalmas and C.J. van Rijsbergen. A model of information retrieval system based on situation theory and dempster-shafer theory of evidence. In *Proceedings of the 1st Workshop on Incompleteness and Uncertainty in Information Systems*, pages 62–67, 1993.

[12] Jérôme Lang, Paolo Liberatore, and Pierre Marquis. Propositional independence - formula-variable independence and forgetting. *Journal of Artificial Intelligence Research*, 18 :391–443, 2003.

[13] Fangzhen Lin and Raymond Reiter. Forget it ! In Russell Greiner and Devika Subramanian, editors, *Working Notes, AAAI Fall Symposium on Relevance*, pages 154–159, Menlo Park, California, 1994. American Association for Artificial Intelligence.

[14] Pierre Marquis. Novelty revisited. In *ISMIS '91 : Proceedings of the 6th International Symposium on Methodologies for Intelligent Systems*, pages 550–559, London, UK, 1991. Springer-Verlag.

[15] Stefano Mizzaro. How many relevances in information retrieval ? *Interacting with Computers*, 10(3) :303–320, 1998.

[16] Guy Politzer. L'informativité des énoncés : contraintes sur le jugement et le raisonnement. In *Intellectica*, volume 1, pages 111–147, 1991.

[17] Dan Sperber and Deirdre Wilson. Relevance theory. In G. Ward and L. Horn, editors, *Handbooks of Pragmatics*, pages 607–632. Oxford : Blackwell, 2004.

[18] Eugeni K. Voishvillo. Philosophical-methodological aspects of relevant logic. In *Bulletin of Section of Logic*, volume 18/4, pages 146–151, 1989.

[19] Michael Wooldridge. In *Reasoning about rational agents*, Cambridge, Massachusetts, 2000. The MIT Press.

# CONFÉRENCES INVITÉES

# The Complexity of Computing a Nash Equilibrium

P. Goldberg*
P.W.Goldberg@liverpool.ac.uk

*University of Liverpool

**Abstract:**
Game theory is the study of mathematical models of strategic, non-cooperative interaction amongst selfish agents. For these mathematical models, one seeks solutions, which model the outcome of the game. In 1951, John Nash obtained his celebrated result that every game admits a - possibly randomised - solution, now known as Nash equilibrium in his honour. The proof of existence is non-constructive however, raising the question of how to compute a solution algorithmically, a question of considerable interest in the context of theoretical Computer Science, as well as Game Theory. In the talk, I will survey work by myself and others that has shed light on the computational complexity of this problem, by relating it to the computation of generic Brouwer fixpoints.

**Biography:**
Prof. Paul Goldberg has been at the University of Liverpool for three years, where he currently leads a newly-formed research group on Economics and Computation. Prior to that, he taught at the University of Warwick, and was a researcher at Aston University and Sandia National Laboratories, New Mexico, USA. Prof Goldberg's paper "The Complexity of Computing a Nash Equilibrium" co-authored with Christos Papadimitriou and Constantinos Daskalakis, was awarded the Game Theory and Computer Science Prize at the 3rd World Congress of the Game Theory Society, in July 2008. Prof. Goldberg has worked on various other aspects of algorithmic game theory, and prior to that, on computational learning theory.

# Défis pour les approches formelles pour l'ingénierie des systèmes interactifs

P. Palanque*
palanque@irit.fr

*IHCS-IRIT,
Université Paul Sabatier - Toulouse 3
118, route de Narbonne
31062 Toulouse Cedex
Tel: +33 561 55 6965
Fax: +33 561 55 6258

**Résumé :**
Cette présentation va faire un tour d'horizon et un historique de l'exploitation des approches formelles dans le domaine de l'Interaction Homme-Machine et plus particulièrement en ce qui concerne la conception, la construction et l'évaluation des systèmes interactifs. Seront ensuite présentés un ensemble de cas montrant l'utilisation effective de ces approches sur des problèmes particuliers en prenant comme champ d'application particulier les systèmes critiques tels que le contrôle aérien, les cockpits d'avions ou les segments sol de gestion de satellites. Nous montrerons en particulier comment la notation ICO et l'outil PetShop, développés au sein de l'équipe, ont été appliqués à ces différents domaines. Cette présentation conclura par l'identification d'un agenda de recherche et de problèmes restant à résoudre ainsi qu'une « roadmap » pour les travaux futurs.

**Biography:**
Dr. Philippe Palanque is a Professor in Computer Science at the University Toulouse 3 – Paul Sabatier and is currently Head of the IHCS (http ://ihcs.irit.fr) at IRIT. From 1995 he spent 2 years at DSNA/DTI/R&D (formally known as the CENA, the French Research Centre of Civil Aviation) to develop and apply formal specification and interactive system design techniques in the field of Air Traffic Control. For 4 years he has been involved in several research projects funded by the French Department of Defense dealing with notations and tools for the specification of real-time interactive systems and applied to cockpits of military aircrafts and drones. More recently, he is involved in developing tools and techniques (in the domain of civil aviation) for the specification and certification of new interactive cockpits (that have to be compliant with ARINC 661 specification standard) as the ones of large commercial aircrafts. He has been working on research projects to improve interactive Ground Segment Systems, funded by the French Space Agency (CNES) for 10 years, beginning in 2001. His current research interests deal with formal description techniques to support design and construction of resilient interactive systems considering in a balanced way usability, safety and dependability. He is chair of the IFIP Working group 13.5 on Human Error Safety and System Development and is adjunct chair for specialized conferences within ACM SIGCHI (Special Interest Group on Human-Computer Interaction). He edited and co-edited eight books or conference proceedings and co-authored more than 100 refereed publications in international conferences and journals.

# Changing conditional norms

L. van der Torre*
leon.vandertorre@uni.lu

*University of Luxembourg

**Abstract:**
AGM theory change was developed as a formal framework for norm change, but its restriction to propositional theories does not make it suitable for conditional norms. In this talk we discuss the relevance of norm change for formal models of interaction, and we present some results for change of conditional norms.

**Biography:**
Leon van der Torre obtained his PhD from the University of Rotterdam in 1997, with his PhD thesis entitled Reasoning about obligations : preferences in defeasible deontic logic. In 2006 he started as a full professor in intelligent systems at the University of Luxembourg, where he is responsible for strategic priority P1 on security and reliability of University of Luxembourg. On May 2nd, 2007, he gave his inaugural speech entitled "Violation games : a new approach of handling norms in intelligent systems." He works on deontic logic, knowledge representation and multiagent systems. He developed the BOID agent architecture (with colleagues from Vrije Universiteit), the input/output logics (with David Makinson) and the game-theoretic approach to normative multiagent systems (with Guido Boella from University of Turin).

# Index des auteurs

Amgoud, Leila, 3, 15, 81

Balbiani, Philippe, 25, 35
Benferhat, Salem, 47
Bonnefon, Jean-François, 213
Bonzon, Élise, 57
Bouchet, François, 69
Bourguet, Jean-Rémi, 81
Burkov, Andriy, 93

Chaib-draa, Brahim, 93
Chanoni, Emilie, 101
Cheikh, Fahima, 25
Cholvy, Laurence, 147, 279

Demolombe, Robert, 113
Dessalles, Jean-Louis, 125
Devred, Caroline, 57
Dimulescu, Adrian, 125
Dupin de Saint-Cyr, Florence, 3

Everaere, Patricia, 135

Feuillade, Guillaume, 25

Garion, Christophe, 147
Goldberg, Paul, 291

Hoet, Shirley, 159
Houy, Nicolas, 171

Jeantet, Gildas, 181

Konieczny, Sébastien, 135, 193

Lagasquie-Schiex, Marie-Christine, 57
Lagrue, Sylvain, 47
Lang, Jérôme, 203
Leblois, Sylvie, 213
Lecroq, Thierry, 101
Lespérance, Yves, 223

Mancarella, Paolo, 233
Marquis, Pierre, 135
Mathieu, Philippe, 233, 243
Mengin, Jérôme, 203
Morge, Maxime, 233

Nongaillard, Antoine, 243

Ouerdane, Wassila, 255

Palanque, Philippe, 293
Pauchet, Alexandre, 101
Pigozzi, Gabriella, 267
Pino Pérez, Ramon, 193

Rossit, Julien, 47
Roussel, Stéphanie, 147, 279

Sabouret, Nicolas, 159
Sansonnet, Jean-Paul, 69
Seban, Pablo, 35
Shapiro, Steven, 223
Slavkovik, Marija, 267
Spanjaard, Olivier, 181

Thomopoulos, Rallou, 81

van der Torre, Leon, 267, 295
van Ditmarsch, Hans, 35
Vesic, Srdjan, 15